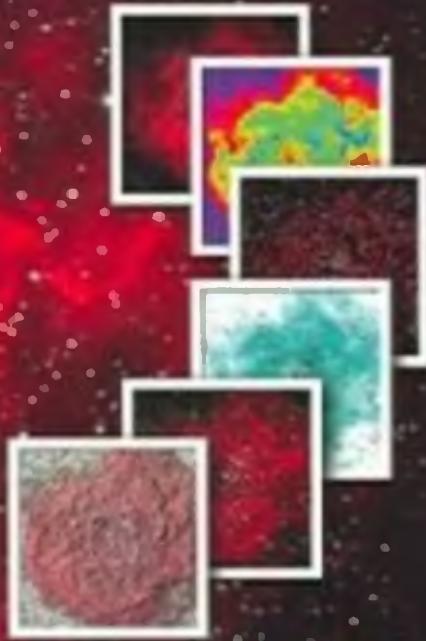


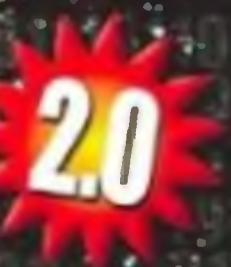
The Handbook of Astronomical

# IMAGE PROCESSING

Richard Berry  
James Burnell



Includes  
**AIP4WIN**  
Software



Published by Willmann-Bell, Inc.  
P.O. Box 35025, Richmond, VA 23235

Copyright © 2000–2005 by Richard Berry and James Burnell  
Second English Edition

All rights reserved. Except for brief passages quoted in a review, no part of this may be reproduced by any mechanical, photographic, or electronic process, nor may it be stored in any information retrieval system, transmitted, or otherwise copied for public or private use, without written permission of the publisher. Requests for permission or further information should be addressed to permissions Department, Willmann-Bell, Inc., P.O. Box 35025, Richmond, VA 23235.

Printed in the United States of America

Library of Congress Cataloging-in-Publication Data.

Berry, Richard, 1946-

The Handbook of Astronomical Image Processing / by Richard Berry and James Burnell.-- 2nd English ed.

p. cm.

Includes bibliographical references and index.

ISBN 0-943396-82-4

1. Imaging systems in astronomy--Data processing. 2. Astronomical photography. 3. CCD cameras. I. Burnell, James. II. Title

QB51.3.I45B46 2005

522.63'028--dc22

2004063730

The following are acknowledged as trademarks: Microsoft, Windows, Visual Basic, Visual C++, SBIG, TeleVue, Texas Instruments, Starlight Xpress, Kodak, Zip, Canon, Nikon.

05 06 07 08 09 10 9 8 7 6 5 4 3 2 1

## **Dedications**

This edition to the memory of my brother,  
Martin Bruce Berry, 1952–1997.

*Richard Berry*

To my wife Denise, and my daughters Megan and Katie, who have  
put up with three years of nights, weekends, and vacations  
with Daddy working at the computer.

*James Burnell*

---

# Table Of Contents

---

Preface to the First Edition .....	xvii
Preface to the Second Edition .....	xxiii
<b>1 Basic Imaging.....</b>	<b>1</b>
1.1 Light .....	1
1.2 Image Formation .....	2
1.2.1 Pinhole Imaging .....	2
1.2.2 Lens Cameras .....	4
1.2.3 Telescopes .....	6
1.3 Detectors .....	9
1.3.1 The Human Retina .....	9
1.3.2 Photographic Emulsions.....	11
1.3.3 Electronic Detectors .....	14
1.3.4 Linearity, Saturation, and Blooming .....	21
1.4 Sensor Geometry.....	21
1.4.1 Aspect Ratio .....	23
1.4.2 Pixel Count.....	24
1.5 Image Capture .....	25
1.5.1 Angular Field of View of a Detector .....	26
1.5.2 Sampling the Image .....	27
1.5.3 Angular Size of a Single Pixel .....	29
1.5.4 Matching Pixels to the Point-Spread Function.....	29
<b>2 Counting Photons.....</b>	<b>33</b>
2.1 What Is a Signal? .....	33
2.2 What Is Noise? .....	33
2.3 Signals and Noise .....	35
2.3.1 The Poisson and Gaussian Distributions .....	38
2.3.2 Collecting More Photons by Image Summing .....	39
2.3.3 Measuring Noise.....	40
2.3.4 Signals Become More Complicated.....	41
2.3.5 Unwanted Signals and More Sources of Noise .....	42
2.4 Signals and Noise in Images .....	43
2.4.1 Signal and Noise in a Raw Image.....	44
2.4.2 Signal and Noise in a Dark Frame .....	47

## Table of Contents

2.4.3 Signal and Noise in a Dark-Subtracted Image . . . . .	47
2.4.4 The Effect of the Sky Background on Signal and Noise . . . . .	49
2.4.5 Signal and Noise in Multiple Averaged Images . . . . .	51
2.5 Signal and Noise Effects from Flat-Fielding . . . . .	53
2.6 A Little Sermon on Signals and Noise . . . . .	55
<b>3 Digital Image Formats . . . . .</b>	<b>57</b>
3.1 File Format Basics . . . . .	57
3.2 FITS: The Standard Format in Astronomy . . . . .	59
3.3 Overview of FITS . . . . .	59
3.4 The FITS Header . . . . .	61
3.4.1 Mandatory Keywords . . . . .	62
3.4.2 Array Value Keywords . . . . .	63
3.4.3 Observation Keywords . . . . .	64
3.4.4 Comment Keywords . . . . .	65
3.4.5 Header Value Formats . . . . .	65
3.5 The FITS Binary Data Array . . . . .	66
3.6 The FITS Tailer . . . . .	68
3.7 Nonconforming FITS Files . . . . .	68
3.7.1 Nonconforming Headers . . . . .	68
3.7.2 Nonconforming Data Arrays . . . . .	68
3.7.3 Uncoordinated FITS Headers and Data Arrays . . . . .	69
3.7.4 Reading Variant FITS Files . . . . .	69
3.8 TIFF: The Standard in the Graphic Arts . . . . .	70
3.9 BMP: Images for Windows . . . . .	72
3.10 AVI: Interleaved Audio/Video from Webcams . . . . .	73
3.11 JPEG: File Compression for the Internet . . . . .	73
3.12 RAW, NEF, and CRW: Proprietary Raw Images . . . . .	75
<b>4 Imaging Tools . . . . .</b>	<b>77</b>
4.1 Sensors and Optics . . . . .	77
4.1.1 Sensor Size and Field of View . . . . .	77
4.1.2 Pixel Size and Resolution . . . . .	81
4.1.3 Spectral Sensitivity . . . . .	83
4.2 Optics for Imaging . . . . .	84
4.3 Auxiliary Optics . . . . .	89
4.3.1 Reducing Focal Length . . . . .	89
4.3.2 Increasing Focal Length . . . . .	90
4.3.3 Correcting Field Curvature . . . . .	90
4.3.4 Correcting Coma . . . . .	91
4.4 Finding Celestial Objects . . . . .	91
4.4.1 Flip Mirrors, Finder Scopes, and Go-To Mounts . . . . .	92
4.5 Telescope Mountings . . . . .	93
4.5.1 Tracking Sensitivity . . . . .	93
4.5.2 Rate Errors and Periodic Errors . . . . .	94
4.5.3 Testing and Tuning a Clock Drive . . . . .	95

## Table of Contents

4.5.4 Periodic Error Correction (PEC) Drives . . . . .	97
4.6 Filters . . . . .	97
4.6.1 Filter Types . . . . .	98
4.6.2 Filters for H-Alpha and Other Emission Lines . . . . .	100
4.6.3 Light Pollution Rejection (LPR) Filters . . . . .	100
4.6.4 Blue-Block and Violet-Block Filters . . . . .	102
4.7 Recognizing and Correcting Equipment Problems . . . . .	102
4.7.1 Hot Spots . . . . .	103
4.7.2 Field Flooding . . . . .	104
4.7.3 Vignetting . . . . .	106
4.7.4 Dust Donuts . . . . .	107
4.8 Reaping the Benefits . . . . .	111
<b>5 Imaging Techniques . . . . .</b>	<b>113</b>
5.1 Accurate Polar Alignment . . . . .	113
5.2 Good Guiding . . . . .	116
5.2.1 Off-Axis Guiding . . . . .	116
5.2.2 Auxiliary Telescope Guiding . . . . .	117
5.2.3 Software Virtual Guiding . . . . .	117
5.3 Critical Focus . . . . .	118
5.3.1 The Focuser . . . . .	119
5.3.2 Focus Techniques . . . . .	119
5.3.3 Automated Focusing . . . . .	121
5.4 Correct Exposure . . . . .	121
5.5 Shooting Calibration Frames . . . . .	123
5.5.1 Basic, Standard, and Advanced Calibration . . . . .	123
5.5.2 Making Bias Frames . . . . .	124
5.5.3 Making Dark Frames . . . . .	124
5.5.4 Making Flat-Field Frames . . . . .	125
5.5.5 Defect Mapping . . . . .	128
5.6 Imaging with Digital SLR Cameras . . . . .	128
5.7 Deep-Sky Imaging . . . . .	130
5.7.1 Strategies for Deep-Sky Imaging . . . . .	130
5.7.1.1 Digital Snapshots . . . . .	130
5.7.1.2 Guided One-Shot Imaging . . . . .	132
5.7.1.3 Guided and Unguided Track-and-Stack Imaging . . . . .	132
5.7.2 Making Good Deep-Sky Images . . . . .	134
5.7.3 Imaging Deep-Sky Targets . . . . .	135
5.8 Lunar, Planetary, and Solar Imaging Techniques . . . . .	141
5.8.1 Obtaining Excellent Images . . . . .	142
5.8.2 Focal-Ratio Matching Optics . . . . .	143
5.8.2.1 Field of View . . . . .	145
5.8.3 Recording High-Resolution Images . . . . .	146
5.8.4 Making High-Resolution Images . . . . .	147
5.8.5 Solar System Targets . . . . .	151

## Table of Contents

5.9 The Role of “Technique” . . . . .	155
<b>6 Image Calibration . . . . .</b>	<b>157</b>
6.1 What’s in a CCD Image? . . . . .	159
6.1.1 Photon Flux . . . . .	159
6.1.2 Nonuniformity . . . . .	161
6.1.3 Dark Current . . . . .	162
6.1.4 Zero-Point Bias . . . . .	163
6.1.5 Quantization . . . . .	164
6.1.6 Calibration is Peeling the Image Onion . . . . .	164
6.2 Calibration Frames . . . . .	165
6.2.1 Bias Frames . . . . .	165
6.2.1.1 Using a Single Bias Value . . . . .	167
6.2.1.2 Bias with Drift-Subtraction . . . . .	168
6.2.1.3 When to Make a Master Bias Frame . . . . .	169
6.2.2 Dark Frames . . . . .	170
6.2.2.1 “Image-Times-Five” Rule for Dark Frames . . . . .	171
6.2.2.2 Thermal Frames . . . . .	173
6.2.2.3 Standard and Scalable Dark Frames . . . . .	174
6.2.2.4 Dark Frame Matching . . . . .	176
6.2.2.5 Changing CCD Temperature . . . . .	177
6.2.2.6 When to Use a Single Dark Value . . . . .	178
6.2.2.7 Cosmic Ray Events . . . . .	178
6.2.2.8 Electroluminescence . . . . .	178
6.2.2.9 How to Make Master Dark Frames . . . . .	179
6.2.3 Flat Frames . . . . .	180
6.2.3.1 Four Types of Flat-Field Frames . . . . .	184
6.2.3.2 How to Shoot Light-Box and Dome Flats . . . . .	184
6.2.3.3 How to Shoot Twilight Flats . . . . .	186
6.2.3.4 How to Shoot Sky Flats . . . . .	186
6.3 Methods of Calibration . . . . .	187
6.3.1 Basic Image Calibration . . . . .	187
6.3.2 Standard Image Calibration . . . . .	189
6.3.3 Advanced Image Calibration . . . . .	191
6.4 The Calibrated Image . . . . .	193
6.4.1 Photons, Dark Current, and Readout Noise . . . . .	193
6.4.2 Noise from Calibration . . . . .	196
6.4.3 Image Stacking . . . . .	199
6.4.4 How to Spot Calibration Errors . . . . .	199
6.4.4.1 Bias Drift . . . . .	199
6.4.4.2 Changing CCD Temperature . . . . .	201
6.4.4.3 Changing Optical Configuration . . . . .	202
6.5 Defect Mapping and Correction . . . . .	203
<b>7 Image Analysis . . . . .</b>	<b>205</b>
7.1 Pixel Measurements . . . . .	205

## Table of Contents

7.1.1 Pixel Coordinates . . . . .	206
7.1.2 Pixel Value . . . . .	207
7.2 Whole-Image Analysis . . . . .	208
7.2.1 Image Statistics . . . . .	208
7.2.1.1 Minimum Pixel Value . . . . .	208
7.2.1.2 Maximum Pixel Value . . . . .	208
7.2.1.3 Mean Pixel Value . . . . .	209
7.2.1.4 Median Pixel Value . . . . .	209
7.2.1.5 Standard Deviation . . . . .	209
7.2.1.6 Low-Point and High-Point Pixel Values . . . . .	210
7.3 The Image Histogram . . . . .	210
7.4 Image Feature Analysis . . . . .	212
7.4.1 Pixel Statistics . . . . .	212
7.4.1.1 Defining a Region of Interest . . . . .	212
7.4.1.2 Minimum Pixel Value . . . . .	213
7.4.1.3 Maximum Pixel Value . . . . .	214
7.4.1.4 Mean Pixel Value . . . . .	214
7.4.1.5 Variance . . . . .	214
7.4.1.6 Standard Deviation . . . . .	214
7.4.1.7 Signal-to-Noise Ratio . . . . .	215
7.4.1.8 Median Pixel Value . . . . .	216
7.4.1.9 Mean of Median Half . . . . .	216
7.5 Determining a Centroid . . . . .	216
7.6 Distance on a CCD Image . . . . .	219
7.7 Image Profiles . . . . .	220
7.8 Astrometry . . . . .	221
7.9 Photometry . . . . .	221
7.9.1 Defining the Star Image . . . . .	221
7.9.2 Photometric Image Profile . . . . .	223
7.10 Spectroscopy . . . . .	225
<b>8 Measuring CCD Performance . . . . .</b>	<b>227</b>
8.1 Goals in CCD Testing . . . . .	227
8.2 Basic CCD Testing . . . . .	229
8.2.1 How to Make Basic CCD Test Images . . . . .	229
8.2.2 Basic Test Analysis . . . . .	229
8.2.2.1 Step 1: Mean and Standard Deviation in the Bias . . . . .	230
8.2.2.2 Step 2: Mean and Standard Deviation of the Flats . . . . .	230
8.2.2.3 Step 3: Measure the Dark Current . . . . .	231
8.2.2.4 Step 4: Compute the Conversion Factor . . . . .	231
8.2.2.5 Step 5: Compute the Readout Noise . . . . .	232
8.2.2.6 Step 6: Compute the Dark Current . . . . .	232
8.2.3 What Results to Expect . . . . .	232
8.3 Advanced CCD Testing . . . . .	234
8.3.1 Constructing a Low-Light Level Source . . . . .	234

## Table of Contents

8.3.2 Setting the Low-Level Light Source for Use .....	238
8.3.3 How to Make Advanced CCD Test Images .....	238
8.3.3.1 Make the Bias Frame Set .....	239
8.3.3.2 Make the Skim Frame Set .....	239
8.3.3.3 Make the Flat Frame Set .....	239
8.3.3.4 Make the Dark Frame Set .....	240
8.3.4 Analyzing the Advanced Test Image Set .....	240
8.3.4.1 Check the Bias Frames for Noise and Interference .....	241
8.3.4.2 Charge Skimming Check .....	243
8.3.4.3 Plotting the Transfer Curve .....	244
8.3.4.4 Checking the Linearity of the CCD .....	247
8.3.4.5 Determining the Dark Current .....	247
8.3.4.6 Check the Uniformity of the CCD .....	248
<b>9 Astrometry .....</b>	<b>249</b>
9.1 Astrometric Catalogs and Coordinates .....	249
9.1.1 The Astrometric Dilemma .....	250
9.1.2 Astrometric Catalogs and Reference Frames .....	251
9.2 Astrometric Theory .....	252
9.2.1 Standard Coordinates .....	253
9.2.2 Plate Coordinates .....	254
9.2.3 Improved Plate Constants .....	257
9.2.4 Solving for Position .....	257
9.3 Practical Astrometry .....	258
9.3.1 CCD Images for Astrometry .....	258
9.3.2 Scanned Photographs for Astrometry .....	260
9.3.3 Making Astrometric Measurements .....	263
9.4 Applied Astrometry .....	264
9.4.1 Astrometry of Newly-Discovered Objects .....	264
9.4.2 Astrometry of Asteroids and Comets .....	265
9.4.3 Using Astrometry to Identify Objects .....	266
9.4.4 Image Scale and Orientation .....	266
9.4.5 Astrometry in Education .....	267
<b>10 Photometry .....</b>	<b>271</b>
10.1 Magnitudes: How Bright Is This Star? .....	271
10.1.1 Magnitudes Are Comparisons .....	272
10.1.2 Aperture Photometry .....	272
10.1.2.1 Summing the Star's Light .....	273
10.1.2.2 Subtracting Sky Background .....	274
10.1.2.3 Raw Instrumental Magnitude .....	275
10.1.2.4 Statistical Uncertainty .....	276
10.2 Putting Photometry to Work .....	278
10.3 Photometric Systems .....	279
10.3.1 From CCD Images to the Standard System .....	286
10.3.2 Atmospheric Extinction .....	286

## Table of Contents

10.3.3 Transformation to the UBV(RI) System . . . . .	291
10.4 Photometric Observing . . . . .	292
10.4.1 Preparing to Observe . . . . .	292
10.4.2 Differential Photometry . . . . .	293
10.4.3 All-Sky Photometry . . . . .	296
10.4.4 “Do-What-You-Can” Photometry . . . . .	298
10.5 <i>Desiderata</i> for Photometry . . . . .	300
10.6 Don’t Be Afraid to Try! . . . . .	301
<b>11 Spectroscopy . . . . .</b>	<b>303</b>
11.1 What is Spectroscopy? . . . . .	303
11.2 Spectra and Spectrographs . . . . .	304
11.2.1 Prisms and Gratings . . . . .	304
11.2.2 Practical Spectrographs . . . . .	305
11.2.3 The Objective Prism Spectrograph . . . . .	305
11.2.4 The Grating-Prism Spectrograph . . . . .	307
11.2.5 The Slit Spectrograph . . . . .	309
11.2.6 The Fiber-Fed Spectrograph . . . . .	310
11.3 Properties of Spectrum Images . . . . .	311
11.4 Extracting a Spectrum from an Image . . . . .	313
11.4.1 Spectra from Objective Prism Images . . . . .	314
11.4.2 Spectra from Slit Spectrographs . . . . .	316
11.4.3 Spectra from Fiber-Fed Spectrographs . . . . .	318
11.5 Spectrum Calibration and Analysis . . . . .	319
<b>12 Geometric Transforms . . . . .</b>	<b>321</b>
12.1 Translation . . . . .	321
12.2 Rotation . . . . .	325
12.3 Scaling . . . . .	328
12.4 Practical Translation, Rotation, and Scaling . . . . .	329
12.5 Flipping and Flopping . . . . .	331
12.6 Cropping and Floating . . . . .	332
12.7 Resampling . . . . .	334
<b>13 Point Operations . . . . .</b>	<b>337</b>
13.1 Point Operations: An Overview . . . . .	337
13.2 Remapping Pixel Values . . . . .	339
13.2.1 Isolating the Range . . . . .	340
13.2.2 Transfer Functions . . . . .	343
13.2.2.1 Linear Transfer Function . . . . .	347
13.2.2.2 Gamma Transfer Function . . . . .	349
13.2.2.3 The Logarithmic Transfer Function . . . . .	350
13.2.2.4 The Gammalog Transfer Function . . . . .	351
13.2.2.5 The Negative Transfer Function . . . . .	352
13.2.2.6 The Sawtooth and Quantize Transfer Functions . . . . .	353
13.3 Direct Specification of the Transfer Function . . . . .	353

## Table of Contents

13.4 Direct Endpoint Specification . . . . .	355
13.4.1 Stretch Scaling . . . . .	356
13.4.2 Nonlinear Stretch Scaling . . . . .	357
13.5 Histogram Endpoint Specification . . . . .	357
13.6 Histogram Specification . . . . .	358
13.6.1 Histogram Equalization . . . . .	362
13.6.2 Gaussian Histogram Shaping . . . . .	362
13.6.3 Exponential Histogram Shaping . . . . .	363
<b>14 Linear Operators . . . . .</b>	<b>365</b>
14.1 Convolution in One Dimension . . . . .	365
14.1.1 One-Dimensional Examples . . . . .	369
14.1.2 Multiple Convolutions with One-Dimensional Kernels . . . . .	371
14.2 Convolution in Two Dimensions . . . . .	372
14.2.1 Convolution using Kernels . . . . .	373
14.2.2 Properties of the Convolution Kernel . . . . .	375
14.2.3 Kernels as Spatial Filters . . . . .	377
14.2.3.1 Smoothing Kernels (Low-Pass Filters) . . . . .	378
14.2.3.2 Sharpening Kernels (High-Pass Filters) . . . . .	381
14.2.4 Edge-Detection and Gradient Kernels . . . . .	385
14.2.4.1 Bas-Relief Operators . . . . .	385
14.2.4.2 Sobel, Kirsch, and Prewitt Operators . . . . .	386
14.2.4.3 Line-Detection Operators . . . . .	387
14.2.4.4 The Laplacian Operator . . . . .	388
14.3 Convolution by Unsharp Mask . . . . .	388
14.4 Generated Kernels for Unsharp Masking . . . . .	390
14.4.1 The Boxcar Unsharp Mask . . . . .	390
14.4.2 The Triangular Unsharp Mask . . . . .	391
14.4.3 The Gaussian Unsharp Mask . . . . .	392
14.4.4 The Power-Law Unsharp Mask . . . . .	393
14.4.5 Other Unsharp Masks . . . . .	394
<b>15 Non-Linear Operators . . . . .</b>	<b>397</b>
15.1 Rank Operators . . . . .	398
15.1.1 Minimum and Maximum . . . . .	398
15.1.2 The Median Operator . . . . .	399
15.1.3 The Rank-Order Operator . . . . .	401
15.1.4 The Multiplicative Rank Processes . . . . .	403
15.2 Non-Linear Enhancement Operators . . . . .	405
15.2.1 The Extreme Value Operator . . . . .	406
15.2.2 Local Adaptive Sharpening . . . . .	408
15.3 Noise Filters . . . . .	410
15.4 Morphological Operators . . . . .	412
15.4.1 Isophote Lines . . . . .	413
15.4.2 Frei and Chen Operators . . . . .	415
15.4.3 The Skeleton Operator . . . . .	418

## Table of Contents

15.4.4 Dilation, Erosion, Opening, and Closing Operators . . . . .	419
15.5 The Topographic Operator . . . . .	419
15.6 Digital Development . . . . .	422
<b>16 Image Operations . . . . .</b>	<b>425</b>
16.1 Image Math . . . . .	425
16.1.1 Addition . . . . .	425
16.1.2 Subtraction . . . . .	426
16.1.3 Multiplication . . . . .	428
16.1.4 Division. . . . .	429
16.1.5 Absolute Difference . . . . .	429
16.1.6 Merge . . . . .	430
16.1.7 Average. . . . .	432
16.2 Image Ranking. . . . .	433
16.2.1 Median Ranking . . . . .	433
16.2.2 Minimum Ranking . . . . .	435
16.2.3 Maximum Ranking. . . . .	435
16.3 Calibrating Images. . . . .	436
16.3.1 Dark Subtraction. . . . .	436
16.3.2 Flat-Fielding . . . . .	438
16.3.3 Basic Calibration . . . . .	440
16.3.4 Standard Calibration . . . . .	441
16.3.5 Advanced Calibration. . . . .	441
16.4 Image Registration. . . . .	442
16.4.1 Registration with Translation Only . . . . .	443
16.4.2 Registration with Translation, Rotation, and Scaling. . . . .	444
16.5 Blinking Images. . . . .	447
16.6 Track-and-Stack Image Summing and Averaging . . . . .	447
<b>17 Images in Frequency Space . . . . .</b>	<b>453</b>
17.1 Exploring Frequency Space. . . . .	453
17.1.1 Spatial Frequency . . . . .	453
17.1.2 The Frequency Spectrum . . . . .	454
17.1.3 Sinusoid Basics. . . . .	455
17.2 Fourier Theory. . . . .	456
17.2.1 Periodic Functions: The Fourier Series . . . . .	456
17.2.2 Nonperiodic Functions: The Fourier Integral. . . . .	458
17.2.3 Properties of the Fourier Transform . . . . .	462
17.2.4 Convolution via Fourier Transform . . . . .	464
17.2.5 Parseval's Theorem . . . . .	466
17.2.6 The Discrete Fourier Series . . . . .	466
17.2.7 The Fast Fourier Transform . . . . .	469
17.3 Image Processing using the Fourier Transform. . . . .	470
17.3.1 Butterworth Spatial Frequency Filters . . . . .	472
17.3.2 Feature Masking in Frequency Space. . . . .	474
17.3.3 Feature Enhancement in Frequency Space. . . . .	475

## Table of Contents

<b>18 Wavelets .....</b>	<b>477</b>
18.1 The Wavelet Transform .....	477
18.1.1 The Wavelet Function .....	479
18.1.2 Properties of the À Trous Wavelet Transform .....	479
18.1.3 Spatial Filtering with the Wavelet Transform .....	483
18.1.4 Spatial Filtering with Star Images .....	485
18.2 Wavelet Noise Filters .....	485
18.2.1 When Is an Image Feature Significant? .....	486
18.2.2 Transforming Poisson Noise to a Gaussian Deviate .....	488
18.2.3 Measuring Noise in Images .....	489
18.2.4 Rejecting Noise and Retaining Significant Features .....	491
18.2.5 The Solution: An Iterative Wavelet Noise Filter .....	492
18.3 Wavelet K-Sigma Filtering .....	494
18.4 Using Wavelets .....	497
<b>19 Deconvolution.....</b>	<b>499</b>
19.1 The Inverse Convolution Problem.....	500
19.2 Image Estimation by Iteration .....	501
19.3 Van Cittert Image Estimation .....	506
19.4 Richardson-Lucy Image Estimation .....	507
19.5 Using Deconvolution with Astronomical Images .....	509
<b>20 Building Color Images.....</b>	<b>513</b>
20.1 Human Color Vision .....	514
20.1.1 The Trichromatic Basis of Color Vision .....	514
20.1.2 Luminance and Chrominance .....	514
20.1.3 Reproducing Color .....	516
20.2 Celestial Light .....	516
20.2.1 Starlight and Continuous Spectra .....	519
20.2.2 Nebulae and Emission Spectra .....	521
20.2.3 The Challenge of Celestial Color Imaging .....	522
20.3 Red/Green/Blue Tri-Color Imaging .....	523
20.3.1 Step 1: Capture Filtered Images .....	525
20.3.2 Step 2: Correct the Filtered Images .....	528
20.3.2.1 Subtract the Skylight Background .....	528
20.3.2.2 Color Balance with G2V Stars .....	531
20.3.3 Create the Color Image .....	534
20.3.4 Summary: White-Balance Using G2V Stars .....	535
20.4 Using Field Stars for White Balance .....	536
20.4.1 Summary: White-Balance Using Field Stars .....	537
20.5 Color Images from Filter-Matrix Cameras .....	537
20.6 Color Space: Geometric Interpretations of Color .....	539
20.6.1 RGB Color Space .....	539
20.6.2 HSL Color Space .....	541
20.6.3 Lab Color Space .....	544
20.7 Luminance/RGB (LRGB) Color Imaging .....	545

## Table of Contents

20.7.1 Creating an Artificial Luminance Image . . . . .	547
20.7.2 Enhancing the Luminance Image . . . . .	547
20.7.3 Creating LRGB Images . . . . .	547
20.8 Practical RGB and LRGB Color Imaging . . . . .	549
20.8.1 Selecting Filters . . . . .	549
20.8.2 Shooting RGB and LRGB Images . . . . .	550
20.8.3 Trade-offs: RGB versus LRGB . . . . .	552
20.9 LLRGB (Multi-Luminance) Imaging . . . . .	552
20.10 Extended-Range and Narrowband Color . . . . .	553
20.11 Color Imaging with CMY Filters . . . . .	554
20.11.1 Selecting CMY Filters . . . . .	555
20.11.2 CMY versus RGB Imaging . . . . .	555
20.12 The Subjective Side of Color Images . . . . .	555
<b>21 Processing Color Images . . . . .</b>	<b>557</b>
21.1 Properties of Color Images . . . . .	557
21.1 Calibrate to Remove Dark Current and Vignetting . . . . .	564
21.1.1 Calibrating Bayer-Array Color Images . . . . .	564
21.1.2 Calibration for Digital Cameras . . . . .	565
21.1 Stacking to Enhance Signal-To-Noise Ratio . . . . .	567
21.2 Making Great Images with a Digital Camera . . . . .	570
21.3 Color Images in Color Spaces . . . . .	572
21.3.1 RGB Color Space . . . . .	573
21.0.1 HSL Color Space . . . . .	574
21.0.1 Lab Color Space . . . . .	576
21.1 Color in Color Images . . . . .	576
21.1.1 White Balance . . . . .	576
21.1 Luminance Enhancement of Color Images . . . . .	578
<b>Appendix A Glossary . . . . .</b>	<b>581</b>
<b>Appendix B Resources . . . . .</b>	<b>597</b>
B.1 CCD Imaging . . . . .	597
B.1.1 CCD Imaging Books . . . . .	597
B.2 Astronomical Optics . . . . .	598
B.2.1 Telescope Design . . . . .	598
B.2.2 Telescopes and Vision . . . . .	599
B.3 Astrometry . . . . .	599
B.3.1 Astrometry Books and Articles . . . . .	599
B.3.2 Astrometry Software . . . . .	600
B.3.3 Astrometry Reference Catalogs . . . . .	601
B.4 Photometry . . . . .	601
B.4.1 Photometry Books and Articles . . . . .	601
B.4.2 Photometric Data: Extinction Stars and Standard Stars . . . . .	602
B.4.3 Photometry Organizations . . . . .	602
B.4.4 Photometric Filters . . . . .	603

## Table of Contents

B.4.5 Photometry Software .....	603
B.5 Spectroscopy.....	603
B.5.1 Spectroscopy Books and Articles .....	603
B.5.2 Spectroscopic Catalogs .....	604
B.6 Image Processing .....	604
B.6.1 Image Processing Books: General .....	604
B.6.2 Image Processing: Science Applications .....	605
B.6.3 Image Processing: Image Restoration .....	606
B.6.4 Image Processing and the Fourier Transform .....	607
B.6.5 Image Processing and Wavelets.....	607
B.6.6 Image Processing Algorithms .....	607
B.6.7 Image Processing Software .....	608
B.7 Color Imaging.....	610
B.7.1 Color Imaging Books.....	610
<b>Appendix C Tutorials .....</b>	<b>613</b>
C.1 Basic Skills .....	616
C.2 Calibration .....	621
C.3 Image Evaluation .....	627
C.4 Astrometry .....	631
C.5 Photometry .....	635
C.5.1 Single-Star Photometry .....	636
C.5.2 Single-Image Photometry .....	638
C.5.3 Multiple-Image Photometry.....	639
C.6 Spectroscopy.....	642
C.7 Image Enhancement .....	644
C.7.1 Brightness Scaling .....	644
C.7.2 Histogram Shaping.....	647
C.7.3 Convolution Filtering.....	649
C.7.4 Unsharp Masking.....	650
C.7.5 Deconvolution .....	650
C.7.6 Wavelet Spatial Filtering .....	653
C.7.7 Morphological Processing .....	654
C.8 Fast Fourier Transform .....	655
C.9 Multiple Image Processing .....	659
C.9.1 Track and Stack .....	659
C.9.2 Multi-Image Processing .....	663
C.9.3 Multi-Image Alignment .....	663
C.10 Image Registration and Blinking .....	663
C.11 Deep-Sky Images .....	665
C.12 Planetary Images.....	670
C.13 Color Images.....	672
C.14 Conclusion .....	674
<b>Index.....</b>	<b>675</b>

---

# Preface to the First Edition

---

CCD cameras have brought a vast new wealth to astronomy. Amateur astronomers and students now undertake observing projects that would have been unimaginable a decade ago: resolving sub-arcsecond details on the planets, recording 21st magnitude stars, measuring the motions of nearby stars, uncovering scores of new asteroids, charting the ups and downs of variable stars, and finding supernovae by the dozen. And a vast wealth of magnificent planet, nebula, and galaxy images resides on web sites around the world. No longer is the taking of high-quality data restricted to professional astronomers.

## Richard's Remarks

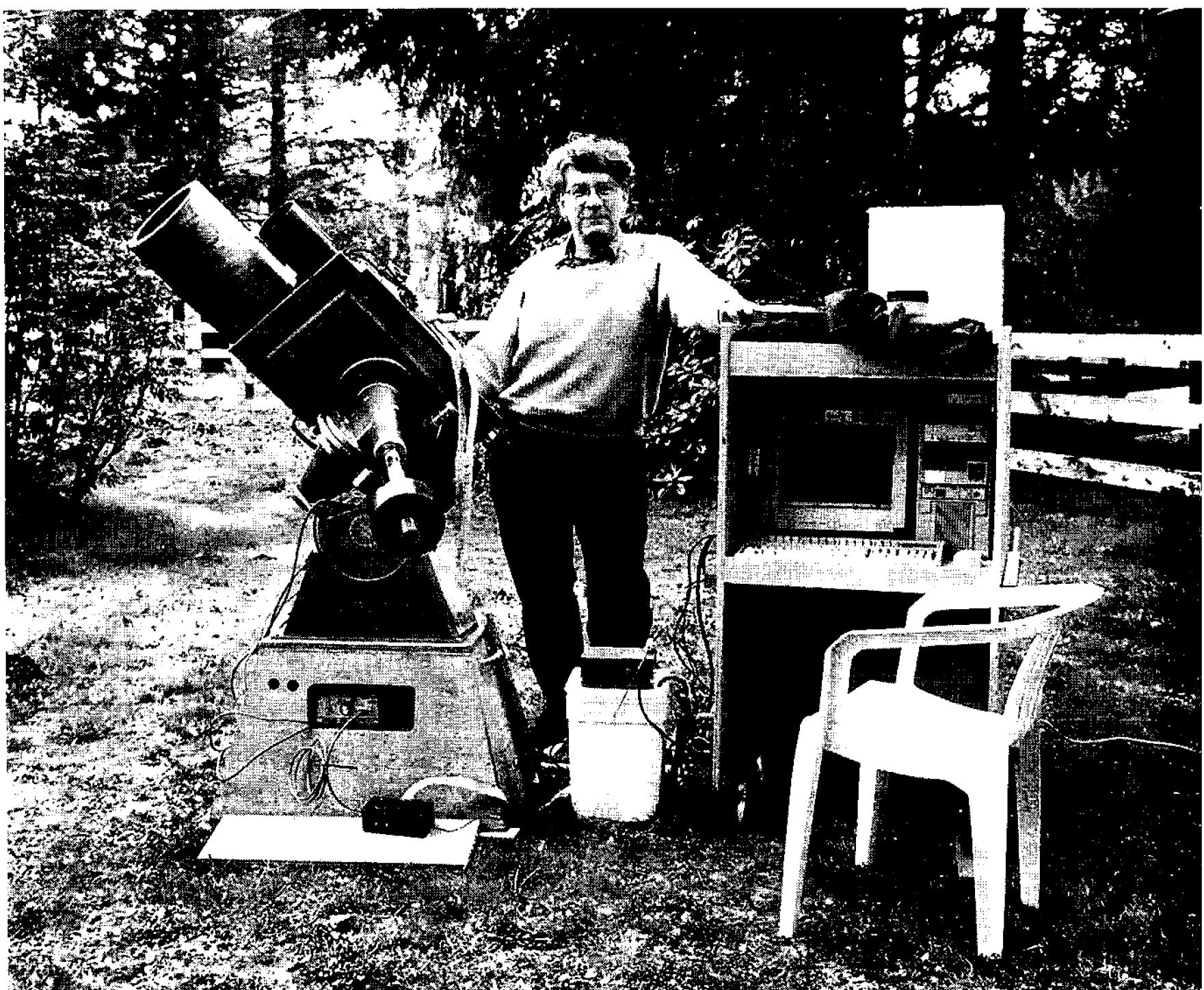
Nine years have passed since *Introduction to Astronomical Image Processing*, the forerunner of this book, appeared. In its preface, I wrote:

“Back in 1965, Mariner 4 sent the first close-up pictures of Mars back to Earth. I was a teenager, and terrifically impressed that pictures could be radioed to Earth from a planet millions of miles away. And, despite the fact that Mariner 4’s camera didn’t work quite right, over a period of months, scientists at the Jet Propulsion Laboratory discovered that Mars is a cratered world because they computer-processed the images. It seemed almost magical. I longed to get my hands on those images and to learn how to manipulate the brightnesses and contrasts of the originals until the forms of the craters emerged from the murk.”

Today, my dream has come true—and that is why I wanted to write this book. This is a book about images and manipulating images—extracting data, seeing the unseen, enhancing detail—until the raw information collected at the telescope has yielded everything there is to see and know. I wanted this book to be a behind-the-scenes look at image processing; not another “drag this and click that” computer book, but an in-depth analysis and exploration of how image processing works. I wanted a book that wasn’t afraid to dig into the math and show you the algorithms that image processing software uses to measure and manipulate images.

Why? Because digital imaging has transformed astronomy. To make the most of CCD imaging, you need to have some idea of what’s going on inside the camera and inside the software. Just as carpenters must know the tools they use,

## Preface to the First Edition



so must observers understand the electronic and software tools they use.

Today a single amateur astronomer has more computer power sitting on the desk than JPL's mainframes had during the first Voyager encounter with Jupiter. A typical amateur astronomer's CCD outperforms the most advanced CCDs professional astronomers had on the 5-meter Hale telescope on Palomar Mountain fifteen years ago. And the skills to measure and manipulate images are open to everyone who seriously wants to learn them.

For me, writing the *Introduction to Astronomical Image Processing* was a splendid opportunity to learn the basics of image processing. The book included a tidy little processing program—*AIP*, alias *AstroIP*—that I had written to accompany the book. The earliest versions of the software ran under a BASIC interpreter on an 8MHz PC XT computer with a CGA graphics card. It was so slow that it took ten seconds to draw an image on the screen. But it was a deep thrill for me—I was writing image processing software!

The year after *Introduction to Astronomical Image Processing* was published, Veikko Kanto, an amateur astronomer from Arizona, built a CCD-based autoguider for astrophotography. When he saw my book he changed the guider to take images—and that camera became the prototype of the Cookbook 211 camera.

Over the next two years, Veikko and I, with John Munger, wrote *The CCD Camera Cookbook* for two CCD chips: the TC211 and the TC245. We put a lot of work into simplifying the design so that anyone willing to acquire basic soldering

skills and test their work carefully could build an inexpensive CCD camera. Since then, thousands of amateur astronomers have built Cookbook cameras and set them to work making wonderful images.

With my own Cookbook camera, I began to explore CCD imaging and image processing more deeply. I carried out experiments to determine how best to use my camera, and I developed software using *QuickBasic 4.5*, to expand what the camera could do. Techniques like multiple dark frames, dark-frame matching, track-and-stack imaging, histogram shaping, and color imaging transformed the Cookbook camera into a remarkably powerful imaging tool. The resulting software suite—*BatchPIX*, *CB245*, *Multi245*, and *QColor*—soon attracted users all around the world.

The summer after the *Cookbook* came out, I set up my computer and Cookbook 245 on the south side of the little pink clubhouse at Stellafane. Lots of people visited and stayed to talk for a few minutes, but one guy stayed and talked for hours. He asked everything about the camera. It got so late, in fact, that his little daughter curled up and fell asleep at our feet. The visitor was Jim Burnell.

Jim went home, built a Cookbook 245 camera; and then to help other members of his astronomy club, he ran classes on building Cookbook cameras. As an electrical engineer, Jim had a solid background in programming, and developed a nifty software package—*Prep245*—for calibrating and displaying images.

Meanwhile, I was trying to convert my programs from QB4.5 to the Windows platform, with very little luck. Those who are not programmers cannot imagine how unfriendly programming for Windows really is. I needed help. Twice I found partners who knew Windows, but the first became ill and the second got a new job and had to drop out of the project. My DOS software was getting more sophisticated—I had astrometry, photometry, FFTs, and Richardson-Lucy deconvolution running under DOS—but DOS could not access enough memory, and it was clear that PC users really wanted graphic-user-interface software.

At that point, I just about gave up. Displaying good-looking grayscale images in Windows seemed impossible—and until I could figure out how to display an image, it wouldn’t do me any good to port my software tools to Windows. Meanwhile, Jim had already written *Prep245*, a solid starting point for a bigger, more ambitious project. We exchanged e-mails—and soon hatched our plan to write **AIP4Win** and the book you now hold in your hands.

Jim and I have devoted nearly three years to the *Handbook of Astronomical Image Processing* and **AIP4Win**. We see the book and software as two complementary parts of a package for learning about imaging and image processing. The book provides background and theory, and the software puts powerful image processing tools at your fingertips. The book is not a manual for our **AIP4Win** program—you’ll find that in the extensive Help file—but an exploration of the measuring tools and enhancement algorithms common to all image-processing software, whatever software package you happen to be running.

## Jim's Remarks

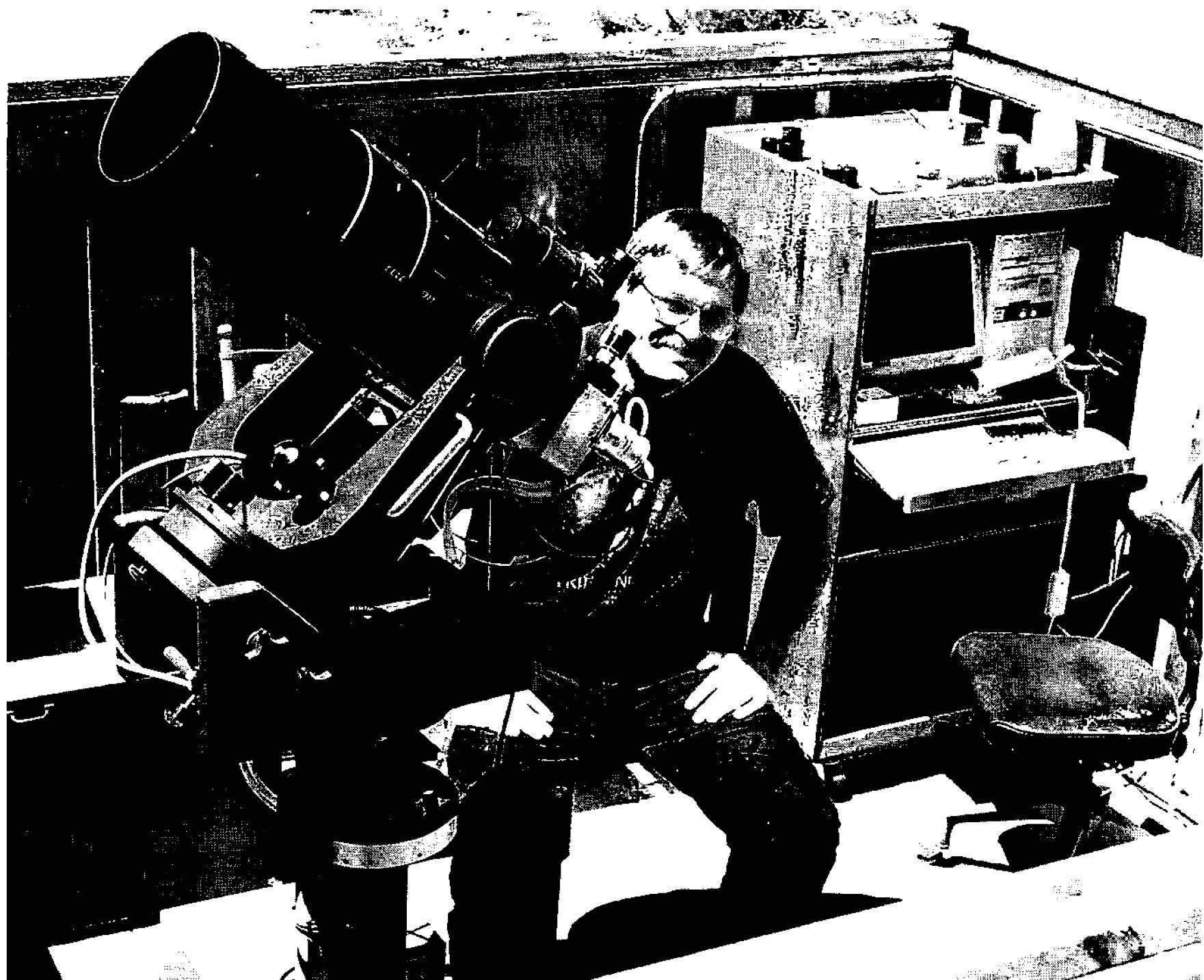
I first became seriously interested in image processing in the late '80s. I was working at Bell Labs at the time and had access to a lot of interesting people and computer resources. I remember playing with the Voyager images as they became available. During lunch and after hours, my buddies and I would experiment to see what we could do with them. I used an image-processing tool called *IMDISP*, written by Ron Baalke, and available via an ftp site at NASA. A bunch of us used to pass the images around on the old USENET, and it was fascinating to see what details we could coax out of the amazing data coming back from Jupiter, Saturn, Uranus and Neptune. This was in the pre-Hubble days, and there was very little deep-sky imagery available over the internet at that time; when someone found a place where you could download images of nebulae and galaxies, it was usually very slow because of all the people downloading the images.

I used a UNIX mainframe to do most of my initial experimenting, as early PCs were just too slow and limited in memory capacity even to consider many of the more involved imaging tasks. It wasn't until the 486-based machines came out that I even bothered building a PC for use at home.

I first ran into Richard at Stellafane in the early '90s. At the time I was planning to build an observatory, and I happened to encounter Richard on the trail on the way to the Friday night tent talks and struck up a conversation on the subject. At the time Richard was editing *Telescope Making* as a side project to his main work on *Astronomy*. There had been some coverage of small observatories both in *TM* and *Astronomy*, and I had a few questions about the mechanics of building a roll-off roof. As we chatted, Richard mentioned a CCD camera project that he and some others were working on—something that he hoped would be made available as a kit. This really captured my interest, as I had been playing around with emulsion-based astrophotography and was frustrated by the long exposure times.

The following year my 9-year old daughter and I were at Stellafane for her very first time. Saturday night after the talks we wandered up to the pink clubhouse on Breezy Hill and there was Richard with the prototype for the Cookbook 245 CCD camera. He had a 50-mm camera lens attached to it and was taking wide-field images of the Milky Way with the rig mounted on a small camera tripod. I joined in and spent the next several hours helping to find objects while Richard demonstrated the operation of the camera. I was hooked.

The next spring my buddy Neil and I built our first Cookbook 245 CCD cameras. We started taking a lot of images and soon realized that we needed something more than the photo editing software that was currently available. I bought a copy of Richard's CB245 package to process my images, but I really yearned for a Windows-based package to do the job. I had been programming in the C language for the previous 16 years and decided to experiment with some of the new Windows-based development tools that were coming out. I wrote a package called *Prep245* using *Visual Basic* to create the GUI and using *Visual C++* to do the number crunching. I spent a few hours on the phone with Richard, from time to time, ask-



ing questions on image processing techniques. His book, *Introduction to Astronomical Image Processing*, had been a really helpful reference and got me started on a lot of the more advanced capabilities.

*Prep245* had been out for about two years, and I was thinking about making some upgrades to it. I started corresponding with Richard and floated the idea that we might collaborate on a Windows version of his popular DOS software to accompany an image processing book that he had been wanting to write. It has taken three years to complete the job, and the result you now hold in your hands.

This software has been written by people who use it regularly to calibrate and process their own CCD images. Our approach has been to analyze the way we work with our images and to write software that facilitates that work flow. Realizing that not everyone has the same work style, we recruited a number of serious amateur CCD imagers, each working on different imaging projects, who pounded on the program and gave the features a real workout. We owe a real debt of gratitude to them for the years of testing each and every feature, retesting each update as we added new functionality and fixed the inevitable bugs. The end result is an intuitive, easy-to-use package that provides a complete solution to just about any astronomical project one would undertake using a CCD camera, whether it is supernova hunting, generating light curves of variable stars, tracking the changing details of Jupiter's atmosphere, recording the spectrum of a planetary nebula, creating accurate color images, or taking pretty pictures of favorite deep-sky objects.

## Acknowledgments

We have used ***AIP4Win*** to process images during the program's whole development cycle. We have processed thousands of images, tested routines over and over, carried out astrometry, photometry, spectroscopy, shaped countless histograms, blinked for supernovae, and made CCD movies. All of the images in the book were prepared using ***AIP4Win***. But we could not have done it alone. We wish to acknowledge the generous help of many friends and associates.

Our beta test team deserves particular credit: Scott Berfield, Lew Cook, Allen Gilchrist, Veikko Kanto, Al Kelly, Errol Mustafa, Neil McMickle, Tim Puckett, Andy Saulietis, Chuck Shaw, Stuart Warmink, Jan Wisniewski, and Rob West. Tim, Neil, and Rob deserve special mention for demanding the most from us, and we are thankful for the opportunity to make ***AIP4Win*** meet their needs. William Greenlee, Jeff Gunn, Al Nagler, John Rogers, Arne Hendon, Brian Manning, Brian Skiff, and Harold Suiter read and commented on draft chapters; and all of the beta testers read and commented on the entire manuscript.

Special thanks go to Neil McMickle for endless hours spent testing and providing feedback, and for his perpetual readiness to try out new routines, acquire badly needed test images, and retest the program at each new update.

We are grateful to Jack Newton and Don Parker, who shared their ideas and practical imaging experience during the early development of Richard's DOS software, and to Dennis di Cicco for sharing images and ideas. Ed Grafton, Al Kelly, and Chuck Shaw were inspirational in the development of Richard's DOS software for RGB and CMY imaging, which evolved into the first version of the color tools. Brian Manning deserves special thanks for teaching us astrometry. Phil Kuebler got us going on photometry, and Gary Frey challenged us to try even harder. Finally, our thanks to Dave Monet of the U. S. Naval Observatory for generously providing us with a copy of the USNO A2.0 astrometric database.

We also want to acknowledge many other contributors whom we have not mentioned. We have been participants on the [ccd@wwa.com](mailto:ccd@wwa.com) listserve—over the years reading thousands of e-mails—and want to thank every member of the list. The roots of this book go back more than a decade, and it is certain that we have overlooked many who helped, advised, and encouraged us. Please accept our apologies—even though we have not mentioned you by name, we are deeply grateful for your help—please jog our memories so that we might correct future printings.

Finally, special thanks in advance to our readers. In the end, you are the most important of all to us. No lengthy book, and no large program, can ever be entirely free of errors. We welcome your comments on the book and your bug reports on the program. Please send them to us through Willmann-Bell so that we can properly log them and act on them. With your help, future editions of the book and software might even approach perfection.

RICHARD BERRY

JAMES BURNELL

---

# Preface to the Second Edition

---

We are proud to present a new edition of the *Handbook of Astronomical Image Processing* to our readers. As anyone who has participated in the rapidly evolving field of astronomical imaging already knows, everything has changed. New hardware and new software tools have emerged, computers are faster, and the Internet has become more inclusive. Yet underneath it all, the fundamentals remain the same. On clear nights we wait as a trickle of photons enters the telescope, strikes the detector, and accumulates into an image of a distant place in the cosmos.

## Jim's Remarks

A lot has happened in the four years since the first edition of this book was published. Astronomical CCDs have gotten much bigger, much quieter and much more sensitive. WebCams have come into their own as imaging devices for the budding, as well as serious, amateur astronomer to capture high-resolution pictures of the Moon and planets. Also digital single-lens reflex (DSLR) cameras have come way down in price and are rapidly becoming a very popular way for the beginner to “get his feet wet” in astroimaging as well as for the veteran to easily take wide-field images of the brighter deep-sky objects.

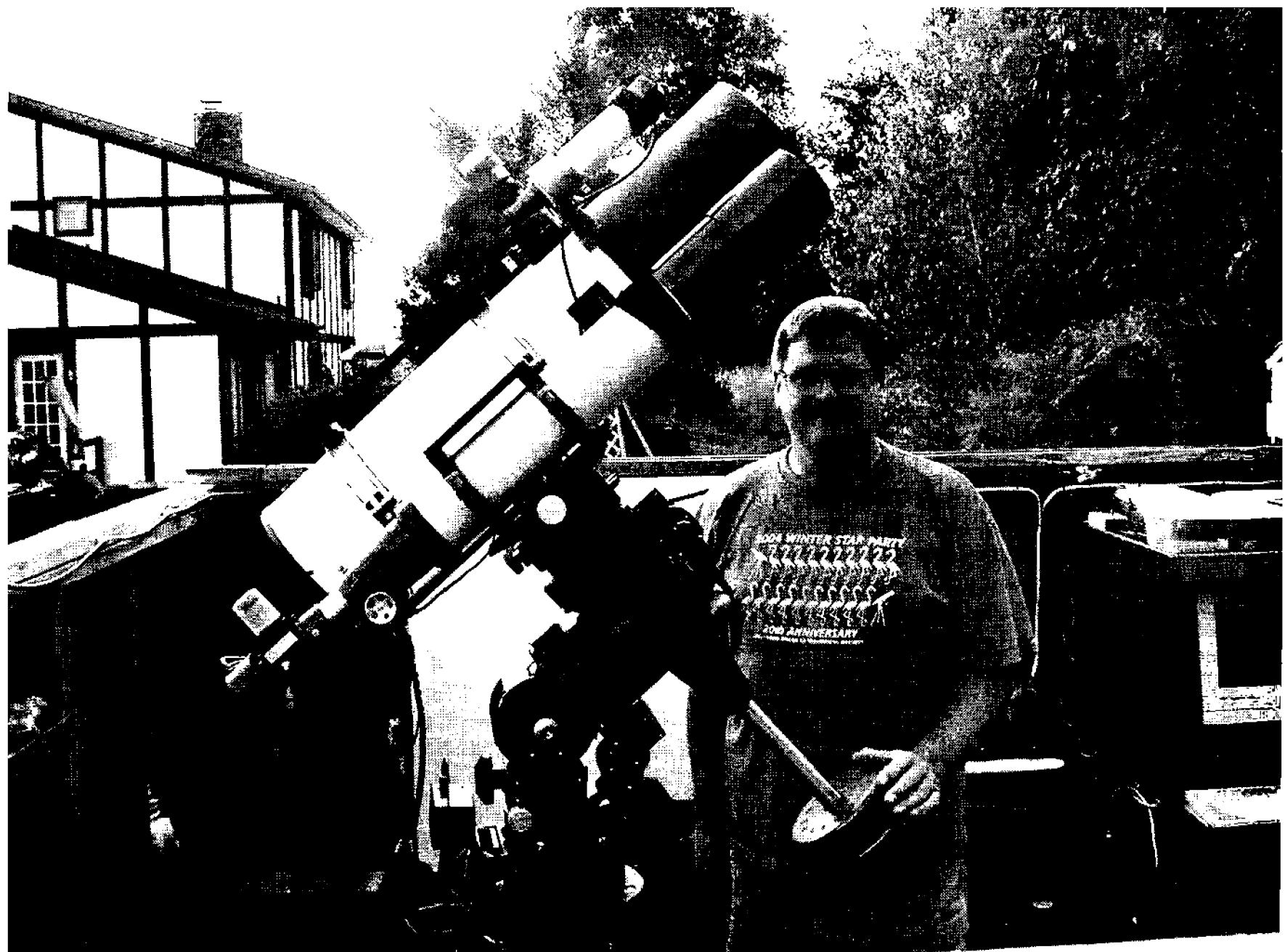
New, high-quality filters are now available at a reasonable cost, that allow us not only to separate the colors found in celestial objects in order to create color images, but also to isolate individual spectral lines. These filters make it possible to isolate fine detail that would otherwise be lost in the background light.

The use of autoguiders is now commonplace, with the result that many astroimagers engage in marathon sessions over multiple nights, creating images with many hours of accumulated exposure time. As a result, amateurs are going deeper than ever before in their quest for faint objects.

The imaging hardware is not the only thing that has changed. Computer hardware, following Moore’s Law, has steadily improved its capabilities to the point where an 80-gigabyte hard drive and 500 megabytes of RAM are found even on entry level systems. This gives the software developer much more freedom to implement numerically intensive image processing routines without the problem of them taking all night to execute.

The art of image processing has undergone continuous growth as well, with

## Preface to the Second Edition



Photograph by Kathleen Burnell.

new techniques, such as wavelets, that remove noise and noiselessly enhance the details in CCD images. New ways of dealing with color give the user more control than ever in producing color images from a CCD camera or DLSR.

With all the growth in the field of astronomical imaging, it was time to update the text and write new software to take advantage of the new equipment and techniques that have become available. As a result, significant portions of the text have been revised, and chapters added. New techniques and methods are covered along with the use of some of the new equipment that has become available.

The software, originally written with a 486 PC having 16MB of memory and an 800x600 display as a minimum system, has been rewritten to take advantage of the current crop of entry level PCs with multi-GHz CPUs and hundreds of mega-bytes of main memory. This new crop of PCs allows routines to be written which were just too computation-intensive to be implemented before. More experience in using the tools in the hands of so many talented imagers has provided tremendous feedback on the techniques and methods implemented in the first version of **AIP4Win**. These suggestions, provided by the user community, formed the basis for the feature list of the new software. While keeping as much of the original user interface as was feasible, this second version is an entirely different creature under the hood. Gone are the 32-bit integers that were needed in order to implement processing in a speedy fashion. The use of floating-point math on the current crop of fast machines has facilitated the implementation of more powerful image processing techniques.

On top of the original book and software, nearly a year of writing, coding and testing have gone into the product you now hold in your hands. Enjoy!

## Richard's Remarks

In the years since the publication of the *Handbook* and **AIP4Win**, we have seen three major groups of readers and users emerge:

- imagers who produce beautiful visions of the heavens,
- observers doing photometry, astrometry, and science, and
- educators teaching and students learning astronomy.

With the second edition of the *Handbook* and the release of **AIP4Win 2.0**, we are pledged to continue serving these constituencies. To that end, we have focused our energies in two major areas: noise reduction and color imaging.

We began our assault on noise by rewriting **AIP4Win** to use floating-point mathematics throughout to eliminate mathematical round-off error. Our old *AIP32* code did many computations using floating-point math, but it stored data in 32-bit integers. Although fine for most users, storing results in integer format produces round-off errors. Furthermore, floating-point math is more accurate (with a “granularity” of  $\sim$ 30 parts per billion) and far more flexible and powerful.

**AIP4Win 2.0** supports “astronomical” pixel values ranging from  $1 \times 10^{-32}$  to  $-1 \times 10^{32}$  and microscopic pixel values as small as  $1 \times 10^{-32}$ . This wide dynamic range means that you can sum images and easily work with pixel counts in the millions and billions, or average and work on images that express pixel values to three or four decimal places. Those decimals mean more dependable dark matching, more accurate flat-fielding, and more precise histogram shaping. In addition, **AIP4Win** fully supports floating-point FITS formats, so everything is saved and nothing lost when you save an image to disk. Every user will benefit from **AIP4Win**'s dedication to robust low-noise internal number-crunching.

In this new edition of the *Handbook*, we treat noise theory more fully than we did in the first. The current generation of CCDs performs so well that the random arrival times of photons—shot noise—is the primary limitation in their performance. In a new chapter titled “Counting Photons,” we explore what Poisson noise means to astronomy. Serious imagers, educators, and students need to be familiar with this fundamental noise source, and those interested in making beautiful images will understand more fully how and why the long integration times that capture millions of photons yield the most beautiful images.

I became fascinated with noise three summers ago, while teaching a course in digital imaging at Portland State University. One practical benefit for the observers, teachers, and students is that **AIP4Win**'s stellar photometry tools now perform a noise analysis based on the count of photons detected and pixels measured, all reported as a routine part of measuring stellar brightness. This analysis will alert observers to the precision they have attained in measuring stellar magnitude, and we trust that the feedback will assist them in making even more precise mea-

## Preface to the Second Edition



surements. For students and educators, it's an object lesson that noise statistics impose fundamental limits on what we can learn about the Universe.

Inevitably, however, one studies noise in search of ways to eliminate it. It was clear that spatial methods (such as pixel averaging) always blur image detail, and frequency methods fail because random noise is random in frequency space. However, a hybrid spatial-and-frequency analysis called the wavelet transform allows one to determine whether a pixel or cluster of pixels is probably real or whether it is just random. **AIP4Win**'s wavelet tools grew out of this, as I boiled theory into practice. Because they are totally noise free, wavelet-processed images look strange to noise-accustomed eyes—so smooth and silky—yet every detail of a noisy image is faithfully replicated. If you were raised on a diet of film grain and random noise, wavelet processed images may not look right to you; but teachers and students are sure to find wavelets a powerful tool for the analysis of severely noise-limited images.

For the image-makers, we developed a noise-averaging spatial filter called the Smooth Background Tool. This tool smooths noise by averaging pixels in the dark sky background parts of an image, but it leaves bright areas untouched. Instead of hiding noise by making sky backgrounds jet black, imagers can lighten the sky to show the faint outer parts of galaxies and nebulae that are lost when the sky is solid black.

Another feature of the new **AIP4Win** is a suite of tools for creating synthetic images. These are intended for the educators and students to explore image constituents—bias, dark current, vignetting, image, and noise. Using these tools, you

can build images with precisely known constituent parts and use them to explore the relationships between the sky background, stars, objects, and detector characteristics.

The other significant addition is enhanced coverage of color imaging. Five years ago, when we developed **AIP4Win**'s first set of color tools, color imaging was quite new. Use of spectral class G2V solar-analog stars to attain accurate color balance had just caught on, and luminance-overlay color (LRGB) was a cutting-edge technique. After a lot of reading, thinking, and experimenting, we took the somewhat scary step of processing color images with **AIP4Win** in a luminance/chrominance (LCh) color space rather than the traditional RGB color space. For astronomy, where luminance varies over a million-to-one range (black sky to brilliant star), the choice of LCh color space offers significant advantages over the limited scale of RGB. There are now two chapters in the *Handbook* devoted solely to color—and processing color images is now “native” to **AIP4Win**.

Three factors appear to be driving the widespread interest in color imaging: the availability of really good color-separation filters for monochrome CCD cameras, a new generation of high-quality Bayer-array CCD cameras, and the appearance of excellent digital single-lens reflex (DSLR) cameras. We developed new routines and procedures for loading, decoding, and displaying images from these cameras. I bought a Nikon D70 to learn the ins and outs of DSLR imaging and Jim and I have both experimented with a Canon 10D to get everything possible from the images they capture, in the processing making thousands of raw images, dark frames, and flat-field frames.

For the CCD imagers, we revised and updated **AIP4Win**'s Join Colors Tool. The software code behind this tool translates your filtered monochrome plus luminance images into color. Working in LCh, RGB, and Lab color spaces, we sought new ways to build images with bright, clean colors and crisp, clear detail. We also added a new Color Effects Tool that operates directly on color images from DSLR cameras *and* astronomical CCD cameras.

Finally, to prevent even the slightest loss of color information, **AIP4Win** can save color images in a special 96-bit FITS file (32-bit floating-point data in each color channel) as well as the traditional 48-bit TIFF format with 16-bits integer data in each color channel. If you save your images in this format, no information is lost, so you can always pick up and continue processing where you left off.

In addition to color tweaking, every one of **AIP4Win**'s tools can access and process the luminance component of an image *without* disturbing its color balance or saturation, and the suite of color effects tools can alter chrominance while leaving image brightness untouched. By cleanly splitting image chrominance from image luminance, **AIP4Win** allows you to apply deconvolutions, correct brightness gradients, fix uneven sky backgrounds, replace bad star images, smooth sky backgrounds, and otherwise process any image—color or black and white—with any tool.

## Acknowledgments

Once again we acknowledge the many contributors whom we have not mentioned individually. Shortly after the first edition went on sale, an independent Yahoo group sprang up to discuss **AIP4Win** under the leadership of Greg Crawford. Although we do not own or control it, [aip4win@yahoogroups.com](mailto:aip4win@yahoogroups.com) has nonetheless served as a rich source of ideas and inspiration to us. We wish to thank the members of that listserver for thousands of intriguing questions and stimulating discussions.

However, your *direct* line to **AIP4Win** with ideas and suggestions is through our publisher, Willmann-Bell. Send your email comments, suggestions, bug-reports, and requests to us at [aip4win@willbell.com](mailto:aip4win@willbell.com). A message to this address is automatically distributed to Richard, Jim, and publisher Perry Remaklus.

To our beta testers we owe another big “thank you.” Beta testing software can be a thankless task, yet our team persisted through mysterious and frustrating crashes, tools that did not work, and more mysterious crashes. Beta testing is an important part of software creation because source code that functions perfectly with standard test images and the right input values will run amok when a beta tester plugs in a wild value or tries to process an image containing zeros and negative numbers. We learn from that, and build better software. It falls to the beta tester to give us the unwelcome news that “the old tool was better,” or “clicking ‘New Image’ made a copy of my old image!” When you look under the hood, software is a complex machine composed of bits and bytes instead of gears and pulleys—and the beta testers tell us when that machine is running smooth and purring like a big contented cat.

Finally, we thank our past and future readers. Your comments and suggestions have been and continue to be important to us. As we said in the preface to the first edition, “no lengthy book, and no large program, can ever be entirely free of errors.” As time passed, however, we found some typos but gratifyingly few significant errors, and, with each subsequent printing of the book, we have corrected all known errors. We do welcome your feedback and comments, your suggestions and ideas (email: [aip4win@willbell.com](mailto:aip4win@willbell.com)) for the next edition of the *Handbook* and **AIP4Win**.

RICHARD BERRY

JAMES BURNELL

---

# 1 Basic Imaging

---

Starlight falls on every square meter of Earth. Yet it must fall on a telescope, the telescope must focus the light into an image, and a detector must respond to the image before an astronomer can use starlight to learn about the heavens. This chapter is about the detection of starlight, and this book is about the step that follows detection—image processing—converting data from the detector into a measurement or an image that is informative and even beautiful.

## 1.1 Light

Light is variously described as *rays* that travel in straight lines, as *waves* of electromagnetic energy, and as *particles* called photons. These descriptions are each valid in the sense that light exhibits ray-like, wave-like, and particle-like properties. None of these descriptions is complete by itself, yet each is a valid characterization of some aspects of the behavior of light.

The ray description treats light as a purely geometric phenomenon—it says nothing about the nature of light. The Greeks knew that light moves in straight lines even though they didn't understand its nature. The ray description is useful for talking about optical systems. The wave description elegantly unravels the phenomenon of diffraction, and the particle description offers insight into the nature of CCD operation. The three seemingly different descriptions only serve to illuminate different aspects of the behavior of light.

Light is electromagnetic radiation. Moving charges generate self-sustaining electric and magnetic fields that propagate away from the source as radio, infrared, visible light, ultraviolet, x-rays, and gamma-rays. Only the wavelength and frequency of the waves distinguish the light we see from forms of electromagnetic radiation we do not see. The relationship between the wavelength and frequency for electromagnetic radiation is:

$$\lambda\nu = c \tag{Equ. 1.1}$$

where  $\lambda$  is the wavelength of the light,  $\nu$  is the frequency, and  $c$  is the speed of light ( $2.99\times10^8$  meters per second). The wavelength characteristic of yellow-green light (to which the human eye is most sensitive) is 550 nanometers, corresponding to a frequency of 545 Terahertz ( $5.45\times10^{14}$  Hz).

## Chapter 1: Basic Imaging

Although light carries energy, nothing in classical physics prepared turn-of-the-century scientists for the discovery that the energy is quantized, behaving exactly like a stream of particles each carrying a specific energy. The energy carried by a single photon of wavelength  $\lambda$  is:

$$E_{\text{photon}} = \frac{hc}{\lambda} \text{ [electron-volts]} \quad (\text{Equ. 1.2})$$

where  $c$  is the speed of light and  $h$  is Planck's constant; their product,  $hc$ , is 1240 electron-volt nanometers. Each photon (i.e., each “particle”) of yellow-green light carries an energy potential of 2.25 electron volts. Long-wavelength photons have lower energy per photon; short-wavelength photons carry more energy. Ultra-high energy gamma-ray photons pack as much punch as a cruising mosquito, but 2.25 electron-volts is barely more than the energy required to kick free one electron in a crystal of silicon.

When we discuss how light travels through space, how it bounces off mirrors, how it bends as it passes through transparent materials, and how it forms images, we will use the ray and wave descriptions of light. When we examine the interaction of light with detectors, where the quantized nature of light becomes evident, we speak of light as photons.

## 1.2 Image Formation

Light alone is not enough. We need to know what direction the light that reaches us comes from—for that we have eyes. Eyes form images, sorting the flood of photons by direction. Imagine an amoeba under a clear sky, bathed in a flux of billions of photons. Although the amoeba can tell the total amount of energy falling on it, it cannot determine with any accuracy where the photons came from.

Images are the marriage of intensity with direction. Images are patterns of light intensity in which the amount of light on any point corresponds to the direction of origin of the light. Once light has been organized into an image, it can be detected by a retina, a silver-halide emulsion, or electronic camera.

### 1.2.1 Pinhole Imaging

The most basic method for finding the direction of light is shadowing. Even the amoeba can tell that the side of its body facing a strong light source is warmer (or more chemically stimulated) than the side facing away, and turn toward (or away from) the light. X-ray and gamma-ray astronomers once used detectors that were in principle only slightly more sophisticated than the amoeba's simple shadowing.

However, the flood of light can be converted into an image with a simple pinhole camera. Any closed box or room (i.e., a *camera* in Latin) with a small aperture (*apertura* = “opening”) can serve as an image-forming device. Rays of light from outside objects enter the aperture and continue in straight lines to the opposite side. On the image surface, light from each source thus has a well-defined location: the angular positions of the light sources have been mapped onto  $(x, y)$ .

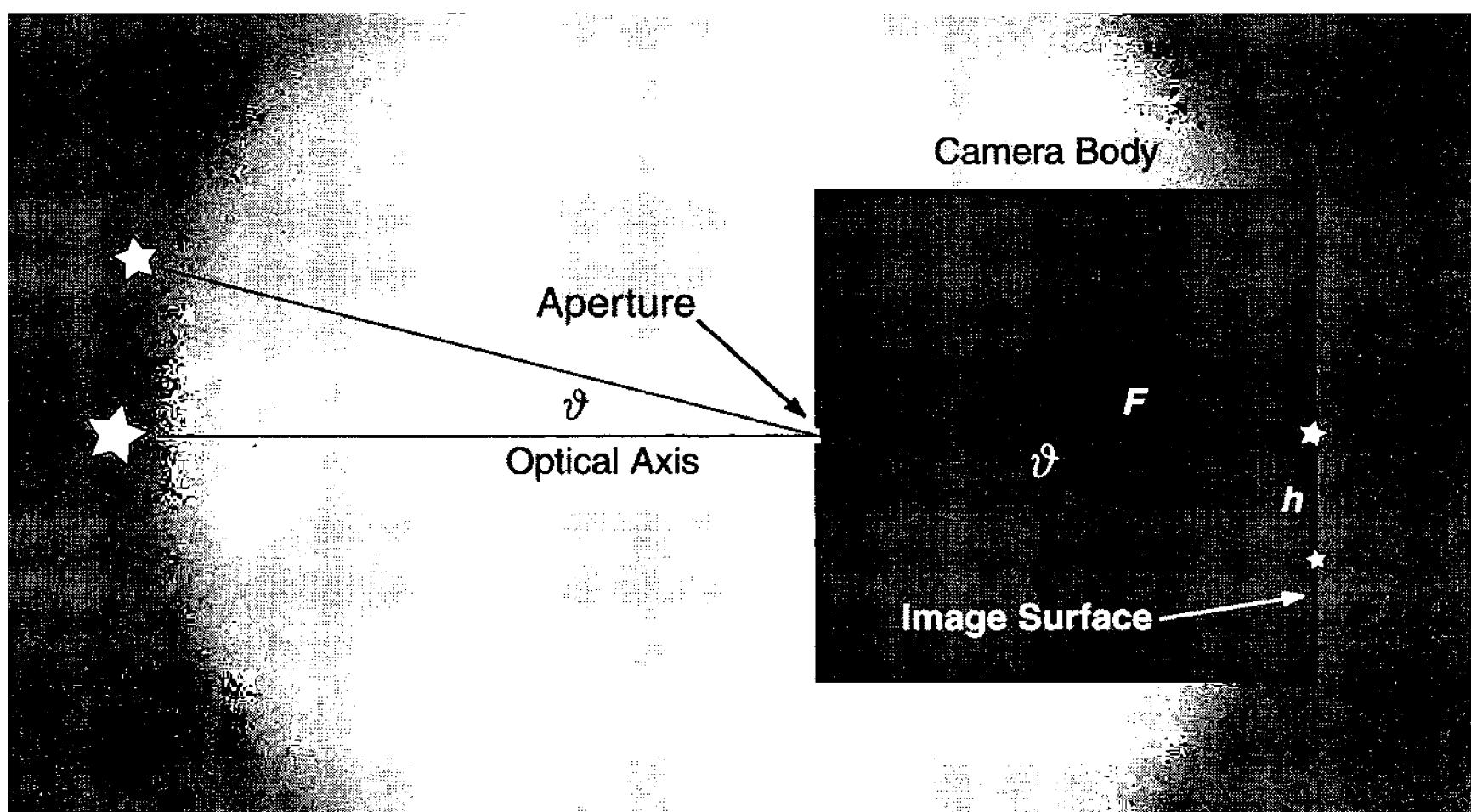


Figure 1.1 At the back of a pinhole camera, light sources at different angular positions in front of it are mapped onto locations in the image. The intensity of the light at each point in the image depends on the flux entering the camera from the light sources outside it.

locations on a surface. The organized pattern of light intensities is an *image*.

The feature that distinguishes an image from illumination is the spatial organization of intensities in the image. The amoeba senses the total illumination without knowing where the light comes from, but in a *camera* the light is sorted into location by its direction of origin. As beings with eyes, we are so used to knowing the angular positions of light sources that it tends to be difficult for us to imagine light without knowing what direction it came from.

Let us now examine the properties of the images formed by a pinhole camera. A pinhole camera consists of a light-tight box. At the center of one end is the aperture and at the other end is a surface on which light falls. (Important to remember: pinhole cameras can be any shape and size, and the receiving surface need not be flat. The camera still works, but the math is more complicated.) Light from many sources falls on the front of the camera, and a tiny fraction of it enters the aperture. Light that enters crosses the interior and falls on the receiving surface.

Inside a camera, we define the line between the aperture and the point that lies directly “under” the aperture as the *optical axis*. This point is the point on the receiving surface that is closest to the aperture—a distance called the *focal length*—and light reaching this point arrives perpendicular to the receiving surface. If we point the camera at a source of light—that is, align the optical axis of the camera with the direction of the source—then the image of the source lies on the receiving surface and on the optical axis.

Now consider the location of a source at an angular distance  $\vartheta$  (theta) from the optical axis. The angle  $\vartheta$  is measured at the aperture of the camera, and the image of this source forms some height,  $h$ , from the optical axis:

## Chapter 1: Basic Imaging

$$h = F \tan \vartheta \quad (\text{Equ. 1.3})$$

where  $F$  is the focal length. The greater the angular distance a source lies from the optical axis, the greater the linear distance its image lies from the optical axis.

Pinhole cameras can easily cover off-axis angles of  $45^\circ$  to  $60^\circ$ , so that the whole image spans a  $90^\circ$  to  $120^\circ$  angle. The rectilinear mapping of source position to image location has some interesting consequences. An array of sources that lies on a straight line in front of a camera will lie on a straight line on the image. In addition, the images of sources that lie on a great circle on the celestial sphere will lie along a straight line on the image. In terrestrial imaging, the sides of buildings are rendered as straight lines, and in astronomical imaging, horizons, equators, and longitude circles are rendered as straight lines. Images in the pinhole camera are said to be both rectilinear and free of distortion.

As imaging devices, pinhole cameras collect too few photons to be practical. If the aperture is enlarged to make the image brighter, light from different sources overlaps, and the image is blurred. If the aperture is too small, diffraction caused by the wave properties of light degrades the image. For practical imaging, light must be collected over a large area and focused into an image.

### 1.2.2 Lens Cameras

The aperture in a pinhole camera is simply a hole: it does nothing to the light that passes through it. Suppose that we place a carefully shaped disk of glass—a lens—in the aperture. Because light passes through glass more slowly than it passes through air, a wavefront encountering glass slows and changes its direction of propagation—that is, the rays of light bend when they pass from air to glass or from glass to air. By shaping the lens so that each ray is bent in direct proportion to its distance from the center of the glass, parallel rays of light from a source will converge until they cross. The rays cross at a point called the *focus*. After reaching the focus, the rays will continue on diverging paths unless they are intercepted at the focus by a viewing screen, a piece of photographic film, or an electronic detector, such as a CCD chip.

Rays that pass through the center of a lens are called principal rays. Principal rays pass through a lens and exit parallel to their original paths. Since the lens is shaped so that all rays from a given source cross the principal ray at the focus at the receiving surface, all of the rays from a source meet at the focus. A camera equipped with a lens collects all of the light from a star that falls on the lens and concentrates it into a single bright point.

In practice, lenses almost always contain multiple elements made of different types of glass. As light passes through the multiple surfaces and glass types, the wavefront undergoes subtle manipulation. A great deal of art and effort goes into designing optical systems that bring light to an accurate focus.

However complex its internal design, a compound lens acts like an equivalent simple lens. The distance between the location of the equivalent simple lens and the focal plane is the *focal length* of the lens. Optical designers work hard to

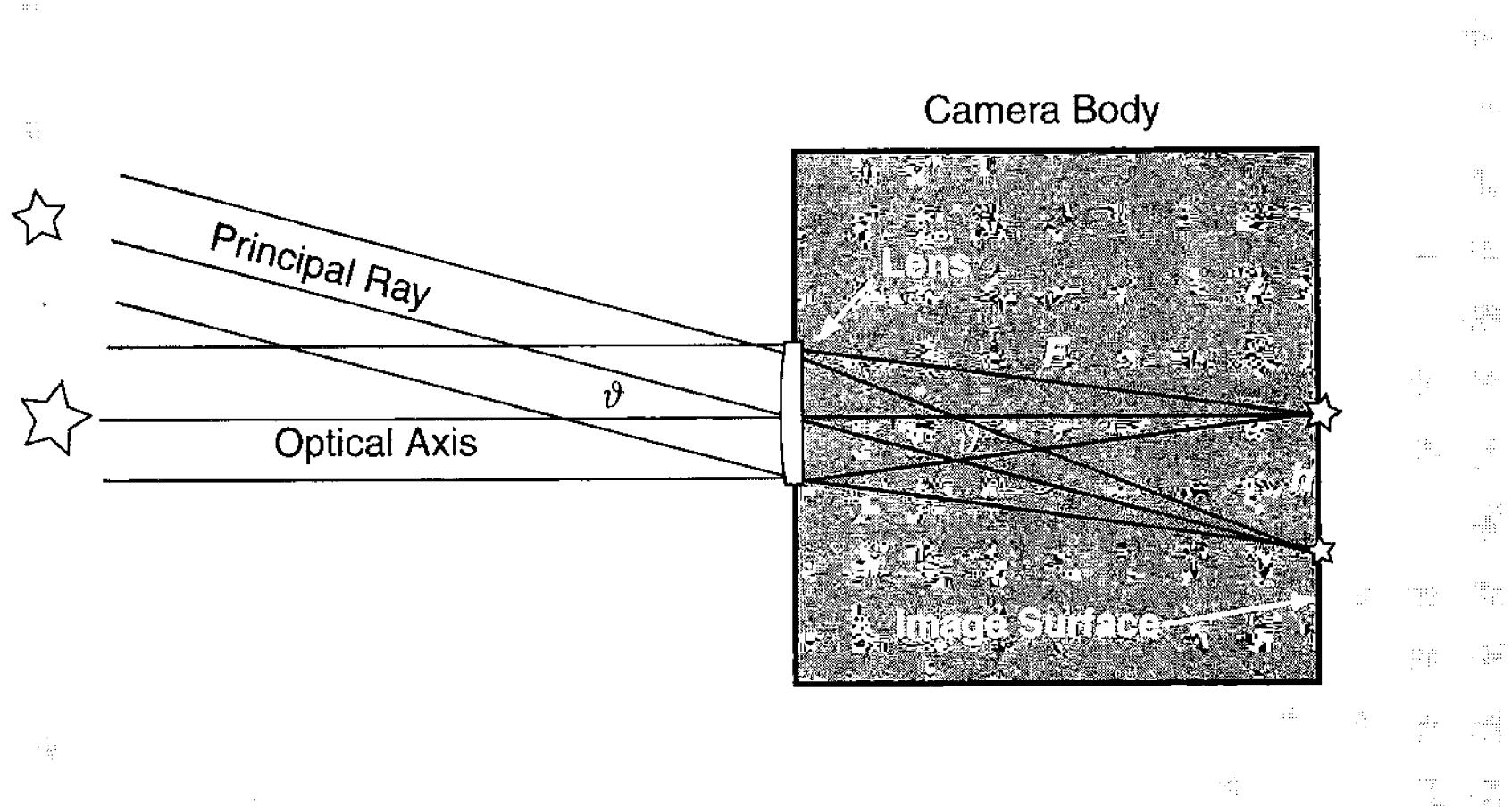


Figure 1.2 The lens of a camera directs rays from a large area to a focal point, resulting in a bright, sharp image. The image of each object falls in exactly the same place that it would in a pinhole camera because the lens directs all rays to converge on the principal rays, which remain parallel to their original paths.

maintain a strict linear relationship between the focal length,  $F$ , the tangent of the angular distance from the optical axis,  $\tan \vartheta$ , and the ray height,  $h$ , from the optical axis:

$$h = F \tan \vartheta. \quad (\text{Equ. 1.4})$$

This is exactly the same rectilinear relationship found in the pinhole camera; that is, under ideal conditions, lens geometry (Equ. 1.4) is the same as pinhole geometry (Equ. 1.3). The lens directs the light from the sources in front of the camera to a tiny point of focus. A camera equipped with a lens forms an image with the same geometric properties as that formed by a pinhole camera, but because the lens admits more light than a pinhole, the image is brighter.

Despite the best efforts of their designers, lenses exhibit aberrations, or departures from perfect imaging. Aberration means that the rays from a point source (such as a star) fail to focus at a common focal point. In spherical aberration, for example, rays at different distances from a principal ray converge ahead of or behind the focus point. Coma and astigmatism are aberrations that affect images away from the optical axis. Lenses also suffer from chromatic aberration, in which rays of different wavelength fail to meet at a common focus. Ordinary camera lenses perform well up to  $20^\circ$  to  $30^\circ$  from the optical axis, but at larger angles the aberrations degrade image quality.

Lenses focus much more light into an image than a pinhole. The ratio between the diameter of the bundle of rays entering the lens and the focal length is called the *focal ratio*. If a lens has a diameter of 100 millimeters and a focal length of 500 millimeters, its focal ratio is  $f/5$ .

## Chapter 1: Basic Imaging

The smaller the focal ratio, the greater the concentration of light onto the focal surface. Typical pinhole cameras have focal ratios of  $f/500$  (i.e., the aperture is  $1/500$  of the focal length), but ordinary camera lenses have focal ratios as low as  $f/2$ . Because the light-gathering area of the  $f/2$  lens is 62,500 times greater than the area of the small pinhole, the image formed by the lens is 62,500 times brighter than a pinhole image.

### 1.2.3 Telescopes

Telescopes are image-forming optical systems designed and optimized for astronomy. Because astronomical sources are faint, astronomers want a telescope with a large aperture; and because large lenses are difficult to manufacture, telescopes larger than about 300 millimeters aperture usually employ mirrors rather than lenses to gather and focus light.

As all amateur astronomers know, telescopes come in three varieties: refracting, reflecting, and catadioptric (combinations of reflecting and refracting). Quality refractors employ several different types of glass to correct chromatic aberration and bring all wavelengths to nearly the same focus. Reflectors employ mirrors which lend themselves to large apertures; the types most common in astronomy are the Newtonian and the Cassegrainian configurations. Catadioptrics use mirrors to gather and focus the light, and lens elements to correct residual aberrations.

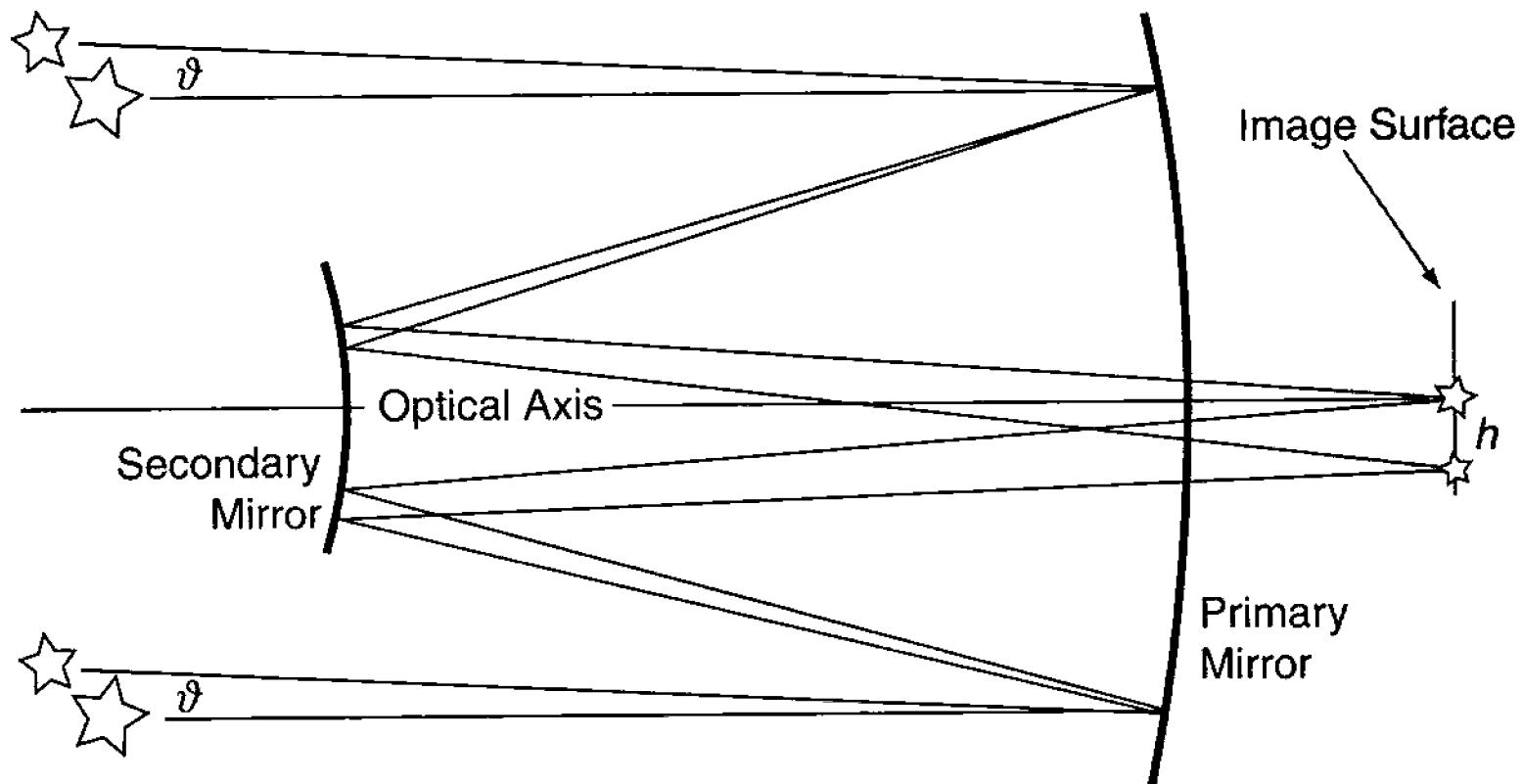
To a excellent approximation, telescopes obey the rectilinear relationship as shown in Equ. 1.4, and their images are therefore rectilinear. Because telescopes cover much smaller off-axis angles than camera lenses, seldom more than 1 or 2 degrees, this formula can be simplified. Over small angles  $\vartheta \approx \tan \vartheta$ , where  $\vartheta$  is measured in radians; so for quick calculations with small angles, the formula becomes:

$$h \equiv \frac{F\vartheta^\circ}{57.3}, \quad (\text{Equ. 1.5})$$

where  $\vartheta^\circ$  is the off-axis angle in degrees,  $F$  is the focal length of the telescope,  $h$  is the off-axis distance in the focal plane, and 57.3 is a constant that converts the angle from degrees to radian measure.

In astronomy, high-quality images are essential. Telescopes are designed and manufactured to form diffraction-limited images; that is, images as nearly perfect as the laws of physics allow. To form a sharp image, light waves from a distant source must meet at the focus of a telescope in phase; that is, with the peaks and troughs of light waves from different parts of the aperture lined up. For this to happen, the total distance along every optical path must be the same to within a fraction of a wavelength of light, or about 100 nanometers (4 micro-inches). Although fraction-of-a-wave accuracy is a demanding criterion, do-it-yourself amateur makers routinely attain this accuracy in their homebuilt telescopes.

Even in an optically perfect instrument, however, the light from a star cannot be focused to a perfect point. As the waves converge toward focus, they arrive in



**Figure 1.3** In a telescope—schematically shown here as a Cassegrainian—parallel rays from a source come together at a focus. Although principal rays would be blocked by the secondary mirror in a real telescope, their paths nonetheless define the focal length of the optical system.

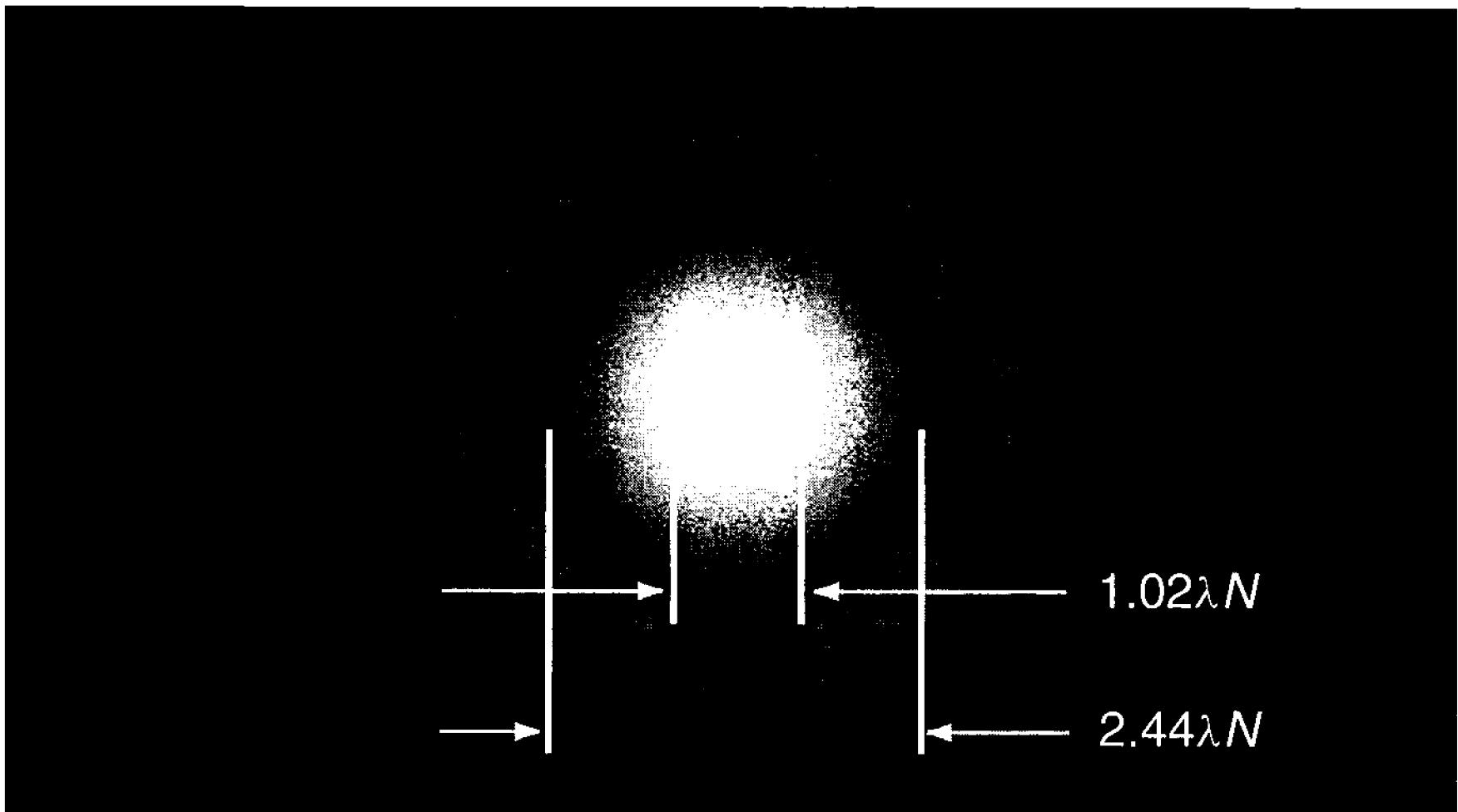
almost perfect phase in a region surrounding the geometric point of focus; so instead of being concentrated in an infinitesimal point, light appears in a small spot called the Airy disk, or diffraction disk. Due to the wave nature of light, some of the light is deposited outside the Airy disk in a faint pattern of rings. In a perfect unobstructed optical system with a circular aperture, the Airy disk contains 84% of the incident energy, and the diffraction rings contain the remaining 16%. If the maximum path-length errors in a telescope can be made smaller than approximately 20% of the wavelength of the light, diffraction will be the limiting factor in the quality of images, rendering improvement of the optical system beyond a fraction of a wavelength less and less practical.

The distribution of light at focus, whether a perfect diffraction figure or a degraded blur resulting from optical aberrations, is called the point-spread function of the telescope. The point-spread function describes how the telescope forms an image of a perfect mathematical point. Because every plane wavefront that enters the telescope is transformed into the point-spread function at the focus, the image consists of countless overlapping copies of the point-spread functions. This is true for images of galaxies, nebulae, and planets as well as stars.

The angular diameter of the Airy disk sets an important constraint on what can be seen or imaged with a telescope, and its linear diameter at the focal plane is a key parameter in image formation. The angular diameter of the Airy disk to the first dark diffraction ring,  $\vartheta_{\text{Airy}}$ , is:

$$\vartheta_{\text{Airy}} = 2.44 \frac{\lambda}{A} \text{ [radians]}, \quad (\text{Equ. 1.6})$$

## Chapter 1: Basic Imaging



**Figure 1.4** In a high-quality unobstructed telescope, 84% of the light from a star lies inside the first dark diffraction ring, and the remaining 16% of the light goes into the surrounding diffraction rings. However, a small, bright inner core of the diffraction disk contains nearly half of the star's light.

where  $\lambda$  is the wavelength of the light, and  $A$  is the aperture in the same units. For example, the angular diameter of the Airy disk at the focus of a telescope with an aperture of 200 millimeters (8 inches) at a wavelength of 656 nanometers is:

$$\vartheta_{\text{Airy}} = 2.44 \times \frac{656 \times 10^{-9}}{200 \times 10^{-3}} = 8 \times 10^{-6} \text{ [radians]}, \quad (\text{Equ. 1.7})$$

which corresponds to 1.6 seconds of arc. (Note that all measurements were converted to the same unit—meters. To convert from radians to degrees, multiply by 57.3; to convert from radians to minutes of arc, multiply by 3438; to convert to seconds of arc, multiply by 206,265.) The linear diameter of the Airy disk,  $d_{\text{Airy}}$ , out to the first dark diffraction ring, is:

$$d_{\text{Airy}} = 2.44 \lambda \frac{F}{A} = 2.44 \lambda N, \quad (\text{Equ. 1.8})$$

where  $A$  is the aperture and  $F$  the focal length of the telescope, and  $N$  is its focal ratio,  $F/A$ . The first term in the formula above is simply the angular diameter times the focal length of the telescope.

However, nearly half of the image-forming light is concentrated in the small, bright central core of the diffraction disk, a much smaller region defined by the diameter at which the light has fallen to half its central intensity. This region is the full-width at half-maximum (FWHM) of the diffraction disk,  $\vartheta_{\text{FWHM}}$ . The angular diameter of the small, bright core of the point-spread function is:

$$\vartheta_{\text{FWHM}} = 1.02 \frac{\lambda}{A} \text{ [radians]}, \quad (\text{Equ. 1.9})$$

and its linear diameter is:

$$d_{\text{FWHM}} = 1.02 \lambda \frac{F}{A} = 1.02 \lambda N, \quad (\text{Equ. 1.10})$$

where  $d_{\text{FWHM}}$  is the FWHM of a perfect star image.

Assuming that the telescope has an  $f/10$  optical system, the FWHM of a perfect star image is:

$$d_{\text{FWHM}} = 1.02 \times 656 \times 10^{-9} \times 10 = 6.7 \times 10^{-6} \text{ meters}, \quad (\text{Equ. 1.11})$$

or 6.7 micrometers. The diameter of the FWHM of the diffraction disk is a realistic measure of the smallest detail contained in an astronomical image.

*Note:* The scientific unit for  $10^{-6}$  meters is the *micrometer*. However, many engineers and technicians use a less formal term for micrometers: the *micron*. In this book, we use both the formal and the informal terms as they are normally used by engineers and scientists.

## 1.3 Detectors

*Detector* is a technical term that refers to a device that generates a signal in response to a phenomenon such as light. Amateur astronomers generally use one of three types of detector: the retina in their eye, a photographic emulsion, or an electronic sensor. In this section, we examine how each works.

### 1.3.1 The Human Retina

The human eye combines an image-forming lens and a versatile detector into one compact unit. The detector in the eye is called the retina; it is an array of cells that detects light by the breakdown of the chemical rhodopsin. The breakdown products of rhodopsin trigger nerve responses that send an encoded signal to the brain through the optic nerve, where the signals are interpreted as a visual scene.

The lens in the eye has a focal length of about 16 millimeters, and a pupil that varies in aperture from 7 to 1.5 millimeters (yielding focal ratios from  $f/2.3$  to  $f/11$ ) to adjust the amount of light admitted to the eye. The retina is located on the curved interior of the eye, opposite the lens and if flattened, would measure about 40 millimeters in diameter. Because the retina is fully integrated into the structure of the eye, it cannot be attached directly to a telescope, but must be used in conjunction with an *afocal* lens system consisting of an eyepiece and the objective lens of the eye itself.

Nevertheless, for comparison with photographic and CCD detectors, we shall examine the properties of the retina as a detector. It contains some 100,000,000 light-sensing cells of two types: rods and cones. Rod cells cover the

## Chapter 1: Basic Imaging

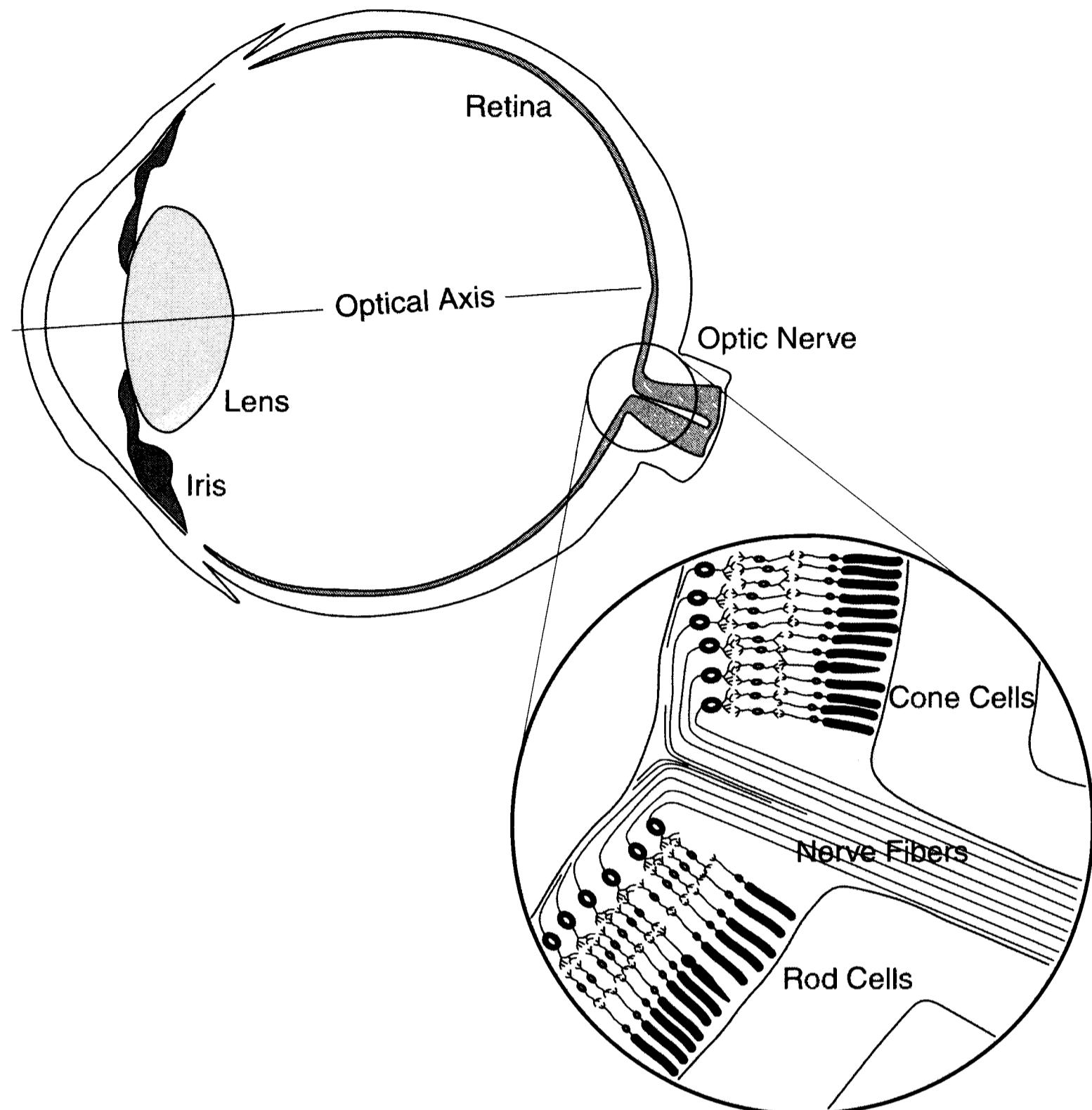


Figure 1.5 The eye is a high-performance imaging system containing a lens, detector array, and an on-board image processing system in the neural network overlying the rods and cones. The eye is so sensitive that under optimal conditions, an observer can see a flash of light consisting of ten photons.

entire retina, and work well at low light levels. Cone cells are clustered near the optical axis of the lens, and operate best at high light levels. Individual cone cells are optimized to detect light at different wavelengths to provide color vision.

Near the optical axis, the retina consists of cone cells—each about 2 microns in diameter—packed to a density of about 1,000 per square millimeter. The cones match the diffraction limit for the lens of the eye, providing an angular resolution of about 80 seconds of arc. Away from the optical axis, resolution is much lower. The light-sensitive parts of the rods and cones are nearly 100% efficient at absorbing photons, but because they are located on the back side of the retinal tissue, approximately half the incident light is lost in passing through the overlying neural network. This layer of nerve cells mediates against noise by sensing light only when several adjacent cones or rods are simultaneously triggered. Because of these losses, the overall quantum efficiency of the retina is about 15% at the peak

rod-cell sensitivity at 505 nm wavelength. The effective integration time of the retina is 100 to 200 milliseconds (consistent with its function in a moving animal), and the generation of the signal is continuous.

In addition to filtering noise, the network of nerve cells preprocesses the signals generated by the light-sensing cells so as to detect edges, lines, and small differences in color. Thus, the signals that travel to the brain are not raw brightness data, but partially processed information on the shape, size, and color of objects in the visual field.

### 1.3.2 Photographic Emulsions

From 1880 to 1970, photographic emulsions were the primary detectors that professional astronomers used to record images. A photographic emulsion consists of silver halide crystals dispersed in a gelatin matrix, coated on a glass plate backing or a flexible plastic film backing. Although silver bromide is the primary silver salt, traces of chloride and iodide may be present. In the manufacture of photographic materials, the size and shape of the crystals is carefully controlled; the most sensitive crystals are approximately one micron across and flattened. In addition, the spectral sensitivity of the finished emulsion depends on dyes that absorb photons and transfer their energy to the silver bromide crystals.

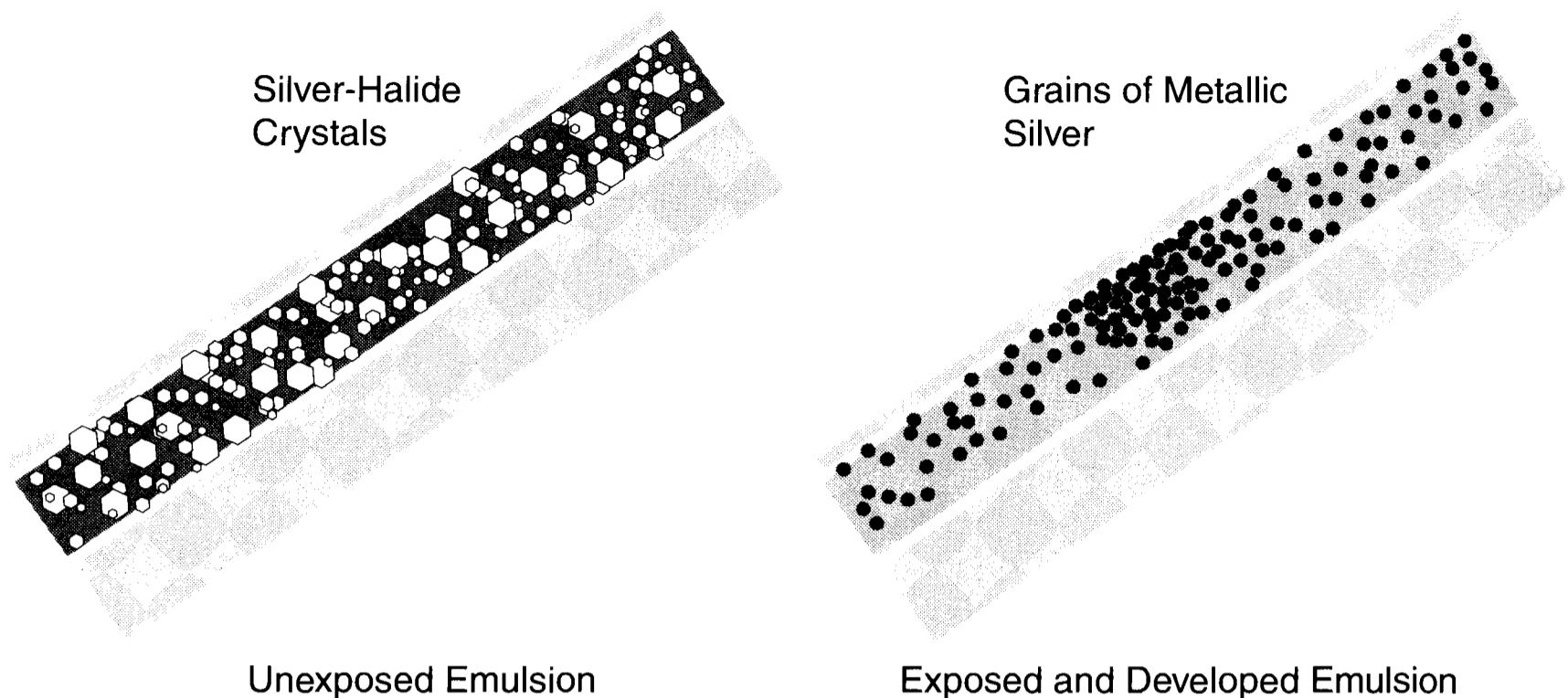
Photographic emulsions differ from the retina and the CCD because exposure to light produces irreversible changes in the detector; in other words, you can use a film or plate once and once only. When a photon of energy greater than about 2 electron-volts impinges on a silver bromide crystal, it creates a defect in the crystal structure. Such defects can migrate through the crystal, and may spontaneously “heal” after several seconds or minutes. When several such defects have been created in a crystal, however, they coalesce into a stable defect called a development center. A stable defect involving only a few dozen atoms is enough to render the entire crystal—containing billions of silver atoms—chemically unstable.

When the emulsion is subsequently placed in a solution of photographic developer, the defect triggers the chemical reduction of the entire crystal to metallic silver. The capture of three to four photons can thus precipitate the formation of a grain of silver containing 10 billion silver atoms—a remarkable feat of chemical amplification. After development, the emulsion is washed in a fixer (a bath that dissolves undeveloped crystals of silver halide), then thoroughly rinsed and dried.

The developed photographic image thus consists of a clear backing coated with a gelatin layer containing microscopic grains of metallic silver where light struck the emulsion. Because the image appears dark or opaque where light fell, it is called a *negative*. By passing light through the negative to another sheet of photographic emulsion coated on white paper, you can make a negative of the original negative, which is called a positive print.

Photographic images have interesting properties. The process is inherently nonlinear with respect to its exposure to light, because once a grain crosses the development threshold, it is developable; and further exposure cannot make it more

## Chapter 1: Basic Imaging



**Figure 1.6** Photographic detectors are made by coating a thin sheet of plastic or glass with gelatin containing microscopic crystals of silver bromide, iodide, and chloride, protected by a top layer of plain gelatin. After exposure to light and chemical development, the image consists of tiny grains of metallic silver.

developable. Because of this, the number of developed grains is less than proportional to the number of photons. Furthermore, since a developed emulsion is three-dimensional, developed grains shadow other developed grains, resulting in a further undercount of the original number of photons.

Photographic emulsions are also nonlinear with respect to exposure time. Unless three or four defects are created in a sufficiently short interval to produce a development center, individual defects may disappear before coalescing; and the grain will not be developable. Photographic materials, therefore, record a smaller fraction of the incident photons in long exposures to low levels of light than they record when the same number of photons arrive in a short time. At the light levels found in astronomical images, faint parts of an image are recorded less efficiently than are its bright parts.

Finally, because the silver halide grains in the emulsion were scattered randomly, the number of developed grains in small regions that have received the same exposure varies randomly around an average value. The random distribution of developed grains causes the “grainy” appearance of photographs.

During the decades when emulsion photography was king in astronomy, numerous ways were found to enhance the performance of emulsions. Bathing plates in a dilute solution of ammonium hydroxide a day or two before exposure made the plate more sensitive. By exposing plates to faint light just before use in the telescope, astronomers found they could create one or two defects in each crystal, reducing the number of photons necessary to cross the development threshold. Baking plates and films for several hours before exposure likewise created defects, enhancing the performance of some emulsions. However, when researchers discovered that the primary reason that single-photon defects decayed was that

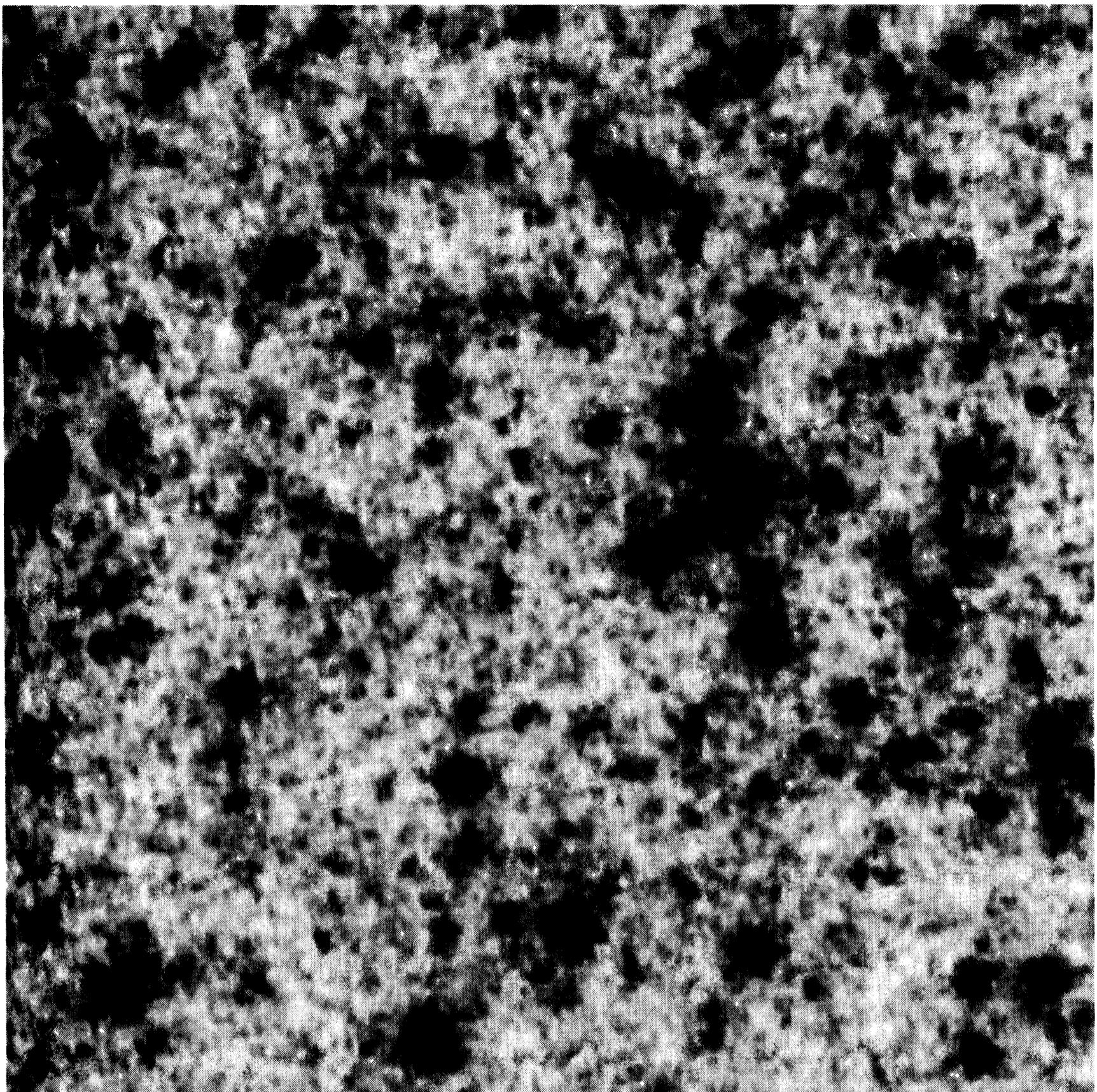


Figure 1.7 This highly enlarged section of an astronomical photograph shows the random distribution of silver grains in a developed image. Random variation in the number of developed grains in a given area causes the characteristic “grainy” appearance of photographic images.

water and oxygen were present in the emulsion, astronomers began baking films and plates in vacuums or gas mixtures containing hydrogen. This process—hydrogen hypersensitizing—drove out the water and reduced the oxygen, and produced a twenty-fold improvement in the fine-grain emulsion of Kodak 2415 Technical Pan. For serious amateur astrophotographers, hydrogen-hypered Tech Pan is the ultimate film.

Modern black-and-white films are made with multiple layers of emulsions with differing crystal size to compensate for the limited dynamic range of simple emulsion coatings. Modern color films are made with multiple layers of emulsions with differing color sensitivity to separate and record the full gamut of color, and sophisticated chemistry to replace developed silver grains with clouds of colored dyes.

Overall, photographic emulsions serve as remarkably good image detectors.

## Chapter 1: Basic Imaging

With hydrogen hypering, modern emulsions like Technical Pan perform well in exposures of several hours. Even without special treatment, modern black-and-white and color films are still efficient in exposures ranging from 5 to 20 minutes.

Although their overall quantum efficiency ranges from around 0.5% to about 4%, with spectral sensitivity peaking in the blue and green regions of the spectrum, photographic detectors are readily available in large sizes. Standard 35-mm film has a detector area 24 mm by 36 mm, standard 120-format roll film produces images 60 mm by 70 mm, and standard 4 × 5-inch sheet films have active areas of 100 mm by 125 mm. Although individual grains are only a few microns across, the smallest effective area capable of producing a good signal-to-noise ratio ranges from 5 to 20 microns. Assessed in the terms used for electronic sensors, one frame of fine-grain 35-mm film offers the resolution of a 10 megapixel electronic detector.

Unfortunately, because photographic detectors are inherently nonlinear, and because a piece of film can be used only once, high-precision measurements of light intensity are not possible. With typical quantum efficiencies of 1%, photographic exposure times must be 20 to 60 times longer than comparable exposures with electronic sensors. For capturing physically large images, however, photographic emulsions are competitive with electronic sensors, especially in applications where the detector must be simple, compact, rugged, and inexpensive.

### 1.3.3 Electronic Detectors

Since their first astronomical use in 1976, electronic sensors have steadily gained ground as the detectors of choice in astronomy. In amateur astronomy, *charge-coupled devices* (CCDs) have been the dominant type. CCDs can detect light over a broad range of wavelengths, and they offer both high quantum efficiency and low noise. Furthermore, dark current and nonuniformities in sensitivity can be subtracted or divided out, thereby minimizing these shortcomings.

Challenging the CCD is another class of electronic sensor, the CMOS device. CMOS stands for *complementary metal-oxide semiconductor*, referring to the method of making them. While CMOS offers lower manufacturing costs, the resulting sensors are less sensitive and noisier than CCDs—a situation that will almost certainly change as the technology improves.

Today, electronic sensors incorporating CCDs and CMOS devices are used widely in amateur astronomy. Electronic cameras designed for astronomical work almost exclusively use CCDs, as do high-end digital cameras; but consumer-grade digital cameras and webcams increasingly rely on CMOS devices.

**How CCDs Work.** CCDs consist of an array of identical metal-oxide semiconductor capacitors formed on a silicon substrate. Each element in the array is called a photo-detector junction or photosite. The charge in each photosite is isolated from the others by a voltage applied through conductive channels on the surface of the silicon. At the beginning of an exposure, the capacitors are charged positively and then disconnected. As photons enter the silicon crystal lattice and are absorbed, they raise electrons from a low-energy valence-band state to a high-

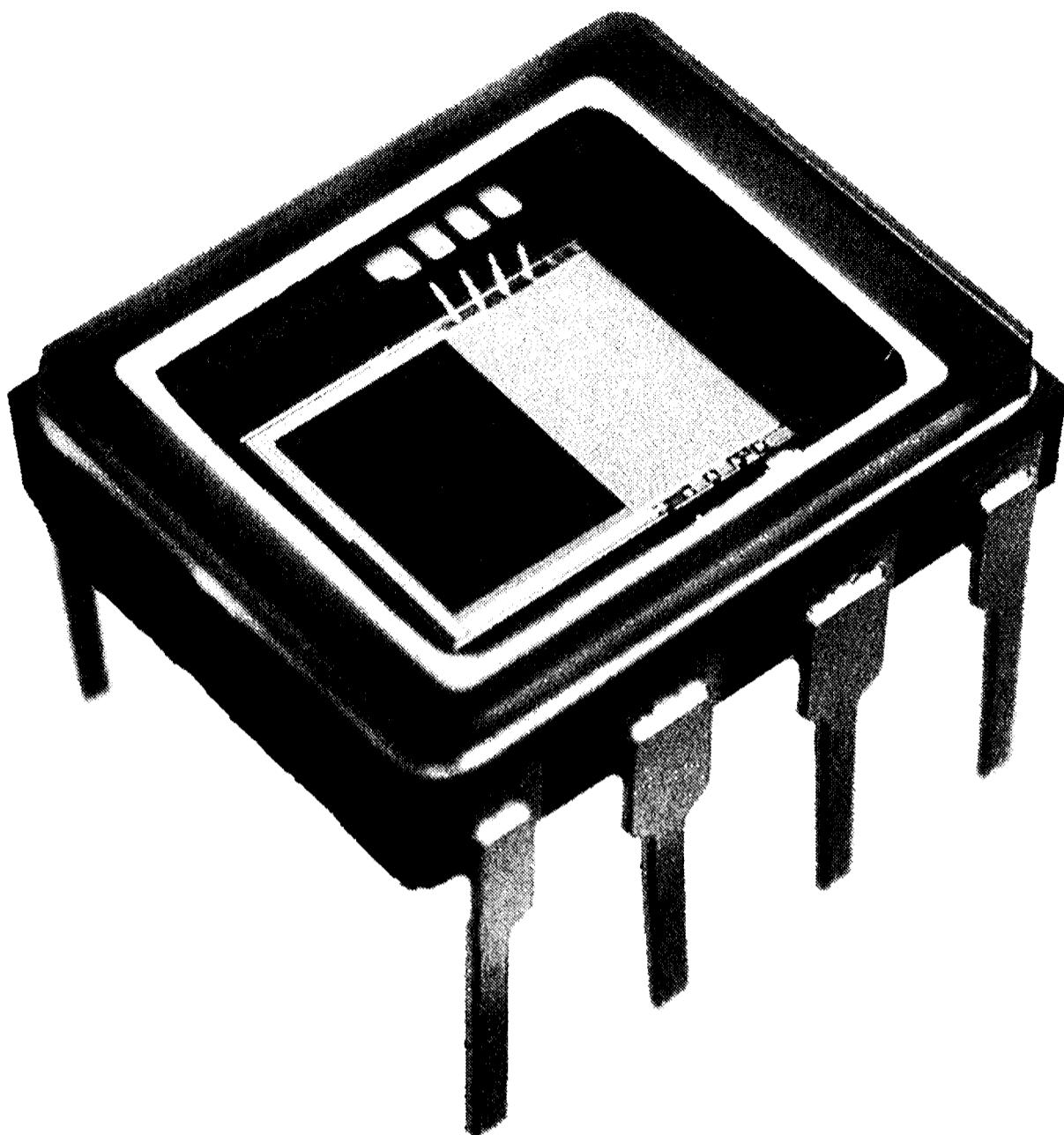


Figure 1.8 Electronic detectors have revolutionized astronomy. They are compact, highly sensitive to light, and—unlike a photographic emulsion—can be used over and over again. Shown here is a Texas Instruments TC237, a 640x480-pixel CCD in a package just half-an-inch long.

energy conduction-band state, partially discharging the capacitors. The degree of discharge of each capacitor is proportional to the number of photons that strike it during the exposure. At the end of the exposure, the electrons remaining in the photosites are sequentially shifted (or “clocked”) to a charge-sensing node, and amplified; then the signals are passed to external circuitry to be digitized and stored.

The detector characteristics of CCDs derive directly from their construction. In silicon, the energy gap between the valence band and the conduction band is 1.1 electron volts, so that only photons with energies higher than that can boost an electron into the conduction band and be detected. This energy limit corresponds to a wavelength of 1100 nanometers, in the near infrared part of the spectrum. At shorter wavelengths, however, silicon becomes progressively more reflective so that the photons never enter the crystalline lattice, and hence cannot be absorbed. Silicon CCDs, therefore, reach peak quantum efficiencies of 40% to 90% between 500 and 950 nanometers wavelength.

In CCDs, the number of electrons boosted into the conduction band is directly proportional to the incident flux of photons. CCDs are highly linear as long as the total charge that collects in a photosite is too small to leak past the charge barriers that separate each photosite from its neighbors. In practice, most CCDs are

## Chapter 1: Basic Imaging

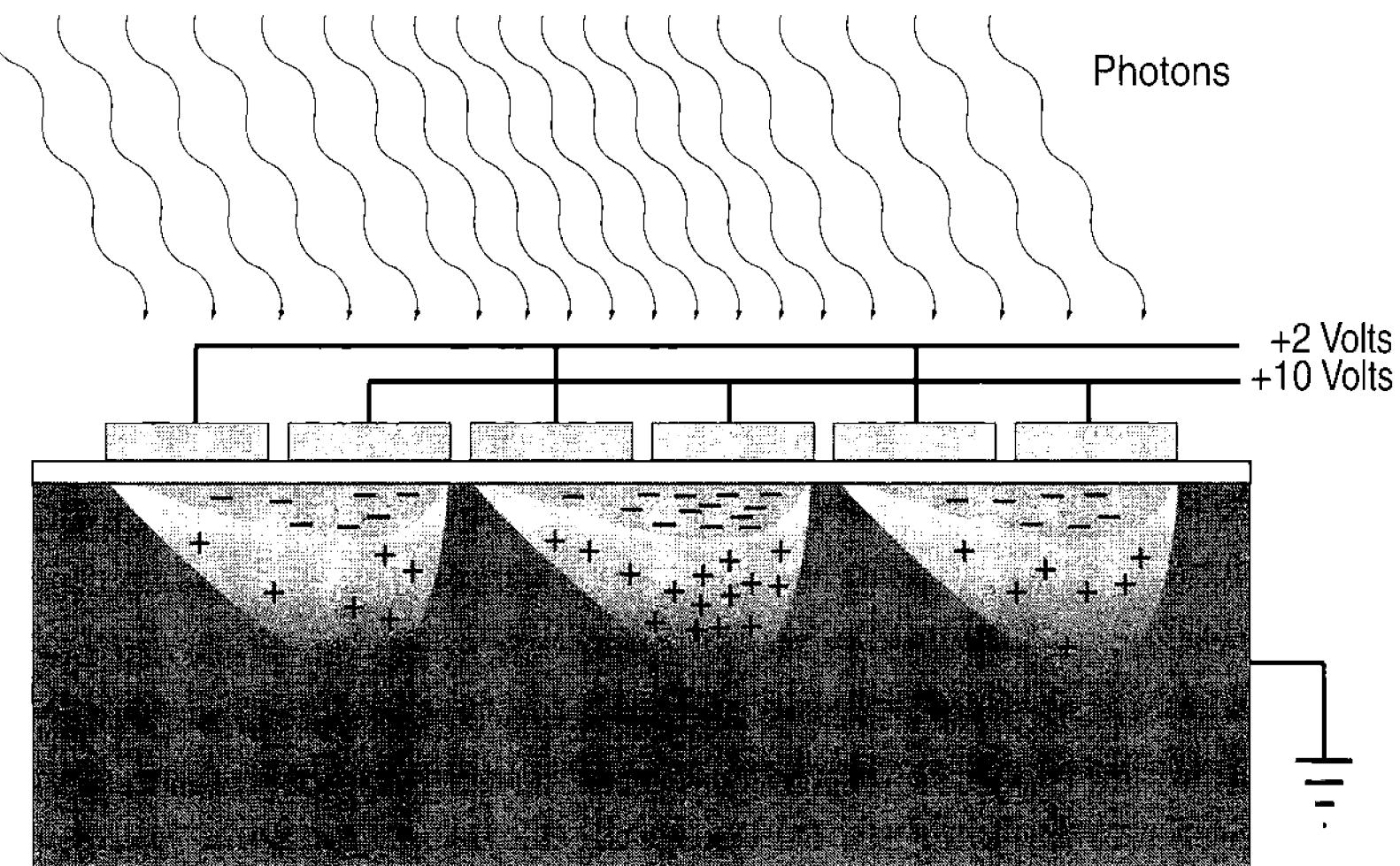


Figure 1.9     CCDs consist of adjacent charge wells, each collecting photo-electrons during integration. To shift the electrons down to a charge detection circuit on the CCD chip, the charge on the electrodes is cycled so that accumulated photo-electrons move from one charge well to the next.

linear as long as the photosites retain at least half their original charge.

Photosites on a CCD are arranged in columns and lines. The size of a photosite ranges from 3 to 25 microns depending on the design of the detector. The lower limit is set by manufacturing difficulties, and the practical problem is that small photosites have small collection areas and therefore intercept few photons, and they can hold only a limited amount of charge. At the upper end, there is no need for photosites that are too large to capture all of the information present in images formed by camera lenses and telescopes.

CCD imagers range in size from 1.3 mm to approximately 70 mm across the diagonal, containing arrays of  $102 \times 102$  to  $8192 \times 8192$  photosites—and even bigger CCDs are being designed all the time. Unfortunately for amateur astronomers, the price of CCDs rises exponentially with the physical size of the chip. Typical CCDs for amateur use measure 4 to 24 mm across the diagonal, contain between 32,000 and 12,580,000 photosites, and cost between \$60 and \$10,000+. Although many CCDs can be read out 60 times per second, astronomical CCD cameras are seldom read faster than 50,000 photosites per second to allow precise measurement and digitization of the signal. Readout times vary from a fraction of a second to about 60 seconds to read a complete image.

The electronics industry makes CCDs in a variety of configurations, from robust camcorder CCDs designed for readout at 60 fields per second, to digital camera CCDs designed to produce sharp megapixel images, to science-grade sensors optimized for high quantum efficiency and low noise. The configurations encoun-

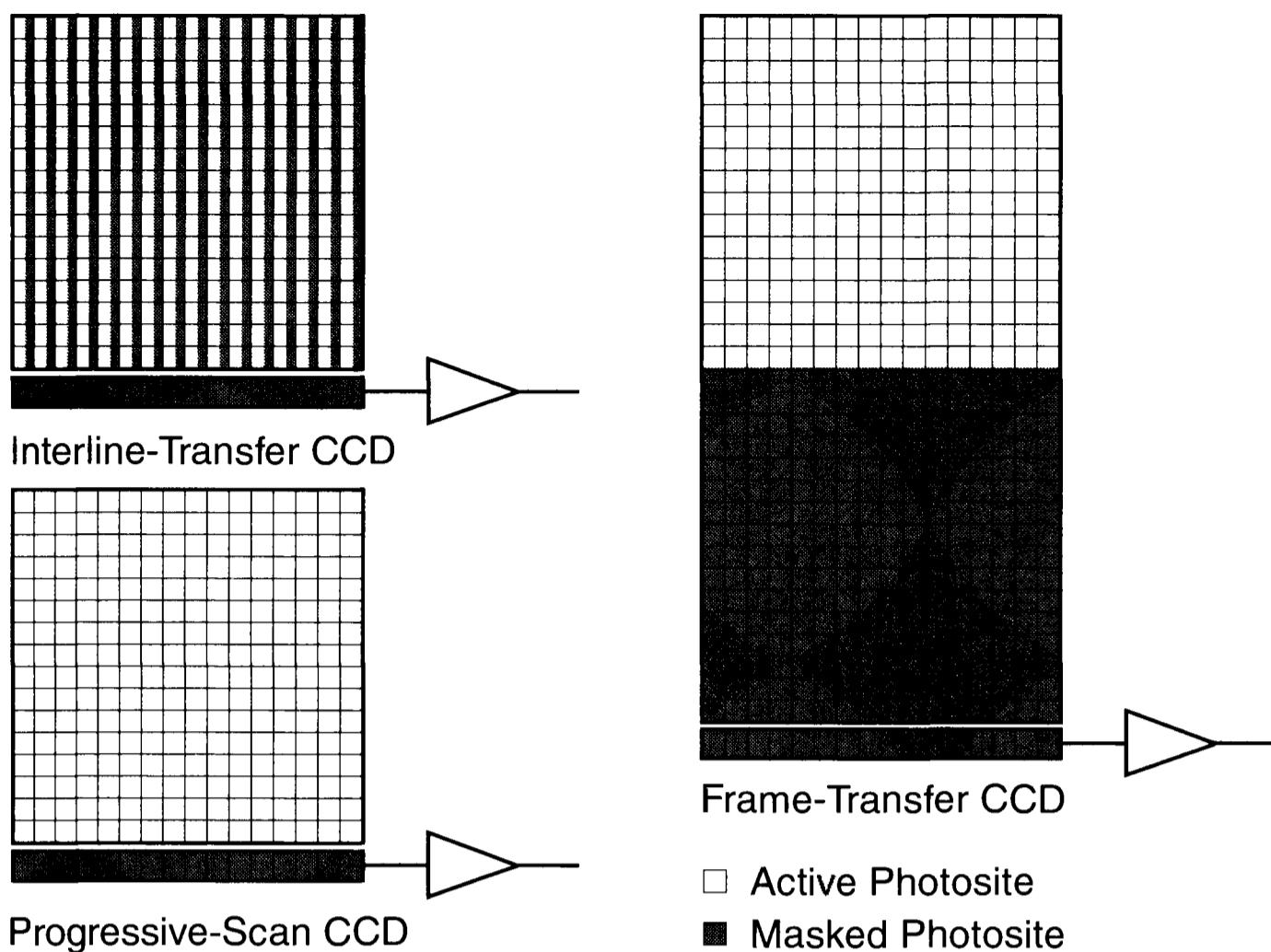


Figure 1.10 CCDs used in astronomy are almost always one of three types: interline-transfer, frame-transfer, and progressive-scan, shown schematically above. In all three types, charge is shifted line-by-line from the array to a serial register, and then shifted to a charge-sensitive amplifier.

tered most often in amateur astronomy are the progressive-scan, frame-transfer, and interline-transfer. In addition, CCDs can be either front-illuminated (as most are) or back illuminated (as are high-performance scientific CCDs).

Progressive-scan CCDs contain a rectangular array of photosites with a special row of high-capacity photosites called the serial register at the bottom line in the array. After exposure, the electrons in all of the columns in the array are “clocked” down one line, and the bottom line enters the serial register. The serial register is then clocked one element at a time into the charge detection node. When the serial register has been emptied, all columns are clocked again to refill the serial register. The sequence is repeated until all charges on the array have been read and digitized. Progressive-scan CCD cameras must have a shutter to prevent light from generating new photo-electrons during the readout phase.

Frame-transfer CCDs have the same basic architecture, except that the lower half of the array is covered by an opaque mask. After the exposure, all of the columns are clocked rapidly to move the accumulated charge in the top half of the array to its bottom half. The bottom of the array is then clocked and read out just like a progressive-scan CCD. Frame-transfer CCDs do away with the need for a mechanical shutter, since charge in the upper half of the array can be transferred to the lower half in about 1 millisecond.

Interline-transfer CCDs are designed for even faster shuttering than frame-transfer ones. In this configuration, alternate columns on the CCD are masked, and the electrodes overlying the CCD are arranged so that the charge in the uncovered

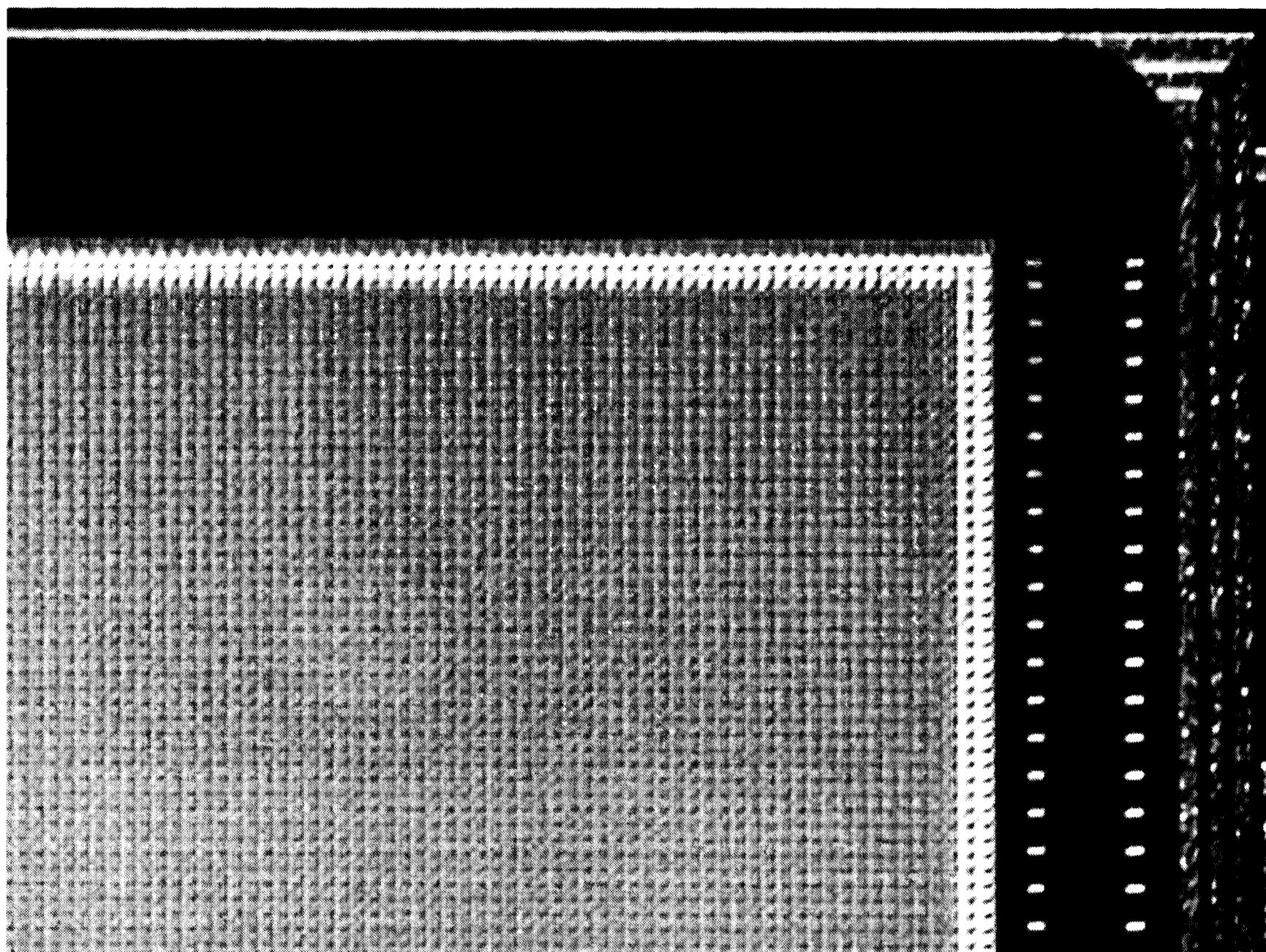


Figure 1.11 In this highly enlarged view of one corner of a CMOS sensor, you can see the regular array of individual pixels. In a CCD, accumulated charge from the whole chip is transferred to a single amplifier; in a CMOS device, each pixel has its own individually addressable amplifier.

columns can be clocked into the covered columns in a microsecond. The covered columns are then read out slowly. The primary difficulty with interline transfer is that half the detection area of the CCD is covered with masked columns, but a new generation of interline CCDs is being made with tiny integral lenses that redirect light from the masked columns to the active sensing columns.

The vast bulk of CCDs are front-illuminated models, meaning that the photosites are formed on a “thick” silicon wafer, and the electrically conductive gates necessary for charging and clocking charge are laid on top. This construction means that to reach the light-sensitive bulk silicon, photons must penetrate the gate structures, which are sometimes nearly opaque to short-wavelength (blue) light. Because of this, front-illuminated CCDs seldom exceed a peak quantum efficiency of 60% at the long-wavelength (red) end of the spectrum.

Back-illuminated CCDs are made the same way as front-illuminated ones, but the silicon wafer is etched to a thickness of around 10 microns, and the silicon mounted so that photons enter the array from the back side. Thinned back-illuminated CCDs are expensive, but their quantum efficiencies are high across the spectrum from 500 to 950 nm, with peak values approaching 90%.

One of the greatest advantages that CCDs enjoy is that the same detector is used again and again, and the output is highly repeatable. This means that the odd-

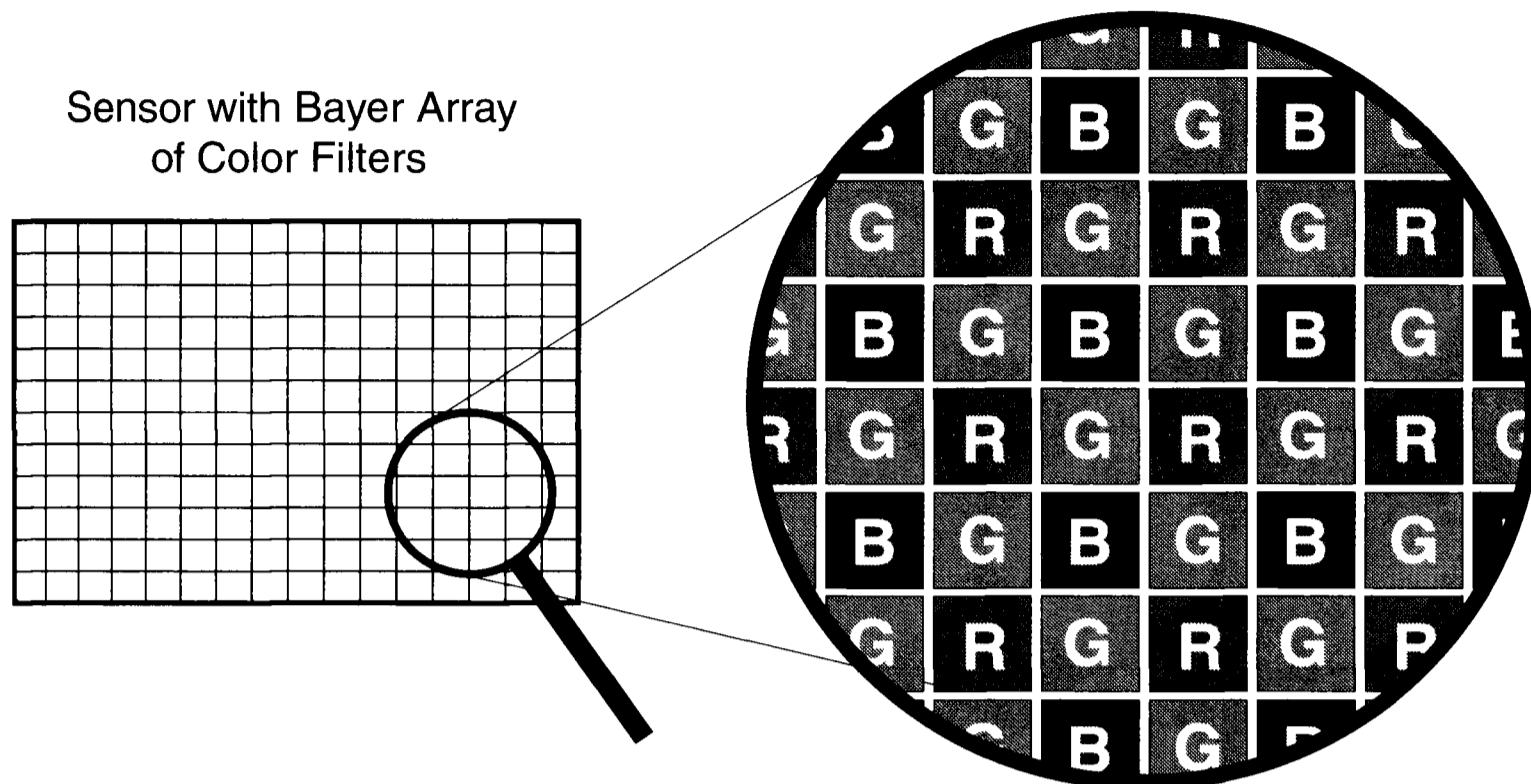


Figure 1.12 For color images, electronic sensors can be made with a checkerboard array of red, green, and blue filters (a Bayer Array). Each filter covers just one photosite, so the resulting raw image has a checkerboard of pixels made with different filters. The final color image must be reconstructed in software.

ball characteristics of a particular CCD can be calibrated out. An image from a CCD contains the following components:

- a bias voltage that is constant,
- systematic variations in the bias voltage,
- random variations in the bias voltage,
- a temperature-dependent dark current,
- systematic variation in the dark current,
- random variations in the dark current,
- a random variation (readout noise) in the output amplifier, and
- a signal generated by photons falling on the CCD.

In addition, photosites vary from one to the next in quantum efficiency, so the sensitivity of the array is not constant. However, because you can use the same CCD over and over, all of the constant and systematic effects can be removed. After calibration, a CCD image faithfully reproduces the amount of light that fell on each photosite in the array.

**CMOS Devices.** Like the CCD, the CMOS device consists of a large array of photosites on a silicon substrate. Unlike the CCD, however, the photosites in a CMOS device are individually addressable; that is, by activating a grid of conductors, any photosite can be read at any time. Instead of reading out the entire image, a small group of photosites can be read; or the CMOS sensor can be read out one line at a time while the remaining lines continue to accumulate signal.

Offsetting these advantages, however, each photosite must have an amplifier to sense charge and transistor switches to return the output signal. The auxiliary

## Chapter 1: Basic Imaging

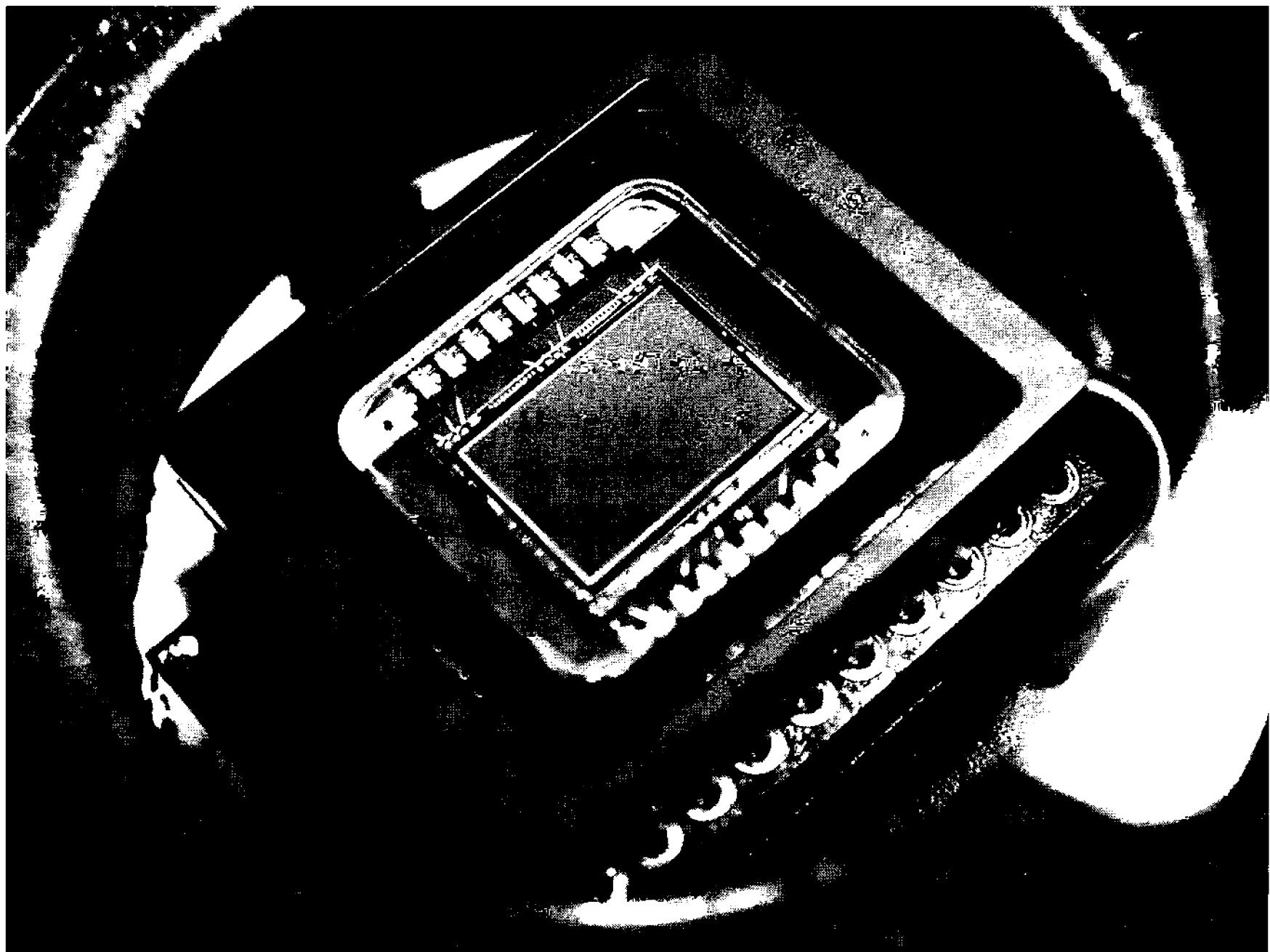


Figure 1.13 In CCD cameras designed for astronomical imaging, the CCD is isolated in a sealed chamber behind a clear glass window. This allows the detector to be cooled, thereby reducing its dark current, while avoiding problems from water condensing and freezing on the detector.

electronics take space that on a CCD would be collecting light, and hard-to-control variations in amplifier characteristics make CMOS devices less uniform than CCDs. Nevertheless, CMOS devices have proven their worth in webcams and a growing number of digital cameras.

From a practical point of view, once an electronic sensor has passed an image to the computer, the image data from CCDs and CMOS devices look very nearly the same. To get the best possible results from either type of detector, it is necessary to make calibration frames and apply them to raw images.

**Color Imaging with Electronic Sensors.** Although they are intrinsically sensitive to light over a wide range of wavelengths, CCDs and CMOS devices are monochrome sensors; that is, they record total incident flux of photons with no color information. To obtain color, observers must use one of three methods:

1. Make separate exposures through color filters—usually red, green, and blue. Each of the filtered images records the photon flux in one band of wavelengths, or one color channel. Observers often back up a tricolor set of images with an unfiltered luminance image that records all three color channels. To construct a color image, the three separated color channels and luminance must be merged into a single image.

2. Make a single exposure using a CCD or CMOS device with an integral color filter matrix, usually called a Bayer array. The filter array is a tiny checkerboard of red, green, and blue filters, each large enough to cover just one photosite. Thus, one exposure records information for all three color channels, at the expense of reduced image sharpness. To construct a color image, the image data must be resampled to provide every pixel with all three color channels.
3. Make a single exposure using a special CMOS device that has three sensing layers. The top layer responds to blue, the middle layer to green, and the bottom layer to red. At the time of this writing, multi-layer sensors are a new technology and not yet available for use in astronomy.

Making separate filtered exposures is by far the most flexible technique because it allows the observer to select a set of filters suited to the imaging task at hand. A sensor with a Bayer array is, however, the easiest way to make color images because “generic” red, green, and blue filters are built into the detector.

### 1.3.4 Linearity, Saturation, and Blooming

Astronomers prize CCDs for their linear response to light, that is, the output signal is directly proportional to the number of photons that fell on each photosite during an exposure. When the number of photoelectrons accumulating during an exposure reaches the holding capacity of the charge well, the photosite is said to be saturated. Ideally, when a CCD reaches saturation, it would cease responding to further photons. Unfortunately, that does not happen.

Bright stars in the CCD field of view continue to generate photoelectrons, and eventually the excess electrons overflow the charge wells and spill into adjacent photosites. On the finished image, a bright star displays streaks called *blooming trails* extending from the star image. Science-grade CCDs are linear, and as a consequence, most of them suffer from blooming trails from bright stars.

In camcorders and digital cameras, blooming trails are unacceptable. To combat blooming, CCD manufacturers incorporate electronic *drains* called *anti-blooming gates* to absorb overflow electrons during heavy exposure. With an active anti-blooming gate, it is possible to make a multi-minute exposure of a brilliant star without blooming trails. Unfortunately, the anti-blooming gate drains away electrons even before a photosite reaches saturation. Instead of trending steadily upward as the exposure increases, the number of photoelectrons rolls off—giving the CCD a nonlinear response to light.

CMOS devices tend to be free of blooming trails because the photosites are relatively independent of one another, but the small and necessarily simple amplifiers in CMOS photosites are themselves nonlinear.

## 1.4 Sensor Geometry

Images formed by a pinhole camera, lens camera, or telescope consist of differing

## Chapter 1: Basic Imaging

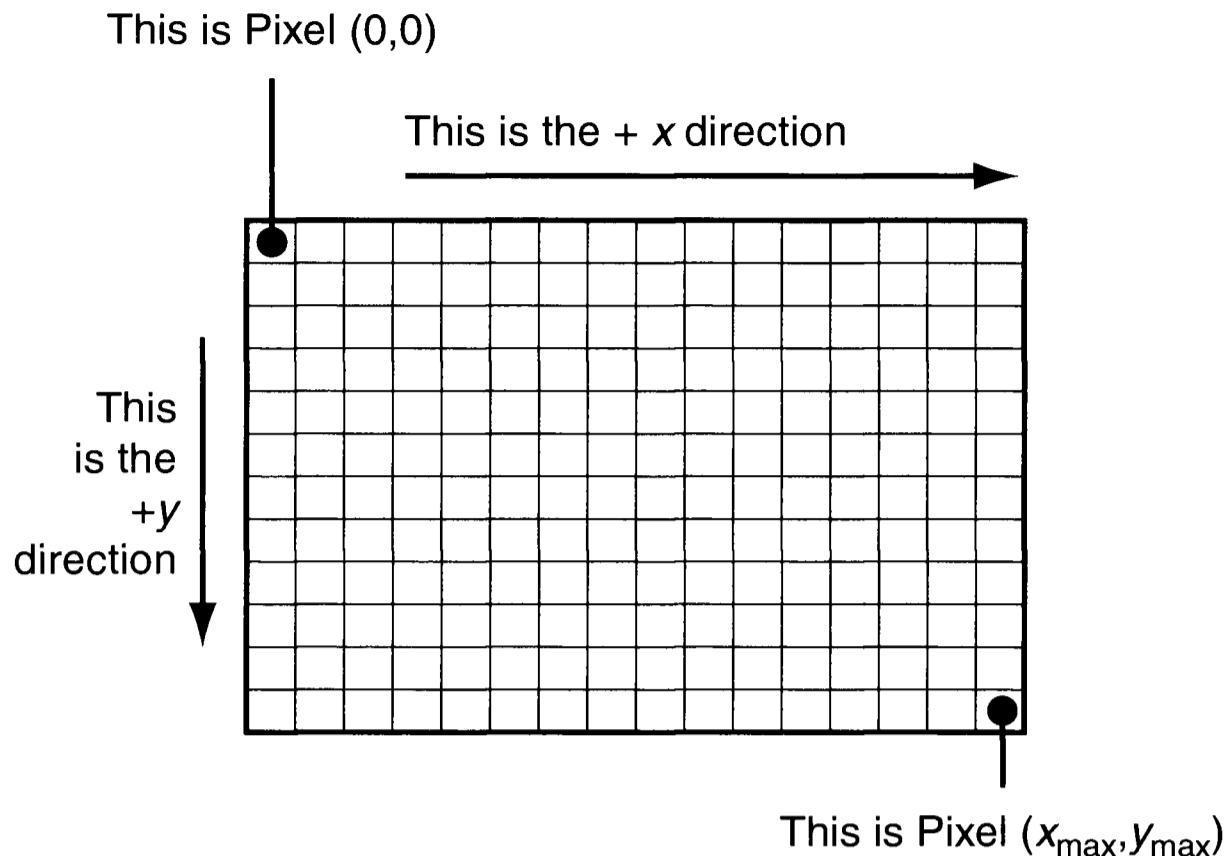


Figure 1.14 Each pixel in a digital image has an “address” corresponding to its place in the image and the location of a photosite on the detector. The first pixel to be read from the detector is pixel (0,0) and the last is pixel ( $x_{\max}, y_{\max}$ ). The image shown here is 16 pixels wide by 12 pixels high.

amounts of light organized by their angular positions relative to the optical axis of the camera. At any given location, the intensity of the image corresponds to the amount of light coming from some particular direction. Detectors break the image into thousands or millions of discrete areas, each represented by a pixel in the resulting digital image.

*Pixel* means picture element. Pixels are to digital images what photosites are to CCDs. A pixel may correspond to a photosite on the CCD, or if the chip is binned, a pixel may correspond to two or more photosites on the CCD. Each pixel in an image has three key properties:

- its *x*-axis or column address,
- its *y*-axis or row address, and
- its pixel value.

The *x*-axis location is the pixel’s location in the coordinate that is clocked more rapidly from the CCD. By custom, the *x*-axis is displayed horizontally on computer screens. It may also be called the *i*-axis, or the *sample* axis.

The *y*-axis location is the pixel’s location in the coordinate that is clocked more slowly from the CCD. By custom, this axis runs vertically on computer screens. It is sometimes called the *j*-axis, or the *line* axis.

The numerical value of a pixel (its *pixel value*) is encoded in bytes stored on the computer’s hard disk. Pixel value is a property of pixels, just as mass is a property of matter. Pixel value can be expressed in different units of measurement, just as mass can be measured in grams, atomic mass units, etc. In raw images, where the pixel has obtained its numerical value directly from the output of the CCD

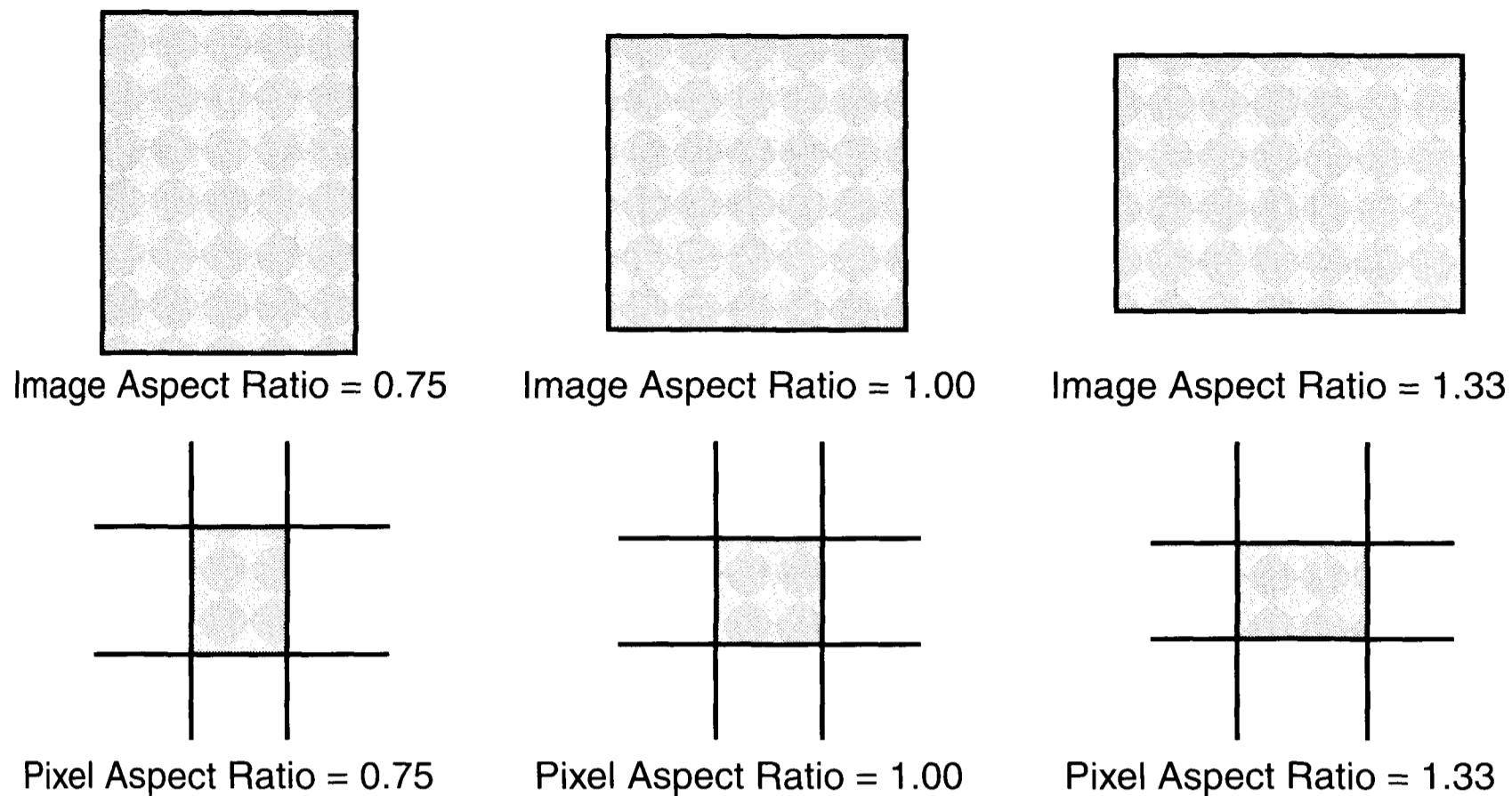


Figure 1.15 Aspect ratio is defined as width divided by height. For images, the width of the entire image is divided by its entire height. For pixels, the width of a single pixel is divided by its height. The wider the image or pixel is, relative to its height, the greater is its aspect ratio.

camera's analog-to-digital converter, the units of pixel value are ADUs (analog-to-digital units) or DN (data number).

ADUs can later be converted into pixel values expressed in units of electrons, ergs/cm<sup>2</sup>/sec., or any other physical unit of measurement you want the pixels in your images to have.

In color images, the pixel value is usually expressed as numerical values of the three additive color primaries: red, green, and blue. These *RGB triads* may be encoded as 8-bit integers (256 gray levels), as 12-bit integers (4096 gray levels), or 16-bit integers (65,536 gray levels), depending on the camera and the intended use for the image.

#### 1.4.1 Aspect Ratio

To reconstruct the image sampled by the sensor, it is necessary to know how the mass of bytes is organized. This topic is treated fully in Chapter 3. This information is usually conveyed from the electronic camera to the computer in a file header. When a computer program opens an image, it begins by reading the header. The image can then be reconstructed on a computer monitor in the pattern of columns and rows matching that of the photosites. If this is done correctly, the computer screen will be a fairly faithful reproduction of the pattern of light that originally fell on the detector.

Aspect ratio defines an important characteristic of detectors: their shape. Electronic sensors have two types of aspect ratio: the *image aspect ratio* that defines the size of the entire image array on the sensor, and the *pixel aspect ratio*, which comes from the shape of the photosites.

## Chapter 1: Basic Imaging

The image aspect ratio,  $\alpha_{\text{image}}$ , is the ratio of the image width to the image height:

$$\alpha_{\text{image}} = \frac{\text{image width}}{\text{image height}}. \quad (\text{Eqn. 1.12})$$

Note that you must use the *physical* width and height of the image array, not the number of pixels, because the width and height of the pixel may differ. The same definition applies to photographic formats: the standard 35-mm film frame is 36 mm wide by 24 mm high, giving an image aspect ratio of 1.5.

The pixel aspect ratio,  $\alpha_{\text{pixel}}$ , is the ratio of the pixel width to the pixel height:

$$\alpha_{\text{pixel}} = \frac{\text{pixel width}}{\text{pixel height}}. \quad (\text{Eqn. 1.13})$$

Note that you must use the physical width and height of the pixel, usually available in the manufacturer's specification sheet. On Kodak KAF400 CCDs, for example, two 9-micron-wide photosites are often binned to form a pixel 18 microns wide, and two 9-micron-high photosites are binned to form an 18-micron pixel, so the pixel aspect ratio is 1.00. On the Texas Instruments TC245, however, two 8.5-micron-wide photosites are binned to form a pixel 17 microns wide, but the 19.75-micron-high photosites are left unbinned, yielding a pixel aspect ratio, for the binned composite pixel, of 0.8608.

### 1.4.2 Pixel Count

Given the pixel count and the physical width of the detector, you can find the pixel size by dividing the detector width by the pixel count:

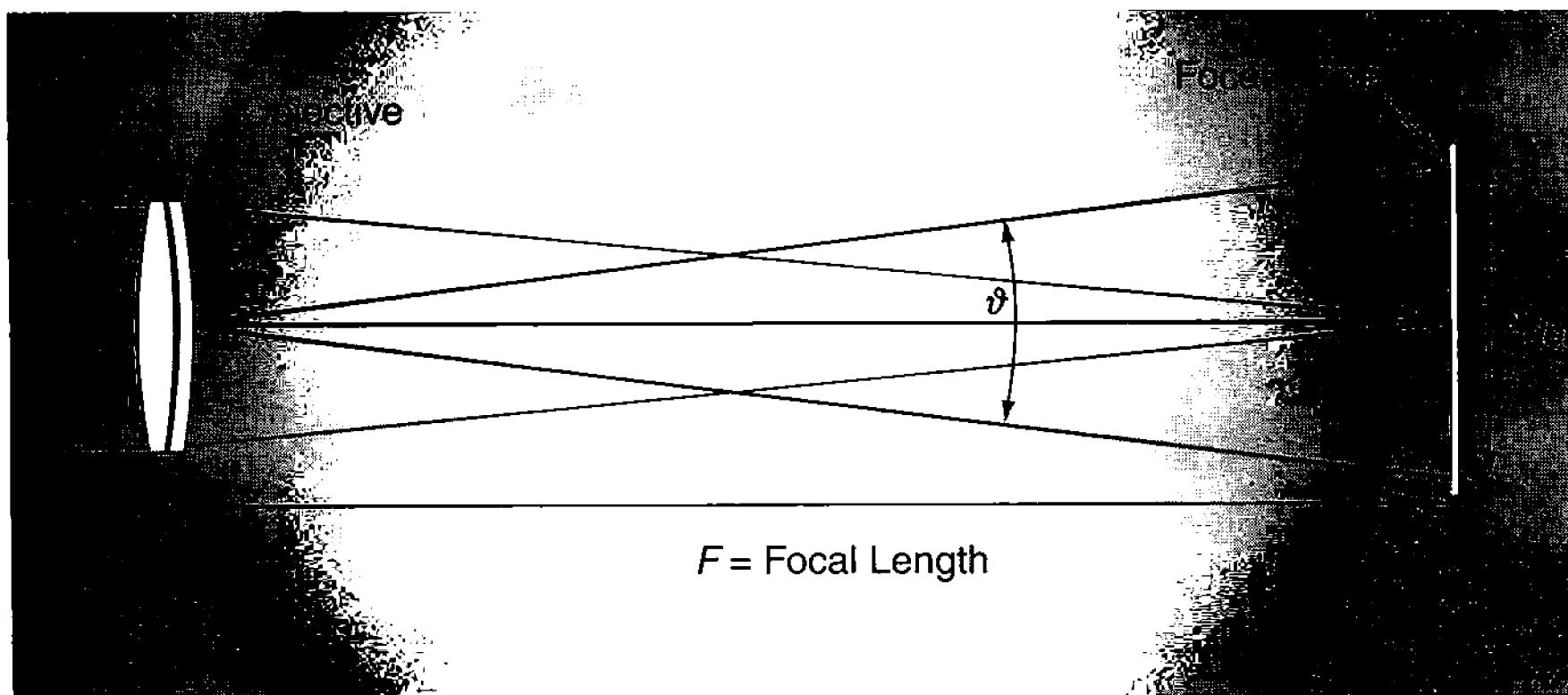
$$d_{\text{pixel}} = \frac{d_{\text{CCD}}}{N}. \quad (\text{Eqn. 1.14})$$

To find the width of the photosites in a webcam with a  $4 \times 3$  mm detector listed as having  $640 \times 480$  pixels, divide 4 mm by 640 pixels:

$$d_{\text{pixel}} = \frac{4}{640} = 0.00625 \text{ [millimeter]}, \quad (\text{Eqn. 1.15})$$

or 6.25 microns. The same holds true for the pixel height, and in this case, the pixels are 6.25 microns square. Given the detector size and pixel size, you can find the pixel count. For the TC245 CCD, Texas Instruments gives an image width of 6.4 mm and an image height of 4.8 mm. With binned pixels 17 microns wide by 19.75 high, the pixel count is 377 pixels wide by 243 pixels high.

Most science-grade CCDs are made with pixels having the same width and height, and they often have pixel counts that are some power of two: 256 x 256, 512 x 512, 1024 x 1024, or 2048 x 2048. These devices have an image aspect ratio of 1.000 and a pixel aspect ratio of 1.000. (When the pixel aspect ratio is 1, the



**Figure 1.16** Whether it's a piece of photographic film or a CCD chip, a detector having a dimension  $d_{\text{det}}$  on a telescope of focal length  $F$  covers an angular field of view  $\vartheta$ . To cover a larger field of view, either the detector must be made larger or the focal length of the telescope must be made shorter.

device is said to have *square pixels*.) These dimensions make geometric computations easier, and allow easy use of a spatial-filtering technique called the Fast Fourier Transform.

Older CCDs manufactured for use in video cameras, such as the TC241 and TC245, usually have an aspect ratio of 4:3, or 1.3333—an image aspect ratio derived from the original 24-mm wide by 18-mm high Kodak/Edison 35-mm movie film frame. When television appeared, it retained the 4:3 aspect ratio, so that CCDs made for use in television cameras also have that aspect ratio. The pixel dimensions are based on standard interlaced video framing.

Many of the current generation of CCD- and CMOS-based video cameras and webcams employ the 4:3 image aspect ratio with square pixels, so they are compatible with VGA and XGA computer displays. Pixel counts tend to fall on or close to the 4:3 ratio “magic numbers” of  $640 \times 480$ ,  $1024 \times 768$ ,  $1600 \times 1200$ , and  $2048 \times 1536$ . The Sony CCDs used in the Starlight XPress CCD cameras also have a 4:3 image aspect ratio, but some have pixel aspect ratios that depart slightly from square.

CCDs made for technical digital video and digital cameras, such as the Kodak KAF-series used in Santa Barbara Instrument Group's CCD cameras, use an image aspect ratio derived from the 3:2 image aspect ratio of the standard 35-mm film frame, but with square pixels and power-of-two pixel counts ( $768 \times 512$ ,  $1536 \times 1024$ , and  $3072 \times 2048$ ) in image height.

## 1.5 Image Capture

The most important characteristics of an astronomical image are its angular field of view and its angular pixel size. The field of view determines whether objects that you wish to record will fit inside a single image, and the resolution determines

## Chapter 1: Basic Imaging

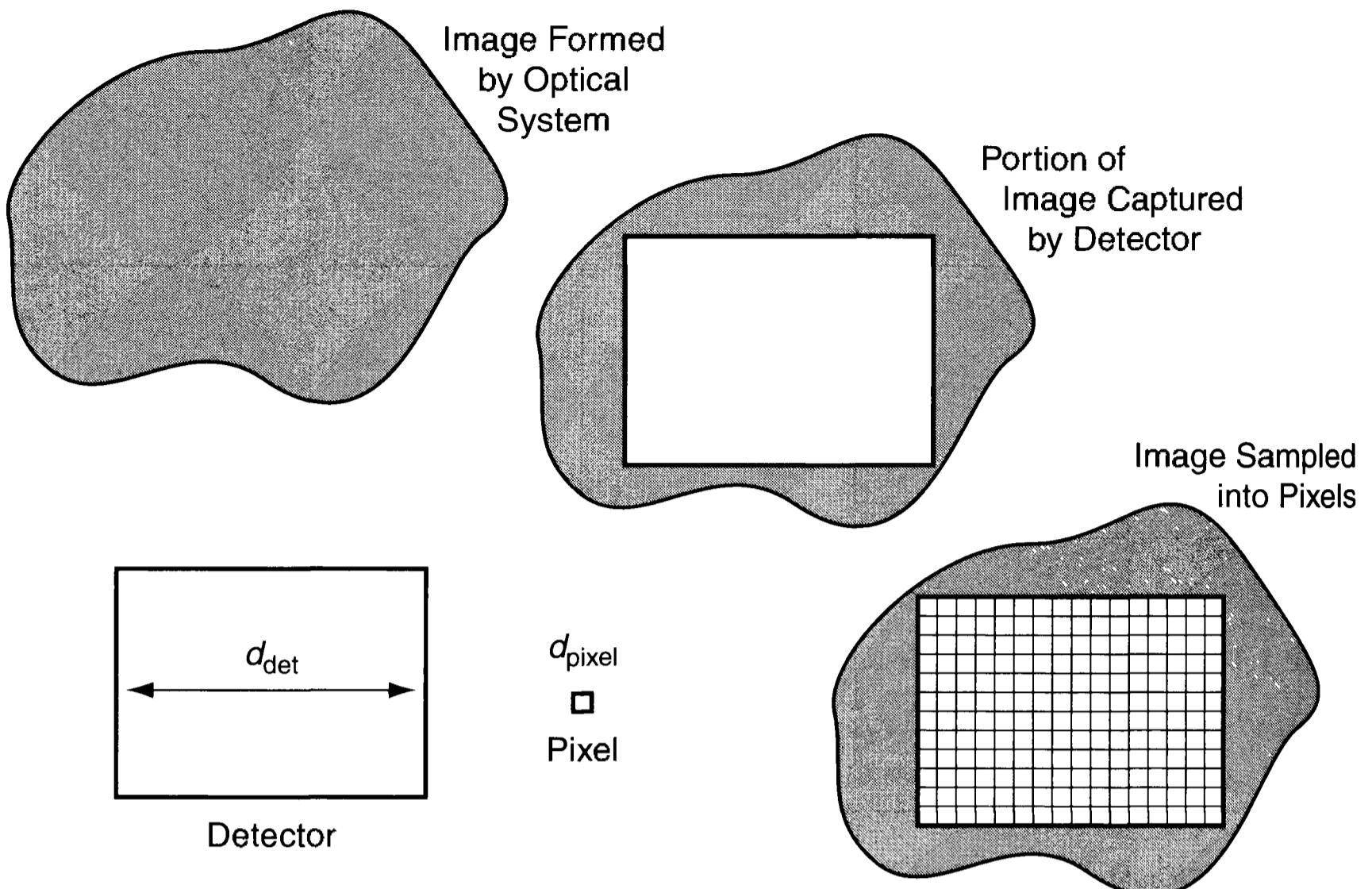


Figure 1.17 Electronic sensors capture the portion of the image that falls on them. If the image formed by a telescope is larger than the detector, the edges of the image are lost. During image capture, the photosite array on the sensor dissects the image into picture elements (“pixels”).

whether the photosites on the detector are the correct size to capture all of the desired image detail.

### 1.5.1 Angular Field of View of a Detector

The detector occupies an angle at the focus of a lens or telescope called the angular field of view. Recall that the image of an object is formed at a height  $h$  from the optical axis:

$$h = F \tan \vartheta, \quad (\text{Equ. 1.16})$$

where  $F$  is the focal length of the telescope and  $\vartheta$  is the angular distance from the optical axis. Assuming that the center of the detector, with a dimension  $d_{\text{det}}$ , is placed on the optical axis of the telescope, then  $d_{\text{det}} = 2h$ , and a detector captures a field of view  $\vartheta_{\text{fov}}$ :

$$\vartheta_{\text{fov}} = 2 \arctan \left( \frac{d_{\text{det}}}{2F} \right) \text{ [radians]} \quad (\text{Equ. 1.17})$$

for a telescope with a focal length  $F$ . (Note: to convert radian measure to degrees, multiply by 57.3; to minutes of arc, by 3438; and to seconds of arc, by 206,265.)

As an example, consider a CCD detector that measures 6.4 mm wide by 4.8 mm high placed on a telescope with a focal length of 1,000 mm. What is the field

of view of the detector on this telescope?

Perform the calculation for the 6.4-mm width as follows:

$$\vartheta_{\text{fov}} = 2 \arctan\left(\frac{6.4}{2 \times 1000}\right) \quad [\text{radians}] \quad (\text{Equ. 1.18})$$

$$\vartheta_{\text{fov}} = 2 \arctan(0.0032)$$

$$\vartheta_{\text{fov}} = 0.36669^\circ = 22'0''.$$

For those unfamiliar with trigonometric functions, most scientific calculators will perform the inverse tangent function and return the answer in degrees.

For the 4.8-mm height, perform the calculation as follows:

$$\vartheta_{\text{fov}} = 2 \arctan\left(\frac{4.8}{2 \times 1000}\right) \quad [\text{radians}] \quad (\text{Equ. 1.19})$$

$$\vartheta_{\text{fov}} = 2 \arctan(0.0024)$$

$$\vartheta_{\text{fov}} = 0.27502^\circ = 16'30''.$$

For the CCD detector and telescope in the example, the field of view is 22'00" wide by 16'30" high. You can compare these dimensions with a star atlas to see whether objects that you want to record will fit in a single image. Note that these equations work equally well for photographic film and electronic detectors.

### 1.5.2 Sampling the Image

Detectors do not reproduce images *in toto*, but rather, they *sample* the image. This means that the image is broken into discrete small chunks. With electronic sensors these are called *picture elements*, or *pixels*. The pixel structure dictates the smallest features that will be visible in a digital image.

Electronic sensors sample images in a very simple way: photons that fall on a photosite are lumped together as a single pixel value in the image. The CCD thus samples the image at the focus of the telescope in a regular grid, with each element in the grid represented by a single numerical value.

The number of photons captured by a photosite is proportional to the flux of photons times the collecting area of the photosite, and the number of electrons generated is the product of the quantum efficiency and the number of photons. Given that  $n$  is the average number of electrons generated, because the photons arrive at random, the number of electrons generated by a photosite during any particular integration is  $n \pm \sqrt{n}$ , and the signal-to-noise ratio is  $\sqrt{n}$ .

The ability of CCDs to store charge during an exposure ranges from 30,000 to 500,000 electrons in the charge well; so the expected random variations are 173 and 707 electrons, respectively. In addition, reading the charge from the CCD

## Chapter 1: Basic Imaging

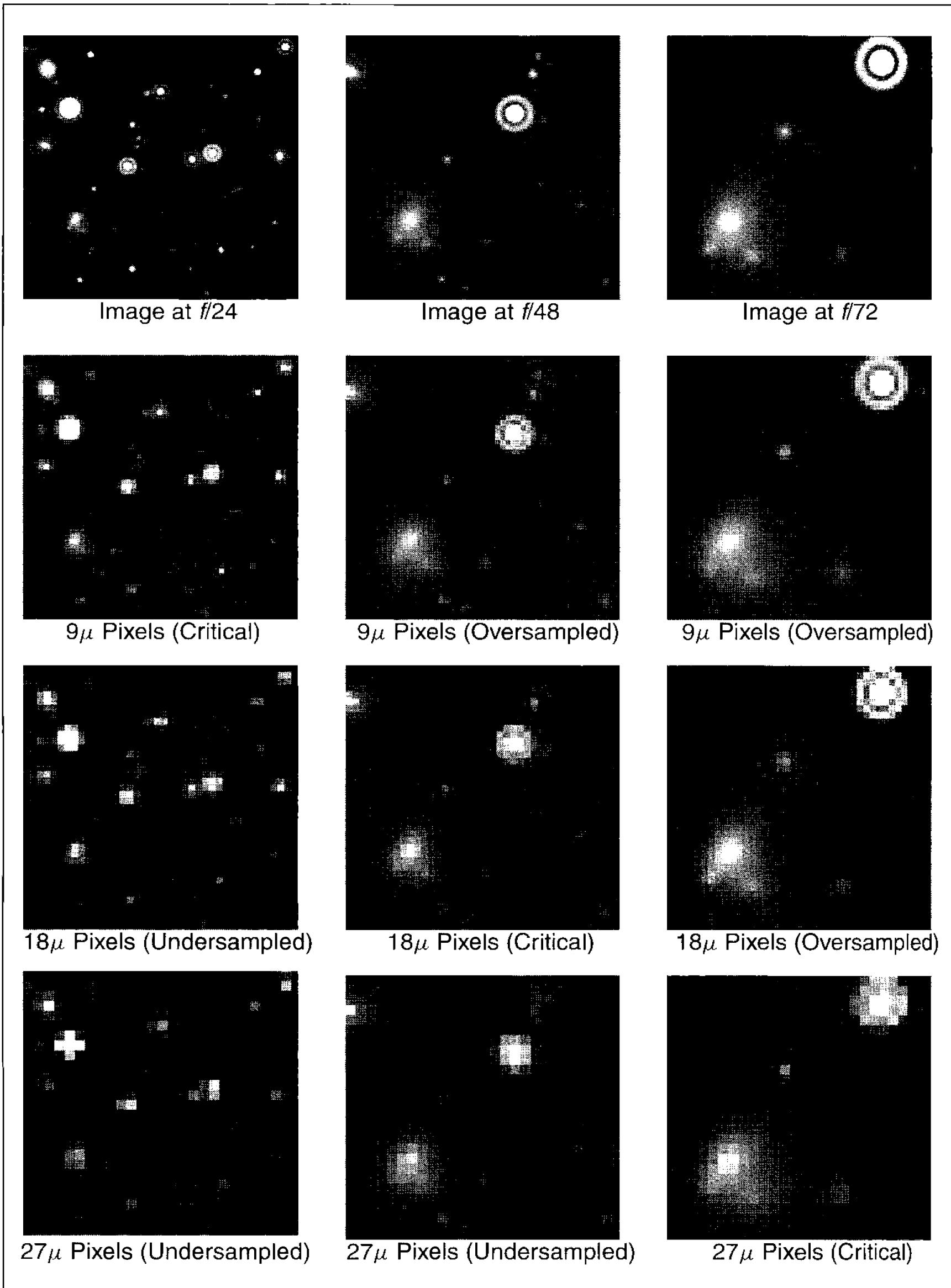


Figure 1.18 Matching pixel size and telescope focal ratio is essential for high-resolution imaging. The top row shows highly enlarged images at the focal plane of telescopes used at  $f/24$ ,  $f/48$ , and  $f/72$ , with an effective wavelength of 730 nm. The succeeding rows show the same images as captured by detectors with 9-, 18-, and 27-micron pixels. At a focal ratio of  $f/24$ , 9-micron pixels satisfy the Nyquist sampling criterion, with two pixels across the Airy disk. With 18-micron pixels, the image is critically sampled at  $f/48$ , and with 27-micron pixels, critical sampling occurs at  $f/72$ . Images with finer sampling are called oversampled; those with coarser sampling are called undersampled.

adds 6 to 50 electrons of random noise. When we apply the photometric bit-depth equation, we find that a normally exposed CCD image contains from 7 to 8 bits of useful information. However, to distinguish  $n$  gray levels containing that useful information requires a 12-bit to 16-bit analog-to-digital converter.

### 1.5.3 Angular Size of a Single Pixel

The equations for calculating the angular size of a photosite on the CCD and the corresponding pixel in the image are the same as those in the preceding section; although, of course, the pixel angles are much smaller.

We continue the example above, noting that the photosites on the CCD are 17 microns wide by 19.75 microns high. Set up the equation as follows:

$$\vartheta_{\text{fov}} = 2 \arctan\left(\frac{0.017}{2 \times 1000}\right) \quad [\text{radians}] \quad (\text{Equ. 1.20})$$

$$\vartheta_{\text{fov}} = 0.000974^\circ = 3.5''.$$

If you carry out the second calculation, you will discover that the pixels are 3.5 arcseconds wide by 4.07 arcseconds high.

### 1.5.4 Matching Pixels to the Point-Spread Function

The point-spread function of a telescope (ideally, the Airy disk) defines a characteristic dimension for the smallest details in a telescope image. To reproduce all of the detail present in the image, the sample size must be small enough to define the bright central core of the diffraction disk reliably. The Nyquist sampling theorem in communication theory states that in sampling a wave, the sampling frequency must be two times the highest frequency present in the original. Music recorded on CDs is therefore sampled at 44 kHz, a bit more than twice the highest frequency (20 kHz) most people can hear.

Applied to image sampling, the Nyquist theorem suggests that the size of a pixel must be half the diameter of the diffraction disk as defined by its full-width half-maximum dimension. Images sampled at this rate are called “critically sampled,” because the image has been broken into *just enough* pixels to capture all detail in the image.

Images sampled with pixels larger than half the full-width half-maximum of the diffraction disk are *undersampled*, because some of the fine structure in the telescope image will be lost. Undersampling is not necessarily a bad thing, since it may be a trade-off necessary to cover a large field of view.

Images sampled with more than two pixels across the core of the diffraction disk are *oversampled*. Oversampling with three to five pixels across the diameter of the full-width half-maximum diffraction disk insures that none of the information present in the continuous telescopic image is lost because the image is broken into discrete samples.

Atmospheric turbulence, telescope shake, poor guiding, and slightly out-of-

## Chapter 1: Basic Imaging

focus images often enlarge the diffraction disk to many times the size of the Airy disk. A practical observer matches the pixel size not to the Airy disk, but to the size of the best seeing encountered at the observing site.

To match pixel size and diffraction disk, recall the formula for  $d_{\text{FWHM}}$ , the diameter of the core region of the diffraction disk:

$$d_{\text{FWHM}} = 1.02\lambda \frac{F}{A} = 1.02\lambda N, \quad (\text{Equ. 1.21})$$

where  $A$  is the aperture of the telescope,  $F$  is the telescope focal length, and  $N$  is its focal ratio. Because the real point-spread function  $d_{\text{PSF}}$  is always equal to or larger than the Airy disk of a perfect telescope, the following holds:

$$d_{\text{PSF}} \geq d_{\text{FWHM}}. \quad (\text{Equ. 1.22})$$

Since the condition for critical sampling is that two pixels of dimension  $d_{\text{pixel}}$  must equal the linear dimension of the diffraction disk at the focus of the telescope,

$$2d_{\text{pixel}} = d_{\text{PSF}}. \quad (\text{Equ. 1.23})$$

This implies that the minimum focal length,  $F_{\min}$ , required for critically sampling a telescopic image is:

$$F_{\min} = \frac{Ad_{\text{pixel}}}{0.51\lambda}, \quad (\text{Equ. 1.24})$$

and that the minimum focal ratio,  $N_{\min}$ , necessary for critically sampling a telescopic image is:

$$N_{\min} = \frac{d_{\text{pixel}}}{0.51\lambda}. \quad (\text{Equ. 1.25})$$

To better understand the implications, consider an example: you have a telescope with an aperture of 200 mm, a focal length of 2,000 mm, a CCD camera with 9 micrometer photosites, and you want to take diffraction-limited images in red light at a wavelength of 630 nanometers. What is the optimum focal length?

Converting these units to meters and substituting into the above:

$$F_{\min} = \frac{200 \times 10^{-3} \times 9 \times 10^{-6}}{0.51 \times 630 \times 10^{-9}}, \quad (\text{Equ. 1.26})$$

works out to  $F_{\min} = 5.6$  meters, or 5,600 mm. Under conditions of perfect seeing, using a 3x Barlow lens to raise the telescope's focal length of 2,000 mm to 6,000 mm would give you a slightly longer focal length than the minimum required for critical sampling. However, if a combination of seeing, drive errors, and telescope shake were to triple the effective diameter of the point-spread function, the images would be critically sampled at the  $f/10$  focus of the telescope.

Here is another example: you want to take diffraction-limited images of Jupiter in green light (550 nanometers) with a camera that has a CCD with 12-mi-

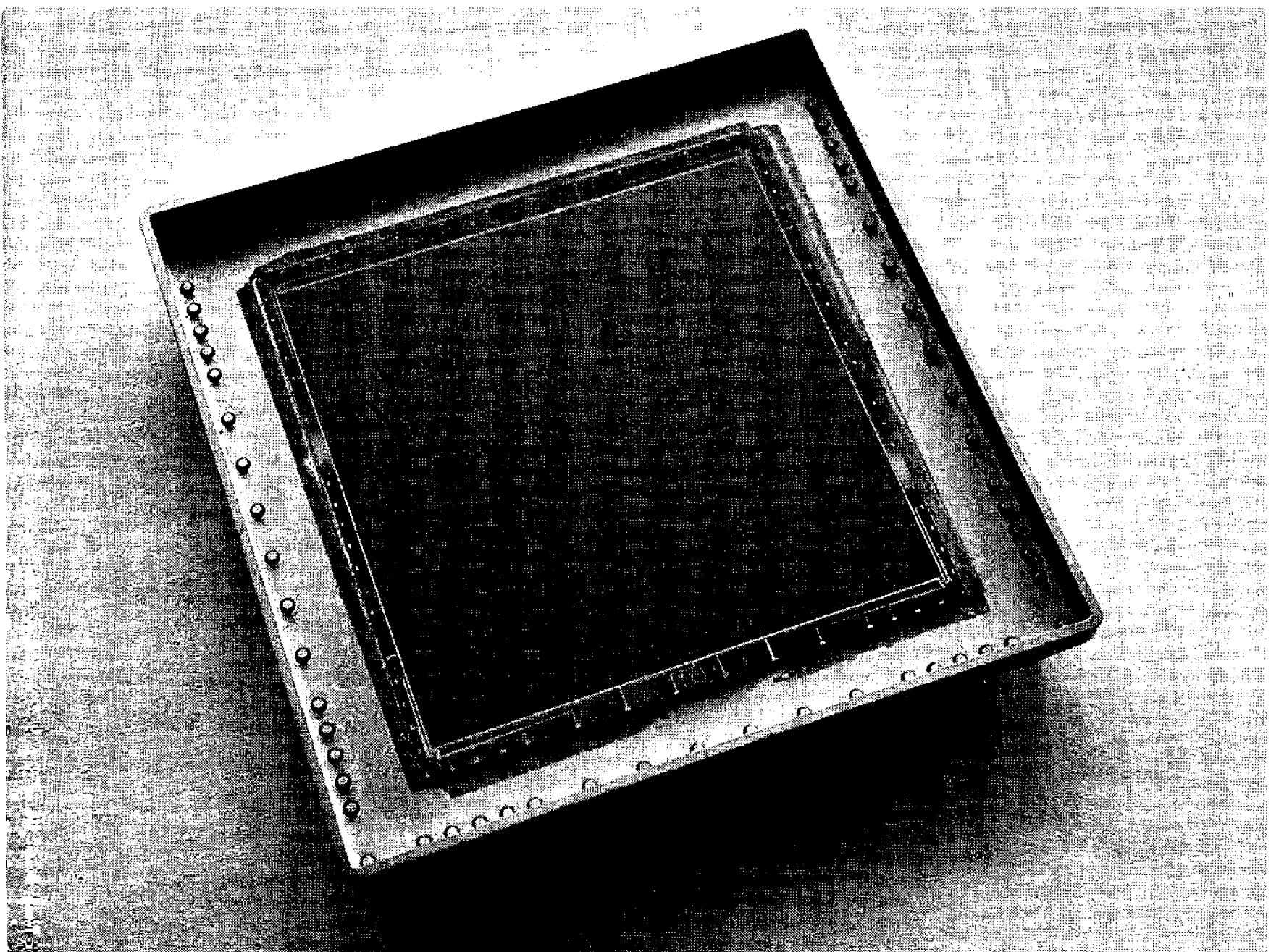


Figure 1.19 To match the focal length and seeing conditions at large modern telescopes, professional astronomers use CCDs that have large pixels and large pixel counts. This CCD is 50 mm square, and has 24 micron pixels. On an amateur-size telescope, it would grossly undersample the images.

micron pixels using the excellent optics of your 16-inch  $f/6$  Newtonian. What is the best focal ratio to use?

Set up the solution by converting to meter units and substituting:

$$N_{\min} = \frac{12(24) \times 10^{-6}}{0.51 \times 550 \times 10^{-9}}. \quad (\text{Equ. 1.27})$$

The result is  $N \geq 43$  for critical sampling. To capture all the detail present in the image on a night of exquisite seeing, you will need to enlarge the image of the planet 7 times using eyepiece projection from the  $f/6$  focus.

On nights of poor seeing, of course, you would use a lower focal ratio because the blurry image of a star would be much larger than a perfect diffraction disk. By dropping to  $f/30$  or even  $f/20$ , you could use shorter integration times, thereby raising your chances of “freezing” moments of steady air. Today’s web-cam and video observers acquire hundreds, or even thousands, of images at a critical-sampling focal ratio; scan them to find moments of best seeing; and then combine those best moments into a diffraction-limited image.

## Chapter 1: Basic Imaging

---

# 2 Counting Photons

---

Astronomy is about counting photons. During short exposures, only a few photons fall on a detector. Lengthy exposures mean that there's time for more photons to arrive—and you get a better picture. Because the photon supply imposes fundamental limits on imaging, this chapter explores what counting photons really mean in astronomy and astronomical imaging.

## 2.1 What Is a Signal?

*Signals* convey information that you want to listen to, look at, measure, or act on. A semaphore flag, the voice on a telephone, television pictures, and the pixel values in a CCD image are all signals. Each of these signals encodes data that you interpret as information, something meaningful. Two lanterns in the church tower were a meaningful signal to Paul Revere; the photons striking the pixels in your CCD camera is a meaningful signal to you.

In digital imaging, the signal reaches us as a stream of photons. The greater the number of photons, the brighter the star, nebula, or galaxy that you have recorded. As we shall see, it turns out to be significant that light comes to us as photons—discrete little packets of energy. To see how bright something out there is, we count how many photons our sensors detect in some specified time interval. That count is our signal. In this key respect, astronomical observation is nothing more than counting photons. To understand imaging and images, it is necessary to understand everything we can about counting photons.

## 2.2 What Is Noise?

Astronomers and engineers use the word “noise” in a special technical sense. To the layperson, noise is people yelling, bad music, the roar of traffic. In the vocabulary of science, *noise* is the random variation in a *signal*. Noise is present in all signals; you hear it in the hiss of a vinyl record, see it in the “snow” in the television signal from a weak or distant station, or more to the point, you experience it as graininess in CCD images.

Digital images are made by measuring the number of photons falling on a light sensor. The photons generate photoelectrons in the individual photosites

## Chapter 2: Counting Photons

(pixels) of the sensor, and the output from each photosite is proportional to the number of photoelectrons detected during the time you expose the sensor to light. At first, you might expect the number of photons to be exactly proportional to the length of the exposure. After all, when you put gasoline in the tank of your car, if you pump twice as long, you expect twice as much gas. Why are photons any different? The answer lies in the way that photons arrive—each reaching the sensor independent of any other photon. Instead of arriving in a steady flow like the ticking of a clock (“click-click-click-click”) photons arrive at random, like the irregular rattling sound of a Geiger counter (“click-a-click, clickitty, clack”) or the rattle of rain on a tin roof.

In normal daytime activities, so many photons bombard us that their arrival appears to be a steady flow—even though it is not. Only when the number of photons becomes small does their random arrival make a significant difference. Under a dark sky, the sky background signal usually consists of a few hundred photons per minute. Since it usually takes integration time of a minute or more to accumulate that many photons, you can count the number that arrive each second on the fingers of one hand.

Let’s see how this affects astronomical images. Suppose that you shoot ten images of a galaxy, one right after another. You bring the images up on your computer and measure the pixel value of the same spot in each image.

What do you see?

Instead of seeing the same value in each image, you measure a different number in each one. This is because each image is not a precise measure of the rate at which photons arrive, but instead is a *sample* of the photon stream—a snapshot recording the number of photons that arrived *during that particular exposure*. The number of photons varies from one sample to the next.

An astonishingly simple rule describes the expected variation:

- If the average signal consists of  $\bar{x}$  photons, an individual sample will contain  $\bar{x} \pm \sqrt{\bar{x}}$  photons.

But what does the notation “ $\bar{x} \pm \sqrt{\bar{x}}$ ” mean?

The “ $\bar{x}$ ” term, read as “x-bar,” is the *mean value*. It is the average rate over a long time interval, the mean value of a large number of samples. The “ $\sqrt{\bar{x}}$ ” term is the *standard deviation*, a statistical measure of the departure of a typical sample from the mean value. Finally, the “plus or minus” symbol, “ $\pm$ ”, means that an individual sample may be either larger or smaller than the mean value.

The notation “ $\bar{x} \pm \sqrt{\bar{x}}$ ” implies that the distribution of sample values about the mean follows a bell-shaped curve called the *normal distribution*. In the normal distribution:

- 68.3% of samples will lie between  $\bar{x} - \sqrt{\bar{x}}$  and  $\bar{x} + \sqrt{\bar{x}}$ ,
- 95.4% of samples will lie between  $\bar{x} - 2\sqrt{\bar{x}}$  and  $\bar{x} + 2\sqrt{\bar{x}}$ , and
- 99.7% of samples will lie between  $\bar{x} - 3\sqrt{\bar{x}}$  and  $\bar{x} + 3\sqrt{\bar{x}}$ .

In other words, roughly two-thirds of the time, the measured sample value

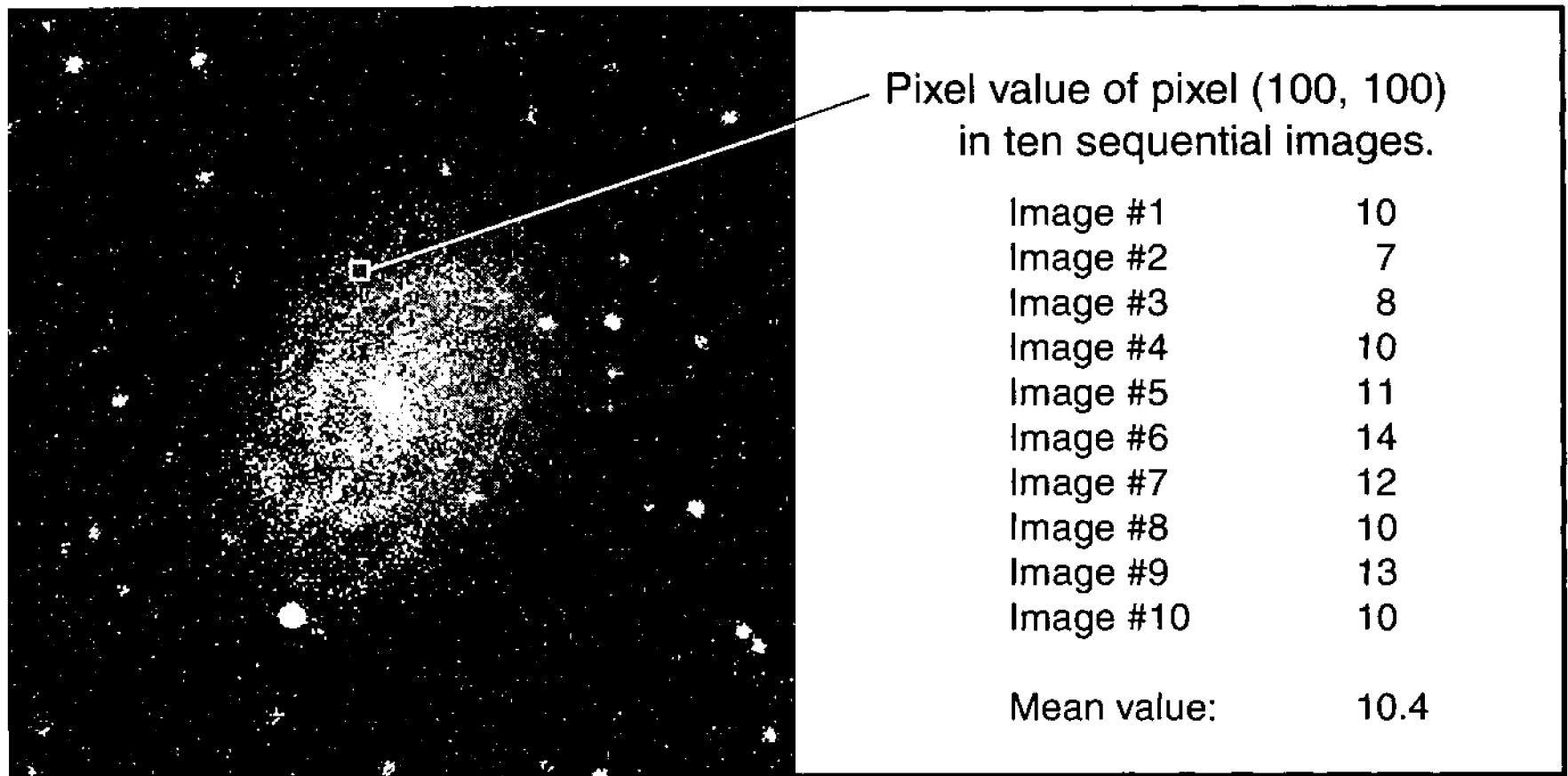


Figure 2.1 In a seemingly identical sequence of images, the number of photons that arrive during a constant exposure time varies randomly. The more photons, the smaller the percentage variation. To make an image, it is necessary to expose long enough to gather enough photons for a good image.

will differ less than one standard deviation from the mean value; and the other one-third of the time, the measured value will differ by more than one standard deviation from the mean value. However, departures greater than three times the standard deviation are rare, and departures greater than five standard deviations are *extremely* rare.

So—getting back to those ten pictures of the galaxy—what do you see?

Of course, you see the galaxy. In the pixels that comprise its image, you see pixel values scattered about the mean value that defines the galaxy. In the next section, we will dig deeper into the relationships between samples, signals, photons counts, and noise.

## 2.3 Signals and Noise

When we spoke about the mean value of the signal,  $\bar{x}$ , we spoke as if we knew its value. In fact, we *do not know the value* of  $\bar{x}$ . If we have taken one image, we have only one *sample* of the signal, which we will call  $x_1$ . If we take another image, we can call that sample  $x_2$ , and a third,  $x_3$ . If we take  $n$  samples, the  $n^{\text{th}}$  sample we take is called  $x_n$ .

To find the mean value of the signal,  $\bar{x}$ , we simply add the samples together and divide by the number of samples:

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n}. \quad (\text{Equ. 2.1})$$

We can express the same operation of finding the mean value of a set of samples using summation notation:

## Chapter 2: Counting Photons

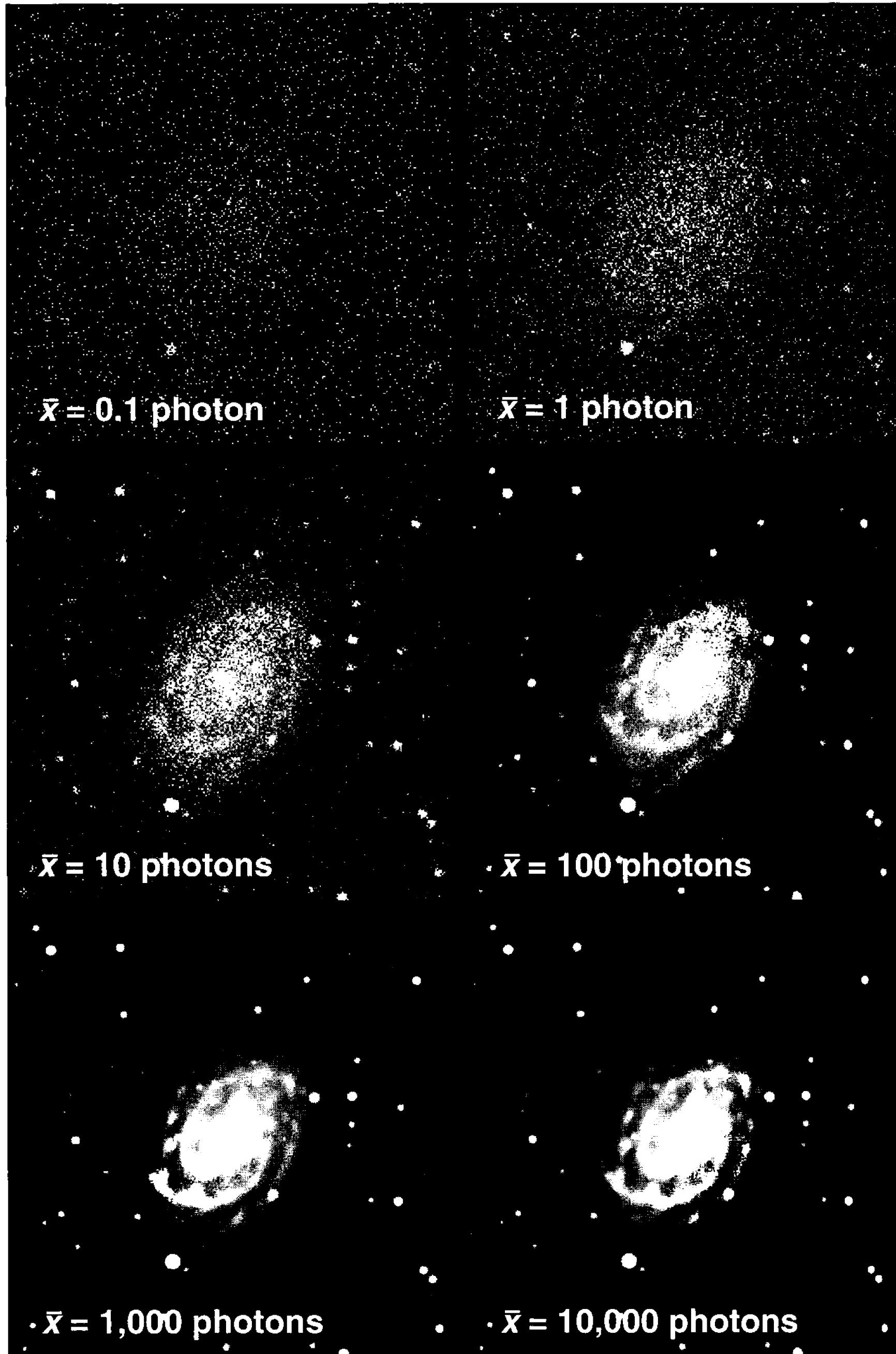


Figure 2.2 The greater the number of photons, the better the signal-to-noise ratio. In this case study, the photon count across the face of the galaxy image varies from a mean value of 0.1 photons to a mean value of 10,000 photons. The signal-to-noise ratio equals the square root of the photon count.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i. \quad (\text{Eq. 2.2})$$

Even after we have summed  $n$  samples, however, we still don't know the exact[ital] value of  $\bar{x}$  because the mean photon count remains uncertain by  $\sqrt{\bar{x}}$ .

Suppose that instead of examining the same pixel in a sequence of images, we were to examine one image in an area of sky that has no stars. Although you might expect that all pixels making up that area of sky would have the same value, they do not. Just as a sequence of samples of one pixel varies, each pixel in a set of pixels that are side-by-side is an independent sample of sky brightness, and it obeys the same rule that a series of samples at the same location does.

This has a profound impact on the collection of astronomical data. If you take two “identical” images of the same object and compare them, you will find that they are *not* identical. They differ because they are independent samples of the mean photon rate, and each and every pixel in the two images has to follow the  $\bar{x} \pm \sqrt{\bar{x}}$  photon-counting rule.

It may not be immediately obvious from the mathematical symbols that *the more photons in the signal, the better the signal quality*. Why should this be so? Consider the following example: we have two signals, one consisting of 25 photons and the other consisting of 100 photons. In the 25-photon signal, the expected variation is  $\sqrt{25}$ , or 5 photons; while in the 100-photon signal, the expected variation is  $\sqrt{100}$  or 10 photons. Since the variation in the 100-photon signal is twice that of the 25-photon signal, does that mean the 100-photon signal is worse? In one sense it is—it has twice the standard deviation. However, the percentage variation in the 25-photon signal is  $5/25$ , or 20%; while in the 100-photon signal, the percentage variation is  $10/100$ , or 10%. Even though it has twice the standard deviation, as a ratio of the signal strength, the 100-photon signal has only half the standard deviation as the 25-photon signal.

To quantify signal quality, communications engineers invented the *signal-to-noise ratio*, or *SNR*. This ratio is nothing more than the signal divided by the noise. For signals in which photon counting is the primary source of noise, the signal-to-noise ratio is:

$$\text{SNR} = \frac{\bar{x}}{\sqrt{\bar{x}}} = \sqrt{\bar{x}}. \quad (\text{Eq. 2.3})$$

The SNR of the 25-photon signal above is 5, and the SNR of the 100-photon signal is 10. The greater the signal-to-noise ratio, the better the image quality. Note, however, that within a given image, the signal is not the same at all places. The sky background will have a lower signal level than the bright center of a galaxy, so it is meaningless to assign an SNR to an entire image because signal-to-noise ratio is meaningful for only one signal level. Nevertheless, astronomers sometimes quote an SNR for an image, and when they do, it refers to the SNR at the signal level of the sky background.

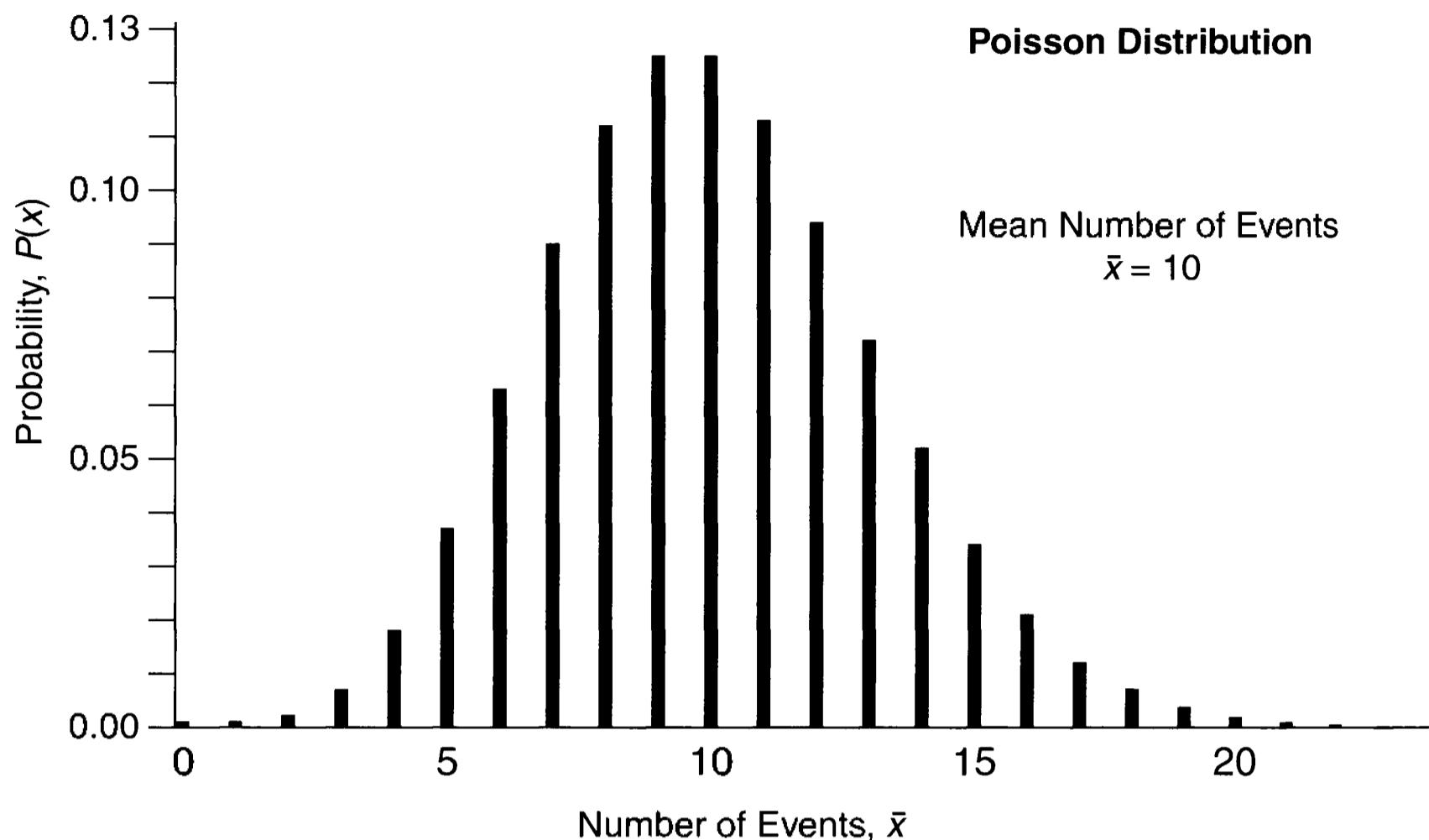


Figure 2.3 The Poisson distribution describes the probability, given some mean number of random events, that  $x$  events will occur. Given a mean rate of 10 events, the probability that 10 events will occur in a particular sample is 0.125, or 12.5%—but on rare occasions zero events can occur.

To place these rather abstract numbers in context, an image with a sky background signal-to-noise ratio of 3 looks rough and grainy. Increasing the SNR to 10 brings a huge improvement—the image still looks coarse, but it is not hopelessly bad looking. By the time the SNR has reached 30, the image appears solid, smooth, and “real.” When you have gathered enough photons to reach an SNR of 100, details are clear and crisp, and nebular features appear smooth and milky.

The dependence of image quality on photon count is graphically shown in Figure 2.2, comparing synthetic images of a galaxy that have mean photon counts ranging from 0.1 to 10,000.

### 2.3.1 The Poisson and Gaussian Distributions

Although the number of photons in a sample varies, when you examine a large number of samples, a clear pattern emerges. For example, if the mean number of photons per sample is 10, the same percentage of samples will have 6 photons, the same 7, and so on. For any mean number of photons per sample, there is a well-defined probability that  $x$  photon detections will occur. For independent random events such as photon detection, the distribution of probabilities is called the Poisson distribution after the French mathematician Siméon-Denis Poisson (1781–1840). The following equation describes the Poisson distribution:

$$P(x) = \frac{e^{-\bar{x}} \bar{x}^x}{x!} \quad (\text{Equ. 2.4})$$

where  $P(x)$  is the probability that  $x$  events will occur,  $\bar{x}$  is the mean number of events,  $e$  is the base of natural logarithms (2.718...), and  $x!$  denotes  $x$ -factorial.

The  $x!$  term computes the number of arrangements of photons that are possible, and  $\bar{x}^x$  describes how many different combinations of photons will yield  $x$  events. The  $e^{-\bar{x}}$  term adjusts probabilities so that for zero to infinity events, the sum of the probabilities is 100%.

The Poisson distribution tells us, for example, that when the mean number of photons,  $\bar{x}$ , is 1 per sample, there's a 36.8% chance that zero events will occur, a 36.8% chance that one event will occur, an 18.4% chance that two events will occur, a 6.1% chance that three events will occur, a 1.5% chance that four events will occur, and an 0.3% chance that five events will occur in each sample. Poisson statistics is the mathematical scaffolding behind the irregular “click-a-click, click-kitty, clack” pattern of the Geiger counter and the random sound of raindrops on the tin roof. Blind chance means that sometimes no events occur, sometimes several events occur, and sometimes (but rarely) many events occur.

In Poisson statistics, the number of events is always a positive integer. Positive because, by the nature of events, fewer than zero cannot happen. Integers because there is no such thing as half an event—an event (such as capturing a photon) either happens or does not happen. Everything we've said about  $\bar{x}$ ,  $\sqrt{\bar{x}}$ , and signal-to-noise ratio in the sections above applies to the Poisson distribution.

When the mean signal grows larger than about 20 events, however, the Poisson distribution becomes nearly indistinguishable from a Gaussian distribution (although technically, it remains a Poisson distribution). The *Gaussian distribution*, or *normal distribution*, is the standard bell-shaped curve that everyone who studies statistics knows and cherishes. The Gaussian distribution describes just about every process in which random deviations from a mean value occur.

The following describes the probability that  $x$  photon detections will occur in a sample interval:

$$P(x) = \frac{1}{\sqrt{2\pi\bar{x}}} e^{-\frac{(x-\bar{x})^2}{2\bar{x}}} \text{ for integer } x, \text{ and } \bar{x} > 20 \quad (\text{Equ. 2.5})$$

where  $P(x)$  is the probability that  $x$  events will occur,  $\bar{x}$  is the mean value, and  $e$  is the base of natural logarithms (2.718...). In this case, the  $1/\sqrt{2\pi\bar{x}}$  term insures that the sum of probabilities from zero to infinity events will equal 1.0.

What we've said in the earlier sections about  $\bar{x}$ ,  $\sqrt{\bar{x}}$ , and signal-to-noise ratio also applies to this Gaussian approximation of the Poisson distribution. In this respect, the Poisson and Gaussian distributions for counting photons are wonderfully simple and well-behaved.

### 2.3.2 Collecting More Photons by Image Summing

Because more photons produce higher image quality, astronomers try to collect as many as possible. One way to increase the total number of photons is to increase the exposure time, which is effective so long as no important parts of the image

## Chapter 2: Counting Photons

reach the sensor's full-well capacity (the largest signal it can produce). Another method is to add a series of images together, which increases the number of photons in the sample, thereby increasing the signal-to-noise ratio.

Assuming that the images are in register so that the same features appear at the same location from one image to the next, then given a mean count of  $\bar{x}$  photons at a given location, adding  $N$  images will put a total flux of  $N\bar{x}$  photons in that location. The signal-to-noise ratio for the summed image,  $\text{SNR}_N$ , will be:

$$\text{SNR}_N = \frac{N\bar{x}}{\sqrt{N\bar{x}}} = \sqrt{N\bar{x}}. \quad (\text{Equ. 2.6})$$

Comparing the SNR for a single image to the SNR for  $N$  images reveals that the summing  $N$  image improves the SNR by a factor of  $\sqrt{N}$ . If you want a signal-to-noise ratio that is four times better than you are getting from a single image, you need to sum 16 images.

As a practical matter, summing can be inconvenient when you are trying to compare summed data made with different numbers of images. Instead of summing the images, you can average them by dividing the sum by the  $N$ . In averaging, both the signal and the noise are reduced by the same factor. The signal-to-noise ratio for the averaged image will be:

$$\text{SNR}_N = \frac{N\bar{x}/N}{\sqrt{N\bar{x}/N}} = \sqrt{N\bar{x}}. \quad (\text{Equ. 2.7})$$

Providing you do not integerize the quotient, the signal-to-noise ratio is exactly the same whether you have summed or averaged multiple images.

The images that you see on amateur websites and for that matter, from the Hubble Space Telescope, are made by summing many images. Hubble's Ultra-Deep Field image of galaxies in early Universe was made by summing hundreds of images taken during a two-week time frame. A side benefit of summing images is that if some of the them are marred by airplane trails, bad guiding, or blurring, you just throw away the bad images and sum good ones.

### 2.3.3 Measuring Noise

Up to this point we have assumed that you have magically known how many photons fell on the detector. Knowing this enabled you to compute the noise and the signal-to-noise ratio. But in fact, you seldom know directly how many photons created the signal—all you have is a series of samples expressed in arbitrary units, *ADUs*, or *analog-to-digital units*. The numerical pixel values in the images from your digital camera or CCD are in units of ADUs.

Despite not knowing the true number of photons, you can still determine the signal-to-noise ratio at a specified signal level such as the sky background. Here's how you can go about doing just that.

You already know how to calculate the mean value of the sample in a set of signal values—as in Equations 2.1 and 2.2, you sum the signal values and divide

the sum by the number of samples. That gives you the signal. The standard deviation measures the departure of the samples from the mean value; so suppose that after computing the mean, you subtract the mean value from each sample value. Sometimes an individual sample will be larger than the mean so the deviation from the mean will be positive, and sometimes the sample will be smaller than the mean so the deviation will be a negative number. If you average all the deviations together, the positives and negatives will largely cancel. To make all deviations positive, you square the deviations, take the mean value, and then to counteract having squared the deviations, you take the square root of the mean of the deviations. This gives the “root mean square” deviation, or *standard deviation* from the mean of a series of measurements.

In mathematical notation, the root-mean-square operation looks like this:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{(n-1)}}. \quad (\text{Equ. 2.8})$$

In this equation, we've used the Greek letter  $\sigma$  (sigma) to denote the standard deviation,  $\bar{x}$  for the mean value, and  $n$  for the number of samples we have of  $x$ . Many pocket calculators and adding machines have a built-in standard deviation function—you enter a set of values for  $x$ , press the statistics key, and the machine computes  $\bar{x}$  and  $\sigma$ .

You can summarize what you know about the signal with the statement that the signal equals  $\bar{x} \pm \sigma$ . The value of  $\bar{x}$  expresses the mean, and the value of  $\sigma$  captures the degree of uncertainty in the mean value. If the deviations in the value of  $x$  have a normal distribution, then:

- 68.3% of samples will fall in the range  $\bar{x} \pm \sigma$ ,
- 95.4% of samples will fall in the range  $\bar{x} \pm 2\sigma$ , and
- 99.7% of samples will fall in the range  $\bar{x} \pm 3\sigma$ .

For these statistical properties to apply, the signal must follow a Gaussian distribution. While not the case with every type of random event, when it comes to counting reasonably large numbers of photons, these statistical rules do apply.

### 2.3.4 Signals Become More Complicated

Although it is tempting to expect the signal-to-noise ratio to be  $\bar{x}/\sigma$ , that simply is not true. The reason is that although you have computed the mean value of  $x$  in ADUs, you don't actually know whether zero ADUs means zero photons. In many types of measurement, and in all CCD cameras, the zero point in ADUs is offset from the true zero signal level. This zero offset is called the *signal bias*, or just plain *bias*. In other words, even when zero photons fall on the detector, the camera reports a bias signal, usually a small positive value such as 100.

Measurement bias is nothing new—in fact, you are used to dealing with it in

## Chapter 2: Counting Photons

your daily life. If you want one pound of cashew nuts at the grocery store, you can weigh the nuts. However, before putting any nuts on the scale, you check its reading. If the empty scale reads half a pound, you know that you must add nuts until the scale reads one-and-a-half pounds. The zero-point offset of the scale is exactly the same as the bias value in a digital camera.

To find the signal-to-noise ratio, you must subtract the bias value, which we will call  $b$ , from the mean signal, thus:

$$\text{SNR} = \frac{\bar{x} - b}{\sigma}. \quad (\text{Equ. 2.9})$$

Although the signal bias can be subtracted, it introduces another complication: you need to measure the signal bias before you can subtract it. In the grocery store, you know that if you tap the scale lightly, the needle will move and settle at a slightly different bias value. To determine the bias accurately, you might decide to tap the scale and read the bias value half-a-dozen times; and then you would need to calculate the mean of the bias readings.

If you were fanatical about buying that pound of cashews, you would also compute the standard deviation in the bias readings, thereby recognizing and acknowledging the fact that your measurement of the bias also has an associated uncertainty that contributes to the uncertainty in the measured weight of the nuts.

The same thing happens with images from digital cameras and in CCD images: for accurate measurements, you need to determine the zero point by taking bias frames and computing their mean value. You can then subtract the bias to obtain a measurement of the true signal level. With most digital cameras and CCDs, the bias value is small and steady, and adds only a small dose of uncertainty to the signal in your images.

Equations 2.4 and 2.5 assume that the signal consists entirely of photon counts. If you have studied statistics, you are already familiar with the *Gaussian distribution*. The Gaussian distribution gets its name from the legendary German mathematician (and astronomer) Carl Friedrich Gauss (1777-1855). The general equation for the Gaussian distribution is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\bar{x}}{\sigma}\right)^2} \quad (\text{Equ. 2.10})$$

where  $f(x)$  is the function describing the probability that  $x$  events will occur,  $\bar{x}$  is the mean number of events,  $e$  is the base of natural logarithms (2.718...), and  $\sigma$  is the standard deviation of  $\bar{x}$ . Whereas Equation 2.5 is specific to event probabilities such as photon counting, the Gaussian function makes no assumptions about the signal—given measured values for  $\bar{x}$  and  $\sigma$ , it will generate the expected distribution of signal values.

### 2.3.5 Unwanted Signals and More Sources of Noise

In addition to detecting photons, CCDs and CMOS devices generate an unwanted

## Section 2.4: Signals and Noise in Images

signal called *dark current*. Dark current is a true signal in that it conveys useful information about the detector and—unlike noise—dark current is not random. When we make images, the *raw image* combines the desired photon signal with the unwanted dark current signal as well as the signal bias.

Since most astronomers are interested only in the photon signal, we resent the need to make *dark frames* (images in which no light is allowed to reach the detector). Dark frames are, however, needed to measure as accurately as possible the dark current from every pixel, but only so that we can subtract the dark current from our raw images.

Dark current arises from lattice defects that generate extra electrons in the crystal lattice of the silicon detector. Each electron liberated is a statistically independent event, so dark current obeys the same Poisson statistical model that detected photons do. This means that dark current has noise. If we sample a dark current containing a mean of  $\bar{x}_d$  electrons, the electron count will have a statistical uncertainty of  $\sigma_d = \sqrt{\bar{x}_d}$  electrons. (To highlight the similarity of the statistics, we'll use the variables  $x$  and  $\sigma$  to denote both detected photons and dark-current electrons, but we'll distinguish those associated with the dark current by the "d" subscript.)

In the preceding sections, we have mentioned signal bias several times. Bias is another unwanted signal, but it's not due to a random process. Bias is a constant output offset,  $b$  ADUs, originating in the electronics that detect and amplify the signal from the detector. However, the bias is accompanied by *readout noise*, a random variation in the amplifier circuits that is imposed on everything output from the detector. Unlike the noise associated with counting photons or counting dark current electrons, readout noise adds a constant uncertainty regardless of the amplitude of the bias and signal levels. We'll denote readout noise as  $\sigma_{ron}$ . Readout noise is usually expressed in units of root-mean-square electrons.

## 2.4 Signals and Noise in Images

You now have enough background to examine the signals and noise that you will encounter in real images from your CCD or digital camera. Before beginning that examination, we need to get certain units of measurement squared away.

We introduced signals and noise in terms of detected photons, electrons, and ADUs; that is, in different units. To understand digital images, it is necessary to know how these different units relate to one another.

- When photons strike a detector, not all of them generate a signal. In a typical CCD, roughly 40% to 80% of the photons are detected—meaning that the photon has liberated an electron. Therefore, the detected photon count,  $x$ , translates directly into electrons.
- The statistical uncertainty in the photon count,  $\sigma$ , often called *shot noise*, has units of root-mean-square electrons.

## Chapter 2: Counting Photons

- Dark current electrons,  $x_d$ , are already in units of electrons.
- The dark current noise,  $\sigma_d$ , has units of root-mean-square electrons.
- Bias,  $b$ , is almost always given in ADUs.
- Readout noise,  $\sigma_{ron}$ , is almost always given in units of root-mean-square electrons to make it easier to compare readout noise to other sources of noise.
- In CCD and CMOS devices, electrons are converted to a voltage that is digitized and sent to your computer as an integer value. For lack of a more mellifluous term, the integer value is called an *analog-to-digital* unit (ADU) or *digital number* (DN).
- The *gain* or *conversion factor* relates electrons to ADUs, and is denoted by the symbol  $g$ . The units of the gain are *electrons per ADU*. In most CCD cameras, the gain is reasonably close to 1, but the gain may be as large as  $500 \text{ e}^-/\text{ADU}$  in webcams, or as small of  $0.1 \text{ e}^-/\text{ADU}$  in some CCDs.

As a general rule, astronomers use fundamental physical units (electrons) when they talk about properties of the detector and amplifier; and by necessity, they use the arbitrary derived units (ADUs) that the camera generates when they are discussing the images made by cameras.

Let's look at a few examples to see how these conversions work. Suppose 100 photons fall on a pixel of the detector in your CCD, digital camera, webcam, or whatever detector you happen to be using (and remember, the math is the same for all such devices). Typically 60 of those photons will generate an electron, and therefore be detected as a signal  $x$  in units of electrons. A device such as a charge detection node (in a CCD) senses the electrons and generates a voltage proportional to the number of electrons, and passes that voltage to an analog-to-digital converter. Suppose that the amplifier and A/D converter together have a gain of 2.5. In our example, the A/D converter adds a bias of 100.

The output signal from the camera will be:

$$S = \frac{x}{g} + b = \frac{60 \text{ electrons}}{2.5 \text{ electrons/ADU}} + 100 \text{ ADU} = 124 \text{ ADU}. \quad (\text{Eqn. 2.11})$$

To convert the signal electrons to ADUs, we've divided the signal in electrons by the gain in electrons per ADU, which gives us ADUs. The bias is already in units of ADUs, so we can add the terms.

In the example, we neglected dark current and did not even consider noise, but you can see the need to pay close attention to the units of measurement.

Let's now look at some realistic examples.

### 2.4.1 Signal and Noise in a Raw Image

You decide to make a raw image. What signals and what sources of noise do you expect to find? The signals are:

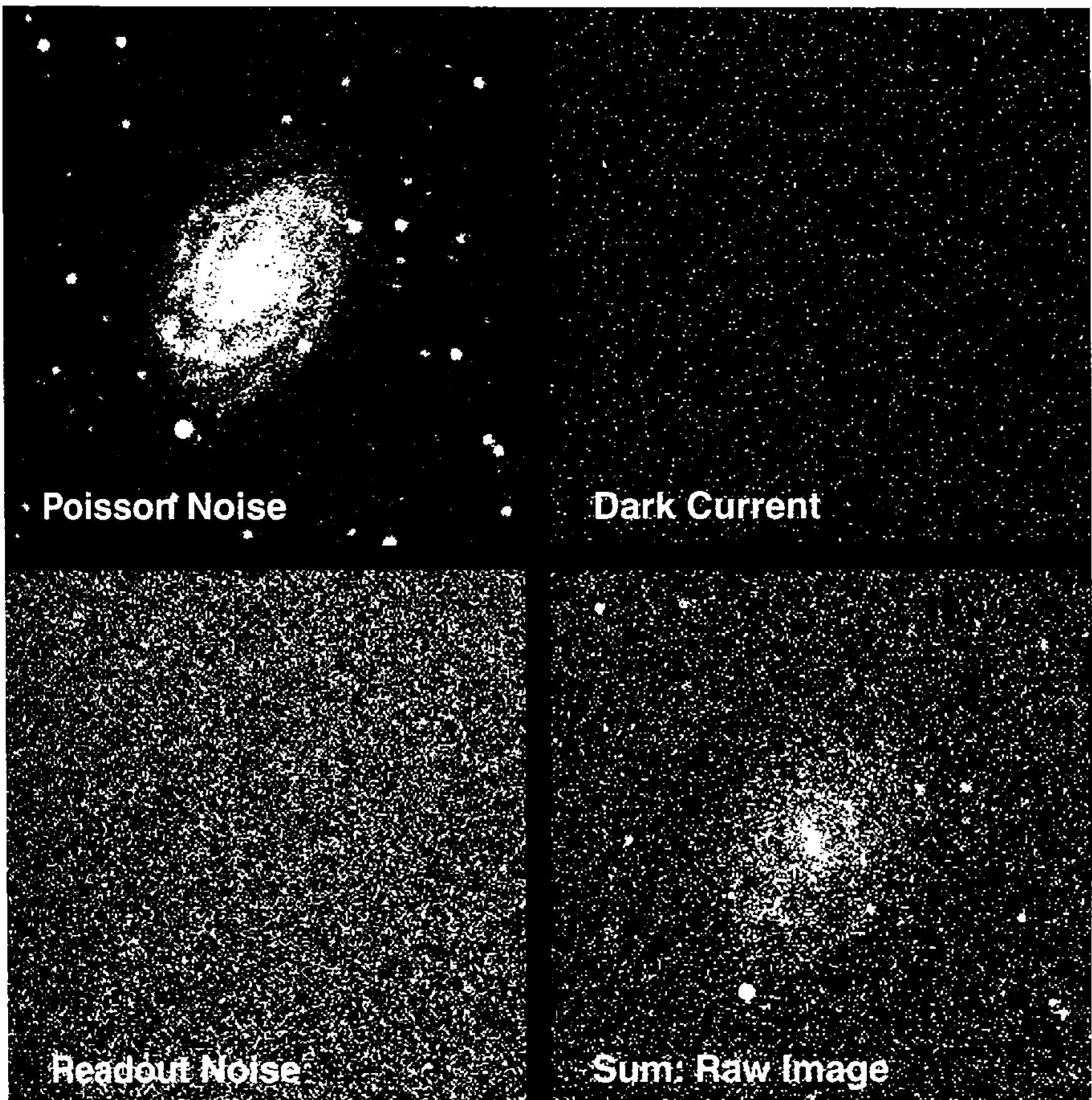


Figure 2.4 The raw image is the sum of a photon signal (with Poisson noise), an unwanted dark-current signal (with Poisson noise), and a bias constant (with readout noise). Although the dark current and the bias can be subtracted, the random noise that they introduce remains.

- The detected photon count,  $x$ , in electrons;
- the dark current,  $x_d$ , in electrons; and
- the bias,  $b$ , in ADUs.

The total signal in the raw image,  $S_{\text{raw}}$ , converted into units of ADUs, is:

$$S_{\text{raw}} = \frac{x}{g} + \frac{x_d}{g} + b . \quad (\text{Equ. 2.12})$$

The noise sources are:

- The photon shot noise,  $\sigma$ , in r.m.s. electrons;
- the dark current noise,  $\sigma_d$ , in r.m.s. electrons; and
- the readout noise,  $\sigma_{\text{ron}}$ , in r.m.s. electrons.

## Chapter 2: Counting Photons

The total noise in the raw image,  $\sigma_{\text{raw}}$ , converted into units of ADUs, is:

$$\sigma_{\text{raw}} = \frac{1}{g} \sqrt{\sigma^2 + \sigma_d^2 + \sigma_{\text{ron}}^2}. \quad (\text{Equ. 2.13})$$

At first glance, it looks like we've just done something sneaky. Whereas we simply summed the signals; when we added the noise sources, we squared each noise, summed them, and took the square root of the sum. However, recall that the noise originates as the random variation in an event count. By squaring the noise, we recover the number of events. To get the total number of events, we add them, and then take the square root to find the expected random variation in the sum of events. Another way of looking at this is that because noise is random, the variations in a collection of noise sources will partially cancel each other.

It is important to recognize that because noise sources represent counts of events, then even when we subtract a signal, we must add its noise! Why? The number of events under consideration has increased, and that's where the noise comes from, not whether we've added the events or subtracted them. In the next section, you'll see why this matters.

As a reality check, let's compute the signal and noise in a raw image using realistic values for a good-quality CCD camera. In a 60-second exposure under fairly good suburban-quality skies, you can expect roughly 1000 photons per pixel, and therefore roughly 600 electrons of signal.

The dark current in a cooled CCD camera will be around 1 electron per second per pixel; so in 60 seconds, the dark current signal will be about 60 electrons. We'll assume a gain of 2.5 electrons per ADU, a readout noise of 8 root-mean-square electrons, and a bias (typical of many astronomical CCD cameras) set at the convenient value of 100.

The output signal is:

$$S_{\text{raw}} = \frac{600}{2.5} + \frac{60}{2.5} + 100 = 364 \text{ [ADUs]}. \quad (\text{Equ. 2.14})$$

The noise in the output signal is:

$$\sigma_{\text{raw}} = \frac{1}{2.5} \sqrt{(\sqrt{600})^2 + (\sqrt{60})^2 + 8^2} = 10.76 \text{ [ADUs]}. \quad (\text{Equ. 2.15})$$

Although we could compute a signal-to-noise ratio for the raw image by dividing the signal we care about (the 240 ADUs of sky) by the total noise, it would not be a meaningful calculation. We've treated the dark current as uniform, but in fact it usually varies considerably from pixel to pixel. The dark current signal adds a highly distracting *noise pattern* to the image plus a scattering of *hot pixels* to it. (Technically speaking, we remind our readers that a noise pattern is not noise, but an unwanted signal.) To see the image properly, we will make a *dark frame* by making an exposure having the same duration as the raw image, but with the camera shutter closed. The dark frame will capture a record of the noise pattern and hot pixels, which we then subtract to produce a clean image.

### 2.4.2 Signal and Noise in a Dark Frame

Making a dark frame is exactly the same as making an exposure *except that the camera shutter is kept closed* so that no light reaches the detector. Most CCD cameras and digital cameras support making dark frames so that the user can remove the dark current pattern.

During a 60-second exposure to make a dark frame, the photon signal is zero photons per pixel. In 60 seconds, the dark current signal will be about 60 electrons, as it was for the raw image. The gain, readout noise, and bias are all the same as before.

The output signal is:

$$S_{\text{dark}} = \frac{0}{2.5} + \frac{60}{2.5} + 100 = 124 \text{ [ADUs].} \quad (\text{Equ. 2.16})$$

The noise in the output signal is:

$$\sigma_{\text{dark}} = \frac{1}{2.5} \sqrt{(\sqrt{0})^2 + (\sqrt{60})^2 + 8^2} = 4.45 \text{ [ADUs].} \quad (\text{Equ. 2.17})$$

The dark frame is a map of the dark current from the sensor. Note that the useful portion of the signal—the 24 ADUs of dark current—is uncertain with a standard deviation of 4.45 ADUs; so the dark frame is not a perfect map of the dark current, but a sample of the dark current with an accompanying error.

### 2.4.3 Signal and Noise in a Dark-Subtracted Image

We are finally ready to investigate signal and noise in a dark-subtracted image. Dark subtraction removes both the bias and the dark current pattern from the raw image, but adds noise from the dark frame to the noise already present in the raw frame. If this somehow seems a bit “unfair,” remember that to remove the dark current pattern, we’ve been forced to count more electrons, which, by the inexorable statistics of event counting, leads to greater uncertainty.

The photon signal in the dark-subtracted image is:

$$\begin{aligned} S_{\text{image}} &= S_{\text{raw}} - S_{\text{dark}} & [\text{ADUs}] & \quad (\text{Equ. 2.18}) \\ &= \left( \frac{x}{g} + \frac{x_d}{g} + b \right) - \left( \frac{x_d}{g} + b \right) \\ &= 364 - 124 \\ &= 240. \end{aligned}$$

The noise in the dark-subtracted image is:

## Chapter 2: Counting Photons

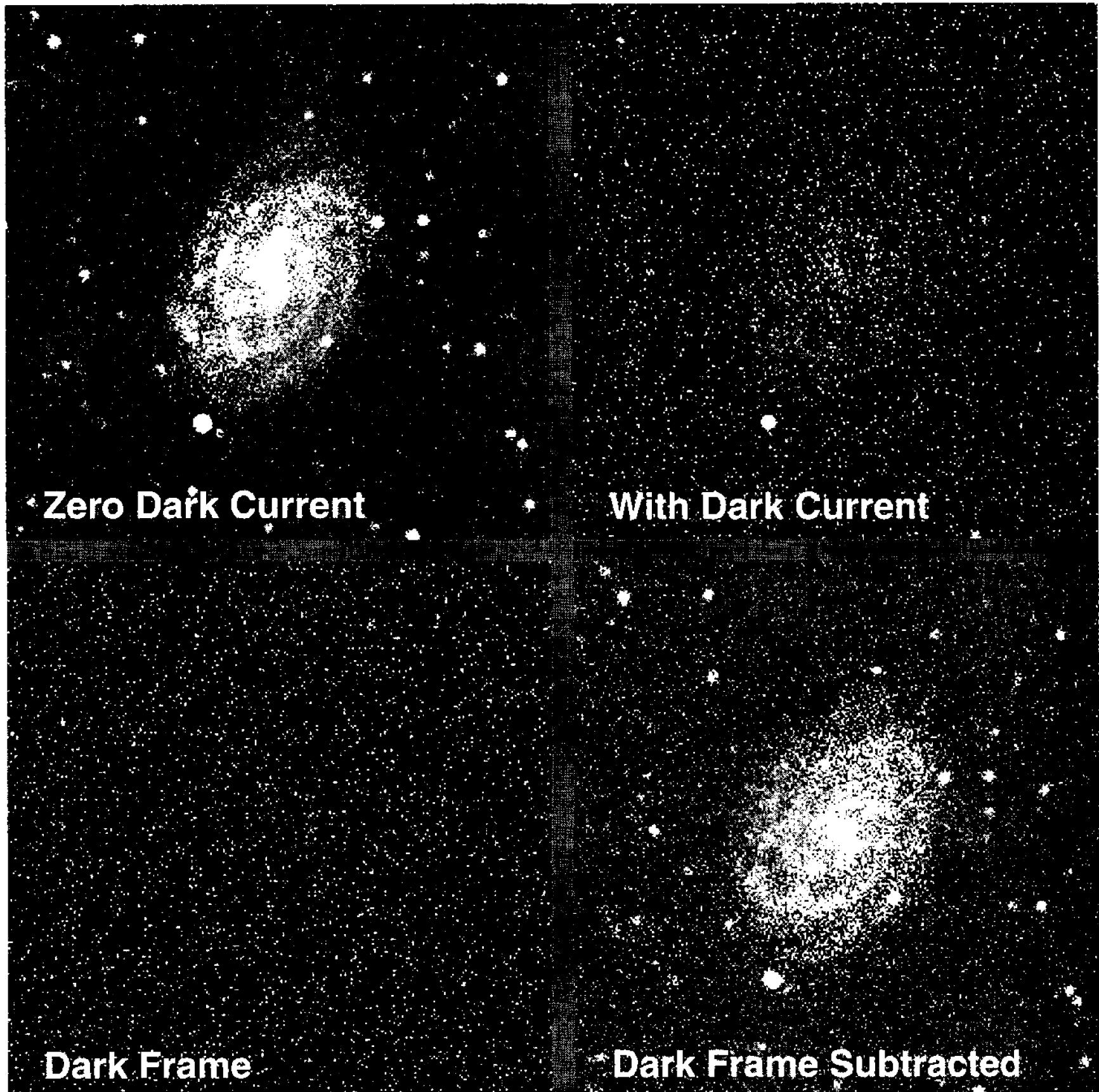


Figure 2.5 Although subtracting a dark frame adds noise, it also removes the distracting pattern of dark current and hot pixels. The “ideal” image (upper left) has zero dark current; the image (upper right) has a mean of 60 electrons dark current. Subtracting a dark frame adds noise but improves the result.

$$\begin{aligned}\sigma_{\text{image}} &= \sqrt{\sigma_{\text{raw}}^2 + \sigma_{\text{dark}}^2} \quad [\text{ADUs}] \\ &= \sqrt{115.78 + 19.80} \\ &= 11.64.\end{aligned}\tag{Equ. 2.19}$$

We can now finally determine the signal-to-noise ratio,  $\text{SNR}_{\text{image}}$ , at the sky background level of 240 ADUs to be:

$$\text{SNR}_{\text{image}} = \frac{S_{\text{image}}}{\sigma_{\text{image}}} = \frac{240}{11.64} = 20.6.\tag{Equ. 2.20}$$

Although an image with this signal-to-noise ratio would look somewhat grainy, faint deep-sky objects would be readily visible in it. Note that this is a *sky-limited image* because the overwhelmingly dominant source of noise comes from

## Section 2.4: Signals and Noise in Images

the photon statistics of the sky background, whose 600 electrons contribute 9.8 ADUs worth of noise. For comparison, the readout noise of 8 electrons root-mean-square adds only 3.2 ADUs of noise, and the 60 electrons of dark current add only another 3.1 ADUs of noise.

Under dark skies, however, dark current noise or readout noise can easily become the dominant noise source. At rural dark-sky sites, you might find a sky contribution of 40 electrons in a 60-second exposure. Photon statistics would then account for a mere 2.5 ADUs of noise, but the readout noise and dark current noise would still be 3.2 and 3.1 ADUs, respectively. Under these conditions, your images would be *detector limited* or *photon limited*.

How good is this image? Against a noise level of 11 ADUs, a single pixel 11 ADUs brighter than the mean level of the sky background—240 ADUs—would not stand out enough to be visible; but a few dozen pixels or a few hundred pixels representing a faint galaxy image would appear as clearly brighter than the surrounding sky. A large, bright nebula would be clear and unmistakable.

In this example, subtracting the dark frame raised the noise level from 10.76 to 11.64 ADUs. As a practical matter, the dark-subtracted image with 11.4 ADUs of random noise would certainly look a lot nicer than a raw image with 24 ADUs worth of dark current patterning and 10.74 ADUs of random noise.

However, it is not necessary to accept even this modest loss. By making several dark frames and averaging them together, you can reduce the  $\sigma_{\text{dark}}$  to a negligible level. Applying Equation 2.7 to averaging dark frames, you can see that by averaging ten dark frames, you could reduce  $\sigma_{\text{dark}}$  to 1.4 ADUs. Subtracting the averaged dark frame from the raw image yields a  $\sigma_{\text{image}}$  of 10.66 ADUs, demonstrating that if it's done properly, dark-frame subtraction causes negligible loss in image quality and removes the hot-pixel noise pattern. If you do your dark frames right, you'll preserve every photon your camera captures!

### 2.4.4 The Effect of the Sky Background on Signal and Noise

In the preceding section, we have considered various noise sources and their impact on the signal-to-noise ratio. We hinted that the sky background has a significant impact on a camera's ability to detect extended astronomical objects such as galaxies and nebulae. In this section, we examine the role of the sky background on imaging performance.

In Section 2.4.1 we defined the count of photons detected,  $x$ . In astronomical observation, however, photons come from two sources: an object of interest, and parasitic illumination from the night sky:

$$x = x_{\text{object}} + x_{\text{sky}} \text{ [electrons].} \quad (\text{Equ. 2.21})$$

We now define a ratio between object photons and sky photons as:

$$\text{OSR} = \frac{x_{\text{object}}}{x_{\text{sky}}}, \quad (\text{Equ. 2.22})$$

## Chapter 2: Counting Photons

in which OSR stands for *Object-to-Sky Ratio*. Sky brightness varies greatly from rural to suburban to urban sites, and also varies with the phase of the Moon. Furthermore, the range of object brightness is enormous. Suffice it to say that for very bright objects (the Moon and planets), the OSR is 1000 or more, and for exceedingly faint objects, it may be as low as 0.001.

Because the OSR varies over such a wide range, you may wish to determine the OSR of objects in your own images specific to your skies and local conditions. From a calibrated image, measure the following:

- $S_{\text{object}}$ , the pixel value of the object in ADUs; and
- $S_{\text{sky}}$ , the pixel value of the sky, in ADUs.

Remember that when you measure the pixel value of the object, it is the sum of the object and the sky. Because of this, the OSR must be calculated as follows:

$$\text{OSR} = \frac{S_{\text{object}} - S_{\text{sky}}}{S_{\text{sky}}} . \quad (\text{Equ. 2.23})$$

To illustrate the impact of sky brightness, we'll now consider the signal-to-noise ratio for three cases: a dark rural sky, a fairly good suburban sky, and a bright urban sky. Our celestial object is a galaxy that yields a photon flux of 250 electrons per pixel in 60 seconds. For this exercise, we'll use a camera with the gain, dark current, and readout noise from the preceding sections.

**Dark Rural Sky.** In this setting, we assume a signal of 40 electrons per pixel in 60 seconds, so for our celestial object, the OSR is 6.3; that is, the object is considerably brighter than the sky background. In a dark-subtracted image, a sky pixel measures  $16 \pm 5.4$  ADUs and an object pixel measures  $116 \pm 8.3$  ADUs. The object is 100 ADUs brighter than the surrounding sky, and it will stand out clearly against the noise in the dark rural sky background.

**Fairly Good Suburban Sky.** For our suburban sky, we take a signal of 600 electrons per pixel in 60 seconds. Our celestial object is now less bright than the sky background; its OSR for this sky is 0.42. In a dark-subtracted image, a sky pixel measures  $240 \pm 10.9$  ADUs and an object pixel measures  $340 \pm 12.6$  ADUs. The object is still 100 ADUs brighter than the surrounding sky, but both the sky and the object noise are greater than they were under the rural sky.

**Bright Urban Sky.** Urban skies are detrimental to good imaging because they are so bright. For the urban sky, we assume a signal of 4000 electrons per pixel in 60 seconds. The OSR of the object has fallen to 0.06. In a dark-subtracted image, a sky pixel measures  $1600 \pm 25.7$  ADUs and an object pixel measures  $1700 \pm 26.5$  ADUs. Our object is still 100 ADUs brighter than the surrounding sky, but the noise levels have become a significant fraction of the object's pixel value, and the image looks quite noisy. Nevertheless, even though the urban sky is 100 times brighter than the rural sky, the celestial object remains visible.

In the section that follows, we investigate the improvement obtained when you average multiple images. One seemingly paradoxical result is that it is neces-

## Section 2.4: Signals and Noise in Images

sary to accumulate more total exposure time under a bright sky than that required to achieve the same result under a dark sky.

### 2.4.5 Signal and Noise in Multiple Averaged Images

In preceding sections, you observed that by averaging several dark frames you could avoid adding noise. In this section, you will see that by shooting an ample number of dark frames and by averaging multiple raw images, you can cut noise and build image quality.

Begin by reviewing the signal and noise information you have on hand:

- You know  $S_{\text{raw}}$ , the raw-image signal in ADUs,
- you know  $\sigma_{\text{raw}}$ , the raw-image sky noise in ADUs,
- you know  $S_{\text{dark}}$ , the dark-frame signal in ADUs, and
- you know  $\sigma_{\text{dark}}$ , the dark-frame noise in ADUs.

Suppose that you decide to shoot a number,  $N_{\text{raw}}$ , of raw images and some other number,  $N_{\text{dark}}$ , of dark frames. You average the raw images, you average the dark frames, and then you subtract them.

Here is the general equation for the signal:

$$S_{\text{combined}} = \frac{N_{\text{raw}} S_{\text{raw}}}{N_{\text{raw}}} - \frac{N_{\text{dark}} S_{\text{dark}}}{N_{\text{dark}}} = S_{\text{raw}} - S_{\text{dark}}. \quad (\text{Equ. 2.24})$$

When you average your raw images and dark frames, the resulting *signal level does not depend on the number of raw images and dark frames*.

Here is the general equation for the noise:

$$\sigma_{\text{combined}} = \sqrt{\frac{\sigma_{\text{raw}}^2}{N_{\text{raw}}} + \frac{\sigma_{\text{dark}}^2}{N_{\text{dark}}}}. \quad (\text{Equ. 2.25})$$

The fact that the  $N_{\text{raw}}$  and  $N_{\text{dark}}$  are inside the square-root operation means that more raw images and more dark frames improve the image quality only as the square root—so getting a factor of two improvement requires four times more images. This difficulty notwithstanding, you’re happy to shoot lots of images if it means that you’ll have outstanding results.

Here is the signal and noise information for the suburban-sky example:

- $S_{\text{raw}}$  is 364.0 ADUs,
- $\sigma_{\text{raw}}$  is 10.76 ADUs,
- $S_{\text{dark}}$  is 124.0 ADUs, and
- $\sigma_{\text{dark}}$  is 4.45 ADUs.

The signal will always be:

$$S_{\text{combined}} = S_{\text{raw}} - S_{\text{dark}} = 240 \text{ [ADUs]}. \quad (\text{Equ. 2.26})$$

The noise depends on the number of raw images and dark frames:

## Chapter 2: Counting Photons

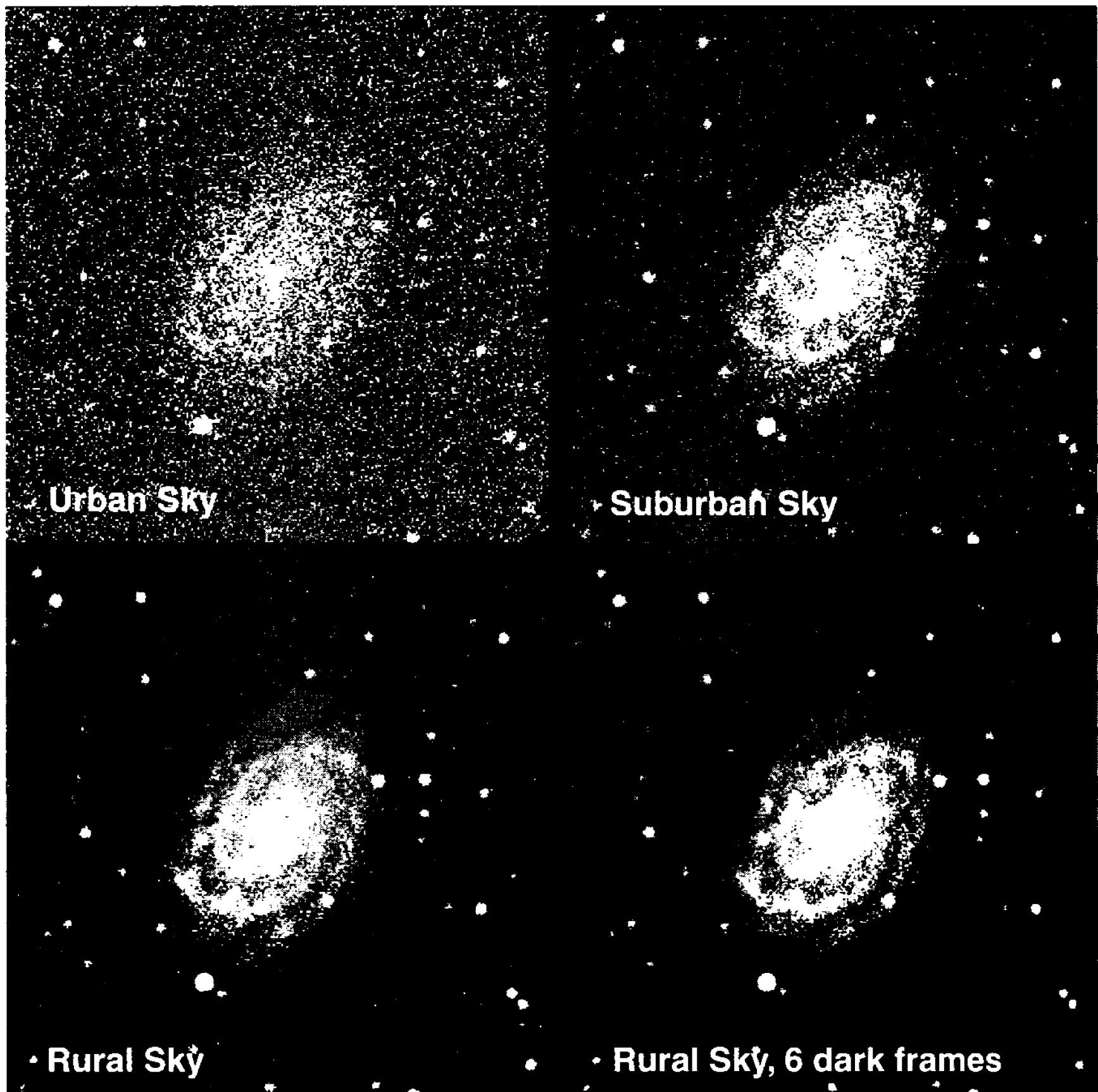


Figure 2.6 Under bright urban and suburban skies, the dominant source of image noise is photon statistics. Under dark rural skies, however, dark current and readout noise may dominate. Under dark skies, therefore, making multiple dark frames decreases noise and produces better images.

$$\sigma_{\text{combined}} = \sqrt{\frac{115.78}{N_{\text{raw}}} + \frac{19.80}{N_{\text{dark}}}} \text{ [ADUs].} \quad (\text{Equ. 2.27})$$

Suppose that you make 10 raw images and 10 dark frames—what signal-to-noise ratio can you expect? Evaluating the equation gives  $\sigma_{\text{combined}} = 3.68$  ADUs, producing sky-level signal-to-noise ratio of 65—suggesting that can expect a combined image of excellent quality!

But—are 10 raw images and 10 dark frames the best use of precious telescope time? Intuitively, the answer is “no” because the noise contribution from the dark frame is so much smaller than that of the raw images. If you’re willing to accept equal noise contributions from dark frames and raw images, then if you shoot some number of raw images,  $N_{\text{raw}}$ , you should shoot  $N_{\text{dark}}$  dark frames:

## Section 2.5: Signal and Noise Effects from Flat-Fielding

$$N_{\text{dark}} = \left( \frac{\sigma_{\text{dark}}^2}{\sigma_{\text{raw}}^2} \right) N_{\text{raw}} . \quad (\text{Equ. 2.28})$$

In the suburban skies example, you only need to shoot one dark frame for every 6 raw images. With 17 raw images and 3 dark frames, Equation 2.27 gives a total noise of  $\sigma_{\text{combined}} = 3.66$  ADUs, for a signal-to-noise ratio of 65—another excellent-quality image.

For the dark-sky example, for which  $S_{\text{combined}} = 16$  ADUs and  $\sigma_{\text{raw}} = 2.5$  ADUs, your images are noise limited. You must make enough dark frames to subtract the dark current as accurately as possible; for dark skies, Equation 2.28 calls for 3.1 dark frames for each raw image. If you shoot 10 raw images and 10 dark frames, the result is  $\sigma_{\text{combined}} = 1.98$  ADUs—considerably less noise than you could attain under suburban skies. Paradoxically, the signal-to-noise ratio has fallen to 8.1 at the sky-brightness level, but that's only because the sky is so very dark. To make a direct comparison, compute the signal-to-noise ratio for a nebula with a pixel value of 240 ADUs, and you will see that the signal-to-noise ratio rises to an outstanding 121.

If you take Equation 2.25 seriously for the dark rural sky and decide to shoot 10 raw images and 32 dark frames, you get  $\sigma_{\text{combined}} = 1.11$  ADUs and a signal-to-noise ratio for the sky background of 14.4; but for a 240-ADU nebula, the signal-to-noise ratio is 216. Dark skies make a huge difference in image quality, especially if you shoot enough dark frames to make photon statistics the dominant source of noise.

## 2.5 Signal and Noise Effects from Flat-Fielding

Flat frames are maps of the sensitivity of the detector. The flat frame records dust donuts, optical vignetting, and nonuniformities in the detector. After combining raw images and dark frames during calibration, the image is divided by the flat frame. In this process, noise in the flat frame will be transferred to the final calibrated image.

To avoid adding significant noise from the flat-field frame, all you need to do is make flats with a very high signal-to-noise ratio. Doing that turns out to be quite easy.

To make a flat frame, your telescope points at a bright, uniformly illuminated surface. Some astronomers prefer the twilight sky; others prefer a screen mounted near the telescope or an internally illuminated light box. In all cases, the source is bright enough to produce a solidly exposed image in 2 to 15 seconds.

In a good flat frame, the signal should be about half the full-well capacity of the detector, or about 32,000 electrons for a typical astronomical CCD in a 10-second exposure. From Equations 2.12 and 2.13, we find the output signal to be:

## Chapter 2: Counting Photons

$$S_{\text{raw}} = \frac{32000}{2.5} + \frac{10}{2.5} + 100 = 12904 \text{ [ADUs].} \quad (\text{Equ. 2.29})$$

The noise in the raw flat-frame signal is:

$$\sigma_{\text{raw}} = \frac{1}{2.5} \sqrt{(\sqrt{32000})^2 + (\sqrt{10})^2 + 8^2} = 71.6 \text{ [ADUs].} \quad (\text{Equ. 2.30})$$

At the same time you make the flat-frame exposures, you make dark frames to accompany them.

The output signal is:

$$S_{\text{dark}} = \frac{0}{2.5} + \frac{10}{2.5} + 100 = 104 \text{ [ADUs].} \quad (\text{Equ. 2.31})$$

The noise in the output signal is:

$$\sigma_{\text{dark}} = \frac{1}{2.5} \sqrt{(\sqrt{0})^2 + (\sqrt{10})^2 + 8^2} = 3.44 \text{ [ADUs].} \quad (\text{Equ. 2.32})$$

To be sure of attaining a high signal-to-noise ratio, a conscientious observer might make 10 flat frames and 10 flat dark frames. Since you are making an averaged image, you can apply Equations 2.24 and 2.25:

$$S_{\text{flat}} = S_{\text{raw}} - S_{\text{dark}} = 12904 - 104 = 12800 \text{ [ADUs]} \quad (\text{Equ. 2.33})$$

$$\sigma_{\text{flat}} = \left( \sqrt{\frac{71.6}{10}} + \sqrt{\frac{3.44}{10}} \right) = 2.74 \text{ [ADUs].} \quad (\text{Equ. 2.34})$$

The resulting signal-to-noise ratio is over 4000. Since it takes just a few minutes to make the flat frames, there is no reason to settle for less.

Dividing a dark-subtracted image by a flat frame will add noise to the image. The following equation gives the resulting signal-to-noise ratio when an image is divided by a flat frame:

$$\text{SNR}_{\text{result}} = \frac{1}{\sqrt{\frac{1}{\text{SNR}_{\text{image}}^2} + \frac{1}{\text{SNR}_{\text{flat}}^2}}} . \quad (\text{Equ. 2.35})$$

Applying this equation to the signal-to-noise ratio at the nebula brightness level for the combined image from the dark-sky site in the previous section, we calculate the loss in signal-to-noise ratio caused by flat-fielding:

$$\text{SNR}_{\text{result}} = \frac{1}{\sqrt{\frac{1}{(216)^2} + \frac{1}{(4000)^2}}} = 210 . \quad (\text{Equ. 2.36})$$

This example demonstrates that the loss of image quality caused by the division in a well-made flat-frame is negligible.

## 2.6 A Little Sermon on Signals and Noise

This chapter presents a great deal of material that will at first seem abstract and irrelevant to the practical matter of making beautiful images. As your sophistication grows, however, you will likely find yourself striving for smoother images with less noise—and you'll want to know what you must do to reach that goal.

It is tempting to fall into the trap of thinking that there is some shortcut, some trick that will work wonders on your images. But there is no trick. Astronomical imaging is driven by fundamental considerations: How many photons did you capture? How bright is your sky background? What is the dark current in your camera? How many dark frames did you shoot? There is no magic. The key is to get the signal level up and beat the noise level down.

## Chapter 2: Counting Photons

---

# 3 Digital Image Formats

---

Digital images are numbers, representing pixels, arranged in an orderly way. Image file formats are the agreed-upon standards for organizing this image information. Only by agreeing on standard file formats and adhering rigorously to the format definition can astronomers (or for that matter, any group that uses digital images) expect to exploit the full potential of digital imaging.

A file format is an agreed-upon way to store image data. Since the early days of computing, thousands of file formats have been invented. Those that remain have survived not only because the format satisfied the needs of a group of users, but also because the group of users worked together to uphold the standard format.

In astronomy, the standard is a file format called FITS, the NASA-endorsed Flexible Image Transport System. Used by astronomers world-wide, FITS makes it possible for any astronomer to exchange images with any other astronomer anywhere. FITS is not the standard because it is technically superior, but because astronomers insure that all FITS files adhere to the defined standard.

This chapter explores the FITS format standard in detail, and takes just a brief look at several non-astronomical image file formats:

- TIFF is the Tagged Image File Format, used widely in the graphics, publishing, and printing industries.
- BMP is the native format developed by Microsoft for imaging in the Windows environment.
- AVI files hold Audio/Video Interleaved data; that is, sequences of images with optional audio data, used for webcam output.
- JPEG is a product of the Joint Photographic Experts Group, a compressed format used for posting images on the Internet.
- RAW, NEF, and CRW are sparsely documented proprietary formats used for raw data from digital camera sensors.

## 3.1 File Format Basics

All file formats share certain basic characteristics. First, there must be some way for software to identify the file as an image meeting the format standard; this may be as simple as a filename extension such as FTS, TIF, or BMP, plus the presence

## Chapter 3: Digital Image Formats

of standard characters at the beginning of the file. All FITS files begin with a set of mandatory keywords; all TIFF files begin with the letters “II” or “MM” depending on whether the file uses little-endian (Intel) or big-endian (Motorola) byte order; and all BMP files begin with the characters “BM.”

Almost all files begin with a header, some designated number of bytes that tell the program reading the file something about the stream of bytes that follows. In a multiuse format like FITS, the header has to tell the reading program whether the file contains an image or some other type of data.

In a format as flexible as TIFF, the header tells how many images are in the file and how they are organized. The reading program needs to know what type of image to expect (grayscale, palette color, RGB color, CMYK color, etc.), how many pixels wide and deep the image is, and how data are stored in the file.

The header may also contain auxiliary information; such as the subject, the date and time the image was made, its processing history, and other information almost without limit. For example, FITS file headers store information on the date and time the image was taken, what telescope and CCD camera was used, and the name of the astronomer. TIFF files often contain tiny “thumbnail” copies of the primary image, so that reading programs can preview what’s in the file without loading the entire file.

In a pared-down header typical of the BMP format, all of the information necessary to read and display the image is stored in a few dozen bytes. However, the BMP image header may also contain a color palette, which expands the header to several thousand bytes.

Following the header is the image itself, which, depending on its size, may run from a few thousand bytes to many megabytes. The FITS standard defines precisely how five different types of numerical data are to be stored in its image data array. TIFF defines variations based on whether the image is binary black-and-white, grayscale, palette color, or 24-bit color, with variations on the variations depending on what type of color is in the image and how the color information is stored—all precisely defined as part of a very complex standard. TIFF even allows data compression by several different methods. As a result of TIFF’s complexity, the only programs that can read every TIFF file possible are high-end graphic-arts software.

In contrast, the BMP defines just a few simple image types, so that virtually every graphics program can read and display BMP files. Microsoft supports BMP to the extent that loading and saving images using the BMP format is built into its popular Visual Basic programming language.

Finally, the file may end in a “tailer.” FITS requires the total length of a FITS file to be a multiple of 2,880 bytes, so the end of every FITS file is padded with ASCII space characters. TIFF allows programmers to store all kinds of information at the end of the file, and some programmers use the option and some don’t. BMP does not have a tailer.

The sections that follow examine the FITS format in careful detail, and outline the basic structure of the TIFF, BMP, and JPEG formats.

## 3.2 FITS: The Standard Format in Astronomy

FITS stands for Flexible Image Transport System. It evolved in the early 1980s when astronomers at Kitt Peak National Observatory and the National Radio Astronomy Observatory (NRAO) began to cope with digital images from the Very Large (radio telescope) Array and digital images from CCD cameras that were then just coming into use. There had to be some way—some standard way—to move data from one observatory to another. In response, Don Wells, then at Kitt Peak; Eric Greisen at the NRAO; and Ron Harten at the Netherlands Foundation for Radio Astronomy developed what has become the world standard for communicating astronomical data.

Using FITS, an astronomer in Australia can send you a file taken with a new CCD camera you've never even heard of—and you can count on loading that image into your favorite image processing software. Cruising the Internet, you can download a section of the Palomar Observatory Sky Survey as a FITS file. Then when the need arises, you can e-mail a copy of your latest supernova discovery image to a professional astronomer with full confidence that the file will load on the observatory's UNIX workstation.

TIFF, BMP, and JPEG are good ways to distribute images that have already been processed, but none of these were ever intended to store the enormous amount of information in a raw CCD image. These formats store data over a limited range of values, and JPEG trades-off image quality for small file size. However, FITS was designed to transport scientific information—data such as 16-bit raw image data from a CCD camera—and to retain every bit of it.

## 3.3 Overview of FITS

FITS is simply a standardized way to store image data. By design, FITS is friendly to both humans and computers, meaning that you can peek into FITS files with a text editor and figure out what's going on. However, the very flexibility of FITS—since it can also store data tables, multiple groups of data, multispectral images, and arrays with up to 999 axes—has almost become its undoing. No all-purpose software can cope with the range of valid non-image FITS files. When Wells, Greisen, and Harten invented FITS, they introduced it as a plain-vanilla format they called Basic FITS—and Basic FITS is perfect for amateur astronomy. Basic FITS can handle any image that amateur astronomers are likely to create.

FITS files consist of three parts: a header, the image data, and a tailer. The header contains information that enables a human or a computer to read and interpret the image stored in the file. The image is coded as binary data in precisely defined binary formats. The tailer adds extra bytes to pad the file to a standard length.

The FITS header, written in ASCII text characters, is an integer multiple of 2,880 bytes in length; that is, the header can be 2,880, 5,760, 8,640, or more bytes long. The header is an integer multiple of 36 lines containing 80 bytes each. Each

## Chapter 3: Digital Image Formats

```
00000000111111112222222233333334444444455555556666666777777778  
1234567890123456789012345678901234567890123456789012345678901234567890  
  
SIMPLE=-----T//this file conforms to the FITS standard-----  
BITPIX=-----16//it consists of 16-bit 2's complement integers---  
NAXIS=-----2//the data is two-dimensional-----  
NAXIS1=-----378//width of the image in pixels-----  
NAXIS2=-----242//depth of the image in pixels-----  
BZERO=-----0.0//pixel_value=BZERO+BSCALE*array_value-----  
BSCALE=-----1.0//pixel_value=BZERO+BSCALE*array_value-----  
OBJECT='M101'//subject of this image-----  
DATE_OBS='2003-05-07'//UT date of integration-----  
TIME_OBS='05:19:30.10'//UT time, start of integration-----  
EXPOSURE=-----300.0//integration time in seconds-----  
TELESCOP='6-inch-f/5-reflector'//telescope used to take this image-----  
OBSERVER='Richard Berry'//the name of the observer-----  
INSTRUME='CB245-CCD-camera'//the device used to capture the image-----  
COMMENT=exceptional night, very clear, no clouds anywhere in the sky-----  
END-----
```

Figure 3.1 This is a Basic FITS header. The number bar at the top is intended to help you follow the columns more clearly. Keywords occupy columns 1 through 8, value indicators columns 9 and 10, and values columns 11 and higher. Integer and floating-point numbers are right-justified to column 30.

line is called a “card image,” and contains one keyword followed by a specified type of data. (The term “card image” dates back to earlier days of computing, when data was stored on punched cards, each of which held 80 characters.) Depending on how much information is stored in the header, there may be 36, 72, 108, or even more card images—but the number must always be a multiple of 36.

Some keywords are mandatory and some are optional. The mandatory keywords tell both humans and computers how the binary data that contain the image are structured—how many bits, lines, and samples make up the image. The optional parts of the header describe when and where the image was taken, what sort of camera took it, what the image shows, and who took it.

The binary array contains the image. Basic FITS supports five types of data: 8-bit unsigned data, 16-bit signed integers, 32-bit signed integers, and 16- and 32-bit floating-point data. The exact format for the data is defined rigidly because different types of computers and operating systems write binary data differently. Because the data format is rigorously defined, no matter what type of computer wrote the FITS file, your computer should be able to read it.

Finally, the tailer consists of the number of ASCII 0 (zero) characters necessary to pad the file length to a multiple of 2,880 bytes. The tailer is a throwback to the old days of computing when data stored on tape had to contain an integral number of records, but it is necessary today so that *every* FITS file conforms to the FITS standard.

The FITS standard admittedly makes tedious reading, but its fussiness means that your computer will be able to read images stored in the FITS format.

## 3.4 The FITS Header

The header consists of 36 card images of 80 bytes each, for a total of  $36 \times 80 = 2880$  bytes, which is an integer multiple of 36 card images. Figure 3.1 shows a typical FITS header containing 16 card images with keywords; the complete FITS header includes an additional 20 card images containing the ASCII space character (20 hex) to round out the length of the header to a total of 2880 bytes.

The header format looks almost trivially easy, and it is easy. Humans can easily read the header and make good sense of it. However, just one misplaced character is enough to trip up an unwary computer program. The FITS standard specifies every detail in the header to make FITS computer-friendly.

**Keywords.** Every card image begins with a keyword in bytes 1 through 8. The keyword identifies what sort of information appears in the rest of the card image. BITPIX, for example, tells how many bits are in the data, and OBJECT specifies the name of the object observed.

Keywords must be left-justified, 8-character ASCII strings with no embedded blanks, and if the keyword is less than 8 characters long, the space must be filled with ASCII blanks (20 hex). Keywords may contain upper-case characters A through Z, the digits 0 through 9, and the period and hyphen characters, but no others.

**Value Indicator.** For keywords that have an associated value, bytes 9 and 10 must contain the ASCII string “= ” (equal sign and space) followed by a properly formatted numerical value (a number) or a string value (usually words). For example, in Figure 3.1 the keyword NAXIS has an associated value and must be followed by a value indicator and a numerical value.

If the keyword has no associated value, bytes 9 and 10 may contain any ASCII characters. If the keyword does not require a value, then bytes 9 through 80 may contain any ASCII text. The keywords COMMENT and HISTORY do not have associated values so they do not require a value indicator, and they can be followed by any ASCII text.

**Values.** The value follows a value indicator and contains the value of the keyword. The value is the ASCII representation of a number or string. Both numbers and strings must be formatted according to the FITS standard. Numerical information is right-justified between bytes 11 and 30, and character strings begin with a single quote mark at byte 11 and cannot extend beyond byte 80.

**Comments.** An optional comment follows the value itself. The FITS standard encourages programmers to separate the comment from the value by a space followed by slash. Since comments are intended for human readers, their format is not specified except that a comment must not extend beyond the end of the card image.

## Chapter 3: Digital Image Formats

### 3.4.1 Mandatory Keywords

Certain keywords that are used to describe the structure of the file are mandatory. These keywords must be used exactly as described in the FITS standard. They are required in all FITS headers. The card images in the header of a CCD image must contain the keywords in this order:

- SIMPLE
- BITPIX
- NAXIS
- NAXIS1
- NAXIS2
- ...
- NAXISn
- END

Each keyword requires an 80-character card image. The mandatory card images are followed by enough card images of the ASCII blank character (20 hex) to bring the total number of card images to 36, for a header 2,880 bytes long.

**SIMPLE**. (Associated value is Boolean.) This is the first keyword in every FITS file. The value is the logical constant with the logical value T (true) if the file conforms to the FITS standard. The T must appear at byte 30. A logical value of F (false) signifies that the file fails to conform to the FITS standard in a significant way, such as storing array data in a signed or byte-reversed format.

**BITPIX**. (Associated value is Integer.) This is the second keyword in any FITS file. The value is an integer that specifies the number of bits representing a data value. Basic FITS can store both integer and floating-point values, but amateur astronomers will seldom need to store CCD images as floating-point data.

**NAXIS**. (Associated value is Integer.) This is the third keyword in any FITS file. The value is the number of axes in the data array. A value of zero signifies that no data follow the header. For one-dimensional data, such as intensity values in a spectrum, the value is 1. Image data are two dimensional, some number of samples wide by some number of lines deep, so the value should be 2. Color images can be stored in three-dimensional arrays of samples, lines, and colors, with a value of 3. Although FITS allows up to 999 axes, amateur astronomers will seldom if ever encounter data requiring more than three axes.

**NAXIS1, NAXIS2, ..., NAXISn**. (Associated value is Integer.) For each axis, FITS requires a keyword that specifies the number of elements along that axis. These keywords are NAXIS1, NAXIS2, NAXIS3, and so forth up to NAXIS999.

The dimensions given by NAXIS1, NAXIS2, and so on specify how the binary data that follow the header are organized. NAXIS1 is the axis whose index varies most rapidly, and NAXIS2 is the axis whose index varies next most rapidly.

For an image in which pixel values are read from the image sample by sample across and line by line down, as words are read from a page, NAXIS1 is the

**Table 3.1 BITPIX Values in FITS**

BITPIX	Type of Array Data	Example	Range
8	8-bit unsigned integer	127	0 to 255
16	16-bit signed 2s-comp integer	4095	-32768 to 32767
32	32-bit signed 2s-comp integer	1000000000	-2147483648 to 2147483647
64	64-bit signed 2s-comp integer	12345678912345	-9223372036854775808 to 9223372036854775807
-32	32-bit IEEE floating-point	3.141592	$1.2 \times 10^{-38}$ to $3.4 \times 10^{38}$
-64	64-bit IEEE floating-point	2.718281828459	$2.2 \times 10^{-308}$ to $1.8 \times 10^{308}$

number of samples in a line and NAXIS2 is the number of lines. However, if the pixel values were stored by reading down each column and reading the columns in sequence, then NAXIS1 would be the number of lines and NAXIS2 would be the number of samples.

EXTEND . (No associated value.) If the header contains more than 36 card images (totaling 2,880 bytes), the optional keyword EXTEND must appear among the first 36 card images, and must also appear in each subsequent block of 36 card images.

END . (No associated value.) This keyword is always the last keyword in the header. END has no associated value and no comment. Columns 4 through 80 must be filled with ASCII blanks (20 hex). END tells both computers and humans that they have reached the last keyword in the header.

### 3.4.2 Array Value Keywords

Array value keywords define the relationship between the contents of the data array and the real physical data. Although they are optional, they can be extremely useful and are often employed. However, if they appear in the header, they must be used as defined by the FITS standard.

BZERO and BSCALE . (Associated value is Floating Point.) These keywords were introduced before FITS allowed floating-point data arrays, with the original intention of allowing astronomers to convert values stored in an integer array into floating-point values. However, these keywords are now used to convert 16-bit unsigned integer CCD image data (*i.e.*, values that range from 0 to 65535) into the two's-complement integer format (values that range from -32768 to +32767) mandated by FITS. BZERO and BSCALE are defined as:

$$\text{pixel\_value} = \text{BZERO} + \text{BSCALE} * \text{FITS\_Value} \quad (\text{Eq. 3.1})$$

To convert signed 16-bit array values into unsigned 16-bit pixel values, BZERO is set to 32768 . 0 and BSCALE should be set to 1 . 0.

The values for BZERO and BSCALE must be floating-point numbers representing the offset and scaling terms in the equation above. The default value for

## Chapter 3: Digital Image Formats

the BZERO keyword is 0.0, and for BSCALE the default is 1.0. If either keyword is omitted, the value of the missing keyword is assumed to be its default value.

DATAMIN and DATAMAX. (Associated value is Floating Point.) These optional keywords specify the lowest and highest pixel values stored in the data array. If these values are present, FITS-reading software can use them to check that scaled data lie between DATAMIN and DATAMAX.

BUNIT. (Associated value is Character.) This is a string giving the units of pixel value, such as electrons or magnitudes/pixel.

BLANK. (Associated value is Integer.) This optional keyword specifies an array value reserved for missing or bad data, such as a defective row or column in a CCD image. The value most commonly used is -32678. If the BLANK keyword appears, the reading software should flag that any pixel assigned this array value contains no data.

### 3.4.3 Observation Keywords

After an observing session, image files can become separated from other records of the observation, so it is desirable to store important information in the file itself. This is especially true in amateur astronomy, where image files may be copied and forwarded many times.

OBJECT. (Associated value is Character.) This is a string giving the name of the object observed, for example, Messier 1, Crab Nebula, NGC 4565.

TELESCOP. (Associated value is Character.) This is a string identifying the telescope used, such as 6-inch f/5 Newtonian reflector, 10-Meter Keck Telescope, or Hubble Space Telescope.

INSTRUME. (Associated value is Character.) This is a string identifying the instrument used, such as SXV-H9, or SBIG ST10.

OBSERVER. (Associated value is Character.) This is a string identifying the observer(s), for example, Richard Berry, Al Kelly, or Dennis di Cicco.

DATE-OBS. (Associated value is Character.) This string identifies the date (and optionally the time) when the data contained in the array were acquired, and in the case of CCD integrations, the time of the *beginning* of the exposure. If possible, the date (and time) should be specified in Universal Time. The date must be specified using the YYYY-MM-DD format, where YYYY are the four digits of the year, MM is the number of the month, and DD is the day of the month. The string 2015-12-25 designates Christmas Day in 2015. The FITS standard encourages using DATE-OBS to hold the time as well as the date. Time is specified using the format hh:mm:ss.ss, where hh is hours, mm is minutes, and ss.ss is decimal seconds. The combined date/time format is: YYYY-MM-DDThh:mm:ss.ss, with the upper-case character “T” joining the date and time strings.

TIME-OBS. (Associated value is Character.) This string identifies the time at which the data contained in the array were acquired. The time is specified

using the format hh:mm:ss.ss, as above. Although TIME-OBS is a legitimate FITS keyword, the FITS standard encourages combining date and time in the DATE-OBS keyword.

**ORIGIN**. (Associated value is Character.) This is a string identifying the observatory, institution or organization responsible for creating the FITS file, such as Rainbow Observatory, NCSA, or KPNO.

**DATE**. (Associated value is Character.) This is a string giving the date on which the FITS file was created. The date must be specified using the YYYY-MM-DD format. July 4, 2010 would be written 2010-07-04.

**AUTHOR**. (Associated value is Character.) This string supplies the name of the person who compiled the data carried in the FITS file. This keyword applies less to images and more to the FITS files containing tabular data.

**CREATOR**. (Associated value is Character.) This is a string giving the name of the software package that created the FITS file, such as CBWinCam or **AIP4Win\_2.1.0**.

#### 3.4.4 Comment Keywords

Any number of comment card images may appear as long as the header's total length is an integer multiple of 36 card images.

**COMMENT**. (Character data.) This keyword takes no associated value and therefore has no value indicator. Columns 9 through 80 may contain any ASCII text. Any number of COMMENT card images may appear in a header.

**HISTORY**. (Character data.) This keyword has no associated value and therefore has no value indicator. Columns 9 through 80 may contain any ASCII text. The text should contain a history of steps and procedures associated with the processing of the associated image. Any number of HISTORY card images may appear in a header.

**Blank Card Images.** If bytes 1 through 8 of a card image contain ASCII blanks, columns 9 through 80 may contain any ASCII text. Any number of card images with blank keywords may appear in a header. Header information from software that uses a different set of keywords could be padded with 8 ASCII blanks for inclusion in a FITS header.

#### 3.4.5 Header Value Formats

Values in a FITS header can be Boolean, Integer, Floating Point, or Character. The correct value format is required for mandatory keywords.

**Boolean Value.** If the value is a logical constant, it must appear as an ASCII character T or F in byte 30 of the card image.

**Integer Value.** If the value is an integer, the ASCII representation of the integer must appear right-justified in bytes 11 through 30 of the card image. NAXIS, NAXIS1, and NAXISn take integer values. Some examples of valid integer values are: 0, 1, 2, 1000000, 123456789, -10000, and 32768.

## Chapter 3: Digital Image Formats

**Floating-Point Value.** If a value is a floating-point number, the ASCII representation must appear right justified in bytes 11 through 30 of the card image. The decimal point must be shown, and letters indicating exponential notation must be upper case. Note that the commonly used keywords BZERO and BSCALE require floating-point values. The following are valid FITS floating-point values: 0.0, 1.0, 3.14159, 1.0E6, -1.2345678E-09, and 3.276800000E004.

**Character String Value.** If the value is a character string, column 11 must contain a single quote (ASCII 27H) followed by the string starting in column 12. The end of the string is marked by a single quote that may not occur before column 20 and must occur in or before column 80. The character string must be composed solely of ASCII text. Proper interpretation of FITS data must not require reading more than the first eight characters of a character string.

### 3.5 The FITS Binary Data Array

The second part of every Basic FITS file is the binary data array itself. This section of the FITS file contains the image data described in the header. In Basic FITS, the binary data array consists of a single array of data. The first array value appears immediately following the end of the header; that is, the first byte of the data array occurs at byte 2881 in the FITS file.

A FITS binary data array is interpreted to be a byte stream; that is, bytes are read in the order that they occur in the file. The array values appear in sequence such that the array index varies most rapidly along NAXIS1, the array index along NAXIS2 varies next most rapidly, and so on.

Consider an image in which pixel values are stored by reading sample by sample across and line by line down as words are read from a page, so that NAXIS1 is the number of samples in a line and NAXIS2 is the number of lines. The data are written into the binary array in the same order that someone would type them if he had to type the data on a sheet of paper.

The data in the array are stored as bytes in a format that is not necessarily the same as produced by today's software and hardware. The programmer is responsible for writing the data in the format specified by the Basic FITS standard.

If the range of pixel values to be stored in the FITS file does not match the range of integer values available, the program should scale and offset the data, and place the BZERO and BSCALE keywords in the file header.

**8-bit Integer Data.** Eight-bit integer data must be represented by unsigned binary integers contained in one byte. The integer is equal to the ASCII value of the byte. The range of values in 8-bit integer data is from 0 (00 hex) to 255 (FF hex).

**16-bit Integer Data.** Sixteen-bit integers must be twos-complement signed binary integers contained in two bytes, with the most significant byte first; that is, what programmers call "big-endian" numbers. Valid array values run from -32768 to +32767 decimal (8000 to 7FFF hexadecimal).

**Table 3.2 Using 16-Bit Twos-Complement Integers**

<b>Unsigned Integer BZERO = -32768.0 BSCALE = 1.0</b>	<b>Signed Integer BZERO = 0.0 BSCALE = 1.0</b>	<b>FITS Data Bytes (Hexadecimal)</b>
0	-32,768	80 00
4,096	-28,672	90 00
28,672	-4,096	F0 00
32,512	-256	FF 00
32,752	-16	FF F0
32,767	-1	FF FF
32,768	0	00 00
32,769	1	00 01
32,783	15	00 0F
32,784	16	00 10
33,023	255	00 FF
33,024	256	01 00
36,864	4,096	10 00
61,440	28,672	70 00
65,535	32,767	7F FF

Because the signal read from CCDs is digitized as a 16-bit integer, raw CCD images are usually written into a 16-bit integer FITS files.

Most PC computing languages do not read or write data directly in this format because computers with Intel CPUs store data in “little-endian” byte order. Programmers should take care that their software reads and writes array values properly. There are several ways to store 16-bit unsigned CCD image data in a FITS file, depending on the quality and dynamic range of the CCD. Since amateur CCDs actually produce 12 to 14 bits of information, one option is to shift the 16-bit pixel values one bit to the left and store the data in the range between 0000 and 7FFF hex. Alternatively, if the dynamic range of the device is limited, then values above 32767 may be truncated at 32767. Another option, one that is most often chosen, is to include the BZERO keyword in the header and to offset the zero point by 32768.

**32-bit Integer Data.** Thirty-two-bit integers must be twos-complement signed binary integers contained in four bytes, with the most significant byte first; that is, in big-endian byte order. Array values run from 80000000 hex to 7FFFFFFF hex, or from decimal -2.139 trillion to +2.139 trillion.

**Floating-Point Data.** Floating-point array data must conform to the ANSI/IEEE-754 standard, which is described in the NASA-sponsored NOST draft standard for FITS. Data stored in IEEE-compliant 32-bit floating-point format has a precision of 1 part in 8,400,000 over a range of more than 75 orders of magnitude—more than adequate for lossless storage of processed CCD images. After calibration and processing, CCD image data can be stored in 32-bit floating-point

## Chapter 3: Digital Image Formats

format without fear of data loss.

### 3.6 The FITS Tailer

Because FITS was originally based on 2880-byte records read from 9-track tape, the specification required that if the data array did not fill the final 2,880-byte logical record, the record would be padded with the ASCII null (00 hex) character. This means that the length of every valid FITS file is a multiple of 2,880 bytes.

### 3.7 Nonconforming FITS Files

Software sometimes creates images that are labeled as FITS files but do not conform to the FITS standard. Even though images saved in these variant versions of FITS may contain perfectly valid image data, they may differ from the FITS standard sufficiently that software written to read conforming files cannot correctly read the variants.

FITS variants fall into three classes: those with nonconforming headers, those with nonconforming data arrays, and those in which the header and data arrays are not coordinated. Nonconforming headers contain nonstandard keywords or use standard keywords incorrectly. Nonconforming data arrays store data in a non-FITS format, or implement the FITS format incorrectly. Files with uncoordinated headers and data arrays have a conforming header and a conforming data array, but the header does not describe the data array correctly.

#### 3.7.1 Nonconforming Headers

The FITS standard allows programmers to create and use new keywords. However, it also specifies that additions to FITS may not override or replace the FITS standard. A case in point might be creating a keyword that invalidates the FITS standard data array.

As an example, imagine that you find a novel keyword DATATYPE in the header of a file. The value for DATATYPE is unsigned, and the data array does in fact consist of unsigned integers. Even though the DATATYPE keyword might make sense to you, it specifies a type of array data that does not conform to the FITS standard. Because this file does not conform to the FITS standard, the value of the first keyword, SIMPLE, should be the value false, or F.

#### 3.7.2 Nonconforming Data Arrays

The data format specified by the FITS standard can be inconvenient for programmers. FITS requires two's-complement signed integers with the most significant byte first. Computers that use Intel CPUs place the least significant byte first, so each time a FITS file is read from disk or saved to disk, the byte order must be swapped. (Byte swapping is the same as swapping the digits in decimal numbers, that is, 81 becomes 18.) By accident or by choice, the programmer may decide to

skip byte swapping. When software expecting a conforming FITS file reads these images, however, the result appears unrecognizable as an image.

Another way in which the FITS format makes life difficult for programmers is that FITS specifies signed 16-bit integers. This poses no problem with 10-bit, 12-bit, 14-bit, and 15-bit data because the array values are always less than the largest two's-complement signed integer, 32,767. With 16-bit data, however, the lowest pixel value in the image is usually 0, and the highest value is 65,535. FITS provides a simple solution: subtract 32,768 from each pixel value as the image is saved, and place the BZERO keyword in the header with a value of `32768 . 0`. The FITS standard recommends this for storing 16-bit image data. However, programmers sometimes give in to the temptation to save the data as unsigned 16-bit integers, so that a pixel value of 0 is written as 0000 hex, and a pixel value of 65,535 is written as FFFF hex. The result is a nonconforming FITS file.

### 3.7.3 Uncoordinated FITS Headers and Data Arrays

Finally, it is possible to have a valid header and a valid data array that do not work together. Consider a file in which the data array consists of correctly written signed two's-complement 16-bit integers, but the header specifies `BZERO = 0 . 0` and `BSCALE = 1 . 0`. The image will appear completely black with just a few very bright stars showing. The data array consists of values from  $-32,768$  to  $+32,767$ , but in order to scale it into the range of pixel values from 0 to 65,535, the header should specify that `BZERO = 32768 . 0`. When a valid header and a valid data array fail to coordinate, the image data stored in the file will not be read correctly.

Another variation on the theme is using valid keywords in a way that does not tell the software how to interpret the data array. Suppose that the DATAMIN and DATAMAX keywords appear in the header, with `DATAMIN = -32768 . 0` and `DATAMAX = 32767 . 0`. The programmer probably hoped that the FITS reading program would deduce the necessary offset and scale the data array values correctly. But DATAMIN and DATAMAX were not intended to function in this way; they specify only the range of pixel values that the image processing software will find in the data. They say nothing about how the data should be read or interpreted.

### 3.7.4 Reading Variant FITS Files

If a FITS file fails to load properly, check the header and data array. Make sure that the data array contains big-endian two's-complement signed integers, and that the header contains appropriate keywords.

The parameters to be deduced are the header length (usually 2880 bytes, or a multiple of 2880 bytes), the byte order, whether the data are signed or unsigned, and the offset to be added to the values found in the file. Usually it takes only three or four tries before a universal binary reader creates a reasonable approximation of the image stored in the file.

## 3.8 TIFF: The Standard in the Graphic Arts

TIFF is the image file format preferred in the graphic arts industry. Photo-editing software, paint programs, and desktop-publishing applications use TIFF because it is a powerful and flexible image format. However, this very flexibility is sometimes the undoing of TIFF. This standard supports black-and-white binary images, 8-bit grayscale images, 8-bit palette color images, 16-bit RGB color images, 24-bit RGB color images, as well as specialized graphics needs such as 32-bit CMYK color. TIFF files can be written in the PC's Intel little-endian byte order as well as Apple's Motorola big-endian byte order. TIFF even supports data in floating-point format, and allows proprietary data formats. In addition, the image data can be uncompressed, or it can be compressed using run-length-limited (RLE), Lemple-Ziv-Welch (LZW), or JPEG compression. Expecting one program to read every possible type of TIFF file is asking a great deal!

Furthermore, a TIFF file can store just one image or multiple images, and the images can be stored whole, broken into strips, or broken into tiles. The most common use for a second file is to store a thumbnail image for quick viewing.

The TIFF header is eight bytes long and begins with an identifier, either the characters II for Intel byte order, or MM for Motorola byte order. In the example shown in Table 3.3, the byte order is Intel ("II" = 49 49 hexadecimal). The byte order is crucial because all of the numbers that follow are hexadecimal. To load an image, the file must be interpreted correctly. Next comes the version number, which is always 42 (2A 00 in Intel byte order and 00 2A in Motorola byte order), followed by the offset in bytes to the first image file directory. The offset tells the TIFF reader how far into the file to look; in the Table 3.3 example, the program will look for the first image file directory that begins at the byte number 8 in the file (the Intel integer 08 00 00 00 equals 8), recalling, of course, that the first byte in the file is byte number 0.

An image file directory (IFD) is a subheader that provides information specific to each of the individual images that may be stored in the TIFF file. The IFD begins with the number of tags that describe the image (there are 15 tags in the example) followed by the tags, each 12 bytes long. The first two bytes are the tag number; the second two bytes are the field, which defines the type of data in the tag (3 = short integer, 4 = long integer, 5 = 32-bit fraction); the next four bytes tell how many data items the tag describes (always 1 in the example); and the last four bytes are either the value of the tag; *or* if the value is longer than four bytes, the byte offset to the location of the value in the file. In the example, all of the tags have their values except tags 11A hex and 11B hex, which give the byte offset to these data (two 32-bit fractions) located after the image in the tailer. Each IFD ends with the offset to the next IFD; an IFD of zero means there are no more IFDs.

Tags describe the image characteristics, such as tag 100 hex with the number of samples per line and tag 101 hex with the number of lines in the image. Bytes marked with XX XX XX XX must be replaced by appropriate hexadecimal values. In the example, the image is divided into strips each containing one line of

## Section 3.8: TIFF: The Standard in the Graphic Arts

**Table 3.3 Basic Grayscale TIFF Header**

Tag	Field	Count	Value	Comment
49 49				Byte order
2A 00				TIFF Version
08 00 00 00				Offset of first image file directory
0F 00				Number of tag entries (= 15)
FE 00	04 00	01 00 00 00	00 00 00 00	Tag 0FEh: New subfile type (0 = no)
00 01	04 00	01 00 00 00	XX XX XX XX	Tag 100h: Image width (= samples)
01 01	04 00	01 00 00 00	XX XX XX XX	Tag 101h: Image depth (= lines)
02 01	03 00	01 00 00 00	08 00 00 00	Tag 102h: Bits per sample (= 8)
03 01	03 00	01 00 00 00	01 00 00 00	Tag 103h: Compression (1 = none)
06 01	03 00	01 00 00 00	01 00 00 00	Tag 106h: Photometric interpretation (1 = grayscale)
11 01	04 00	01 00 00 00	C2 00 00 00	Tag 111h: Image offset (= 194)
15 01	03 00	01 00 00 00	01 00 00 00	Tag 115h: Samples per pixel (= 1)
16 01	04 00	01 00 00 00	XX XX XX XX	Tag 116h: Rows/strip (= samples)
17 01	05 00	01 00 00 00	XX XX XX XX	Tag 117h: Strip byte count = (Tag 100h * Tag 101h)
1A 01	05 00	01 00 00 00	XX XX XX XX	Tag 11Ah: X resolution offset = (Tag 111h + Tag 117h)
1B 01	05 00	01 00 00 00	XX XX XX XX	Tag 11Bh: Y resolution offset = (Tag 111h + Tag 117h + 8))
1C 01	03 00	01 00 00 00	01 00 00 00	Tag 11Ch: Planar configuration (=1)
28 01	03 00	01 00 00 00	02 00 00 00	Tag 128h: Resolution unit (= 2)
3D 01	03 00	01 00 00 00	01 00 00 00	Tag 13Dh: Predictor
00 00 00 00				Offset of next IFD (0 = none)

**Table 3.4 Tailer Data for the Basic Grayscale TIFF Header**

Tag	Field	Count	Value	Comment
48 00 00 00				Numerator of X resolution (=72)
01 00 00 00				Denominator of X resolution (=1)
48 00 00 00				Numerator of Y resolution (=72)
01 00 00 00				Denominator of Y resolution (=1)
00 00				End-of-file padding

the image.

In the grayscale image in the example, pixel values are written as a stream of bytes following the IFD. In the example, the image contains the product of the number of samples per line times the number of lines, and this number is stored in tag 117 hex. The X and Y resolutions are stored after the image, with offset equal to the sum of the header, IFD, and image size in bytes.

**Table 3.5 24-Bit Color BMP Header**

Header	Comment
42 4D	BMP file identifier (= BM)
XX XX XX XX	File size in bytes
00 00	Reserved (not used)
00 00	Reserved (not used)
36 00 00 00	Offset to start of image (= 54 bytes)
28 00	Image header length (= 40 bytes)
XX XX XX XX	Image width in samples (= 4 * (3 * (samples + 1) \ 4))
XX XX XX XX	Image depth in lines (= lines)
00 00 01 00	Number of planes (= 1)
18 00	Number of bits per pixel (= 24 bits)
00 00 00 00	Compression (0 = none)
XX XX XX XX	Size of bitmap
13 0B 00 00	X resolution (pixels per meter)
13 0B 00 00	Y resolution (pixels per meter)
00 00 00 00	Number of colors used (0 = maximum possible)
00 00 00 00	Number of colors in palette (0 = all colors possible)

A full description of the TIFF specification is available from the Adobe Developers Association at <http://www.adobe.com>.

### 3.9 BMP: Images for Windows

The Microsoft Windows Bitmap format (BMP) was developed for Windows/Intel computers, and is a convenient, simple, and widely used image file format for the Windows environment. The BMP format can store black-and-white binary images, 4-bit palette color images, 8-bit grayscale images, 8-bit palette color images, and 24-bit RGB color images.

The BMP header consists of two parts: the file header and the image header.

The file header is only 14 bytes long, and contains information necessary to open the image and load it into the computer's memory. The first two bytes are the characters BM to identify the file as a bitmap image; bytes 2 through 5 contain the file size in bytes; bytes 6 through 9 are reserved for future use; and bytes 10 through 13 contain the byte offset to the start of the image data.

The image header is 40 bytes long, and contains information needed to display the image. This includes the number of samples and lines, the number of bits per pixel, whether the image is compressed, and in the case of palette color, how many colors there are and how they should be handled. The BMP header example shown in Table 3.5 refers to a 24-bit color image, so it includes no information about the color palette.

The image data are stored in byte order, with three bytes (24 bits) per pixel

## Section 3.10: AVI: Interleaved Audio/Video from Webcams

in the example. 24-bit color information is stored in BGR order, and the y-axis follows the Cartesian convention, with the first line of the image at the bottom and the last line at the top of the image.

### 3.10 AVI: Interleaved Audio/Video from Webcams

Webcams generate a stream of images that, when shown in rapid sequence, produce the illusion of continuous motion. The AVI specification is a subset of Microsoft's Resource Interchange File Format (RIFF) designed to hold audio and video data. Every AVI file begins with the characters, "AVI\_" (where the underscore represents a space). Cascaded headers define the internal structure of the file and its contents. Inside an AVI file, however, the image data are stored in "chunks" that have a structure similar to that of BMP images.

For processing, an AVI is normally split into a sequence of individual 24-bit color images that are processed individually and reassembled into an enhanced AVI file, stacked into a single image, or selected on a frame-by-frame basis and then stacked into a single image. For generating movies from CCD images, a sequence of images can be processed, registered, and then combined in order and placed into an AVI file.

### 3.11 JPEG: File Compression for the Internet

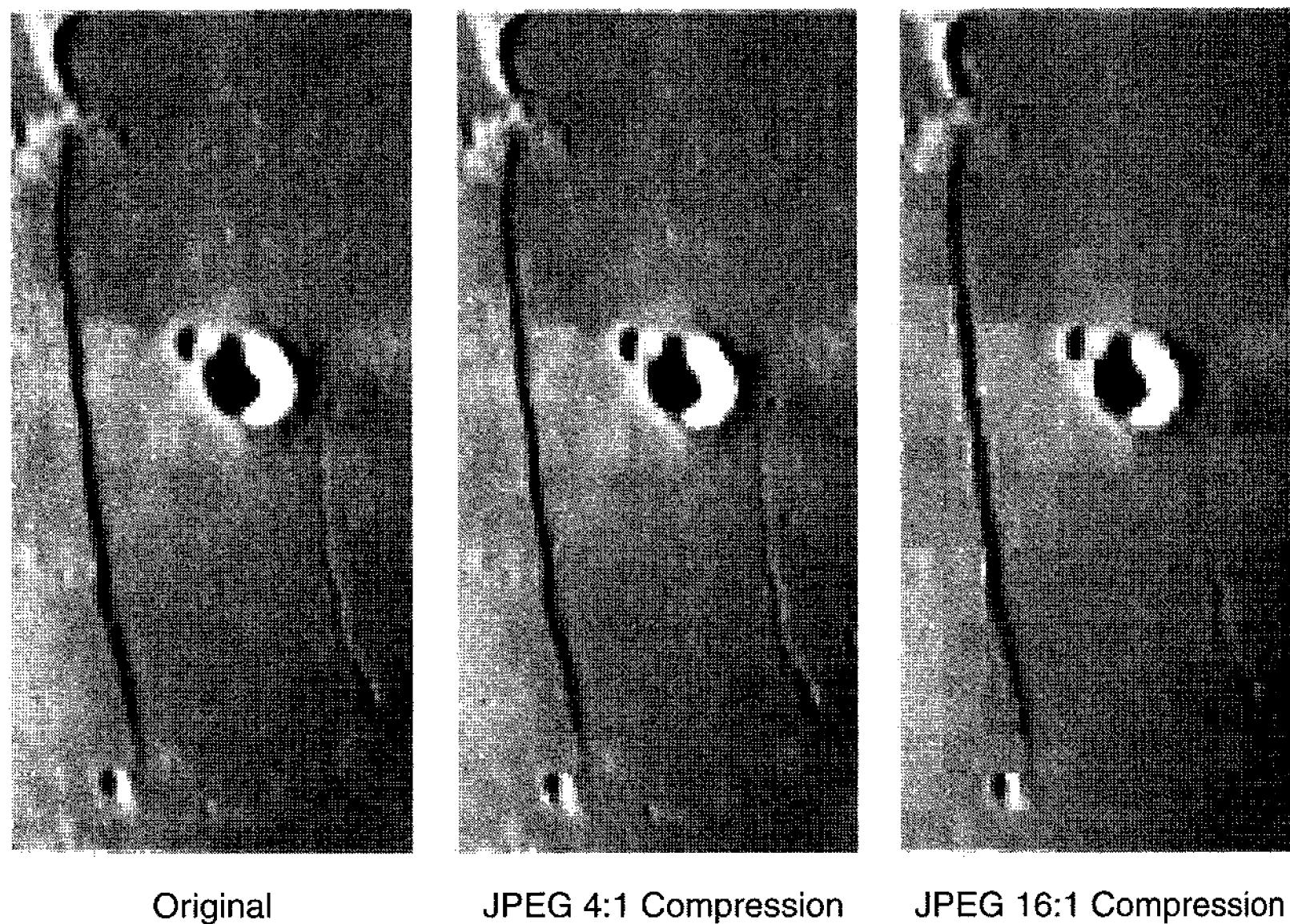
The Joint Photographic Experts Group is a standards organization that defined the method of data compression used in JPEG files, technically known as the JPEG File Interchange Format (JFIF), and informally as JIF or JPG. The JPEG data compression method is sometimes used in TIFF files, as well as other popular file formats.

JPEG exists to enable users to store and transmit 24-bit color images in significantly fewer bytes than a straight bitmap image would require. JPEG is a "lossy" compression scheme, meaning that image information is irrevocably lost when an image is converted to a JPEG file. JPEG's strength is that the information discarded by JPEG encoding tends to have been redundant information, but at compression ratios greater than 10:1, JPEG introduces artifacts that can be distracting and ugly. For fast transmission and on-screen display via the Internet, however, JPEG is an excellent file format.

The JPEG compression scheme is optimized for photographic color images because adjacent pixel values in photographs are similar in brightness and color. This enables it to extract average brightness and color values for groups of pixels and store the group value, rather than storing separate values for each individual pixel.

The first step in JPEG compression is to separate the luminance (brightness) information from chrominance (color) information. Rather than trying to represent color as RGB values, their common luminance (the Y channel) is split off, and the color is reduced to measures of redness (Cr) and blueness (Cb). Since the eye is more sensitive to small variations in brightness than it is to small variations

## Chapter 3: Digital Image Formats



**Figure 3.2** Enlarged sections of this image of the lunar Straight Wall show blockiness and loss of detail that occurs as a result of mild (4:1) and severe (16:1) image JPEG compression. Because smaller files mean speedier loading of web pages, however, some loss of quality is usually considered acceptable.

in color, chrominance is averaged over several pixels, resulting in a fuzzy quantized representation of the color information that requires fewer bits to be encoded.

Luminance is handled by breaking the image into  $8 \times 8$  blocks of 64 pixels and extracting spatial frequency information from each block using the discrete cosine transform (related to the Fourier Transform described in Chapter 17). The lowest spatial frequency is the average luminance of the  $8 \times 8$ -pixel block, followed by the progressively higher-frequency patterns of light and dark. Each pattern is present in some strength, which is represented by a coefficient. JPEG files include as many coefficients as needed to reach a user-determined image quality, and they drop the coefficients for higher-frequency patterns. At maximum compression, luminance information for 64 pixels takes only one byte (the coefficient of the average luminance), but for better definition in the image, more coefficients of the discrete cosine must be used, resulting in a better defined image with lower compression. At high compressions, the  $8 \times 8$ -pixel blockiness is obvious, and discrete cosine transform artifacts appear as a variety of stripes and plaid-like patterns.

JPEG files begin with a simple 20-byte header containing four bytes (FF D8 FF E0 hex) to mark the beginning of an image, a format identifier (the string

## Section 3.12: RAW, NEF, and CRW: Proprietary Raw Images

JFIF plus 00 hex), the JFIF version number, then a four-byte encoding of the pixel aspect ratio, and finally the number of samples and lines in the image. The compressed image then follows as a stream of bytes.

Because JPEG compression is lossy, never store images that will be measured or analyzed in a JPEG file. At best, the original information content of the image is reduced to 8 bits per color channel; and at worst, the image contains compression artifacts as well as brightness and color errors.

### 3.12 RAW, NEF, and CRW: Proprietary Raw Images

Most digital camera owners prefer the convenience of getting a finished JPEG image from their camera. However, a few oddball types such as amateur astronomers want to extract every scrap of information from the sensor—and for them, Nikon, Canon, and other high-end digital camera makers allow users to dump unprocessed output from the CCD or CMOS directly into a file for later analysis and processing. Unfortunately, the data format of these proprietary files is either entirely undocumented or very sparsely so.

Regardless of the sorry state of their documentation, however, these file formats have been reverse-engineered and can be read and displayed by a variety of non-proprietary software (including **AIP4Win**). The files necessarily contain information about the type of camera, the image size, the camera settings, the internal organization of the image data, and often a thumbnail copy of the image in a compressed format such as JPEG.

## Chapter 3: Digital Image Formats

---

# 4 Imaging Tools

---

Chapters 4 and 5 focus on what it takes to make great images. This chapter looks at the hardware side of image making—CCD cameras, telescopes, finders, mounts, and filters—the *equipment* you need to do the job. Chapter 5 focuses on *techniques* that produce good images; that is, how to use hardware. Perhaps the hardest lesson of all in digital imaging is that it takes *both* good hardware *and* good techniques to make great images.

## 4.1 Sensors and Optics

Understanding how the image sensor and the optical system interact allows you to use them more effectively and creatively. The sensor characteristics—the size of the array, the size of a single pixel, and its spectral sensitivity—cannot be changed; but you *can* control the angular field of view, the angular resolution, and the spectral range captured in the image through the use of various optical systems and filters.

### 4.1.1 Sensor Size and Field of View

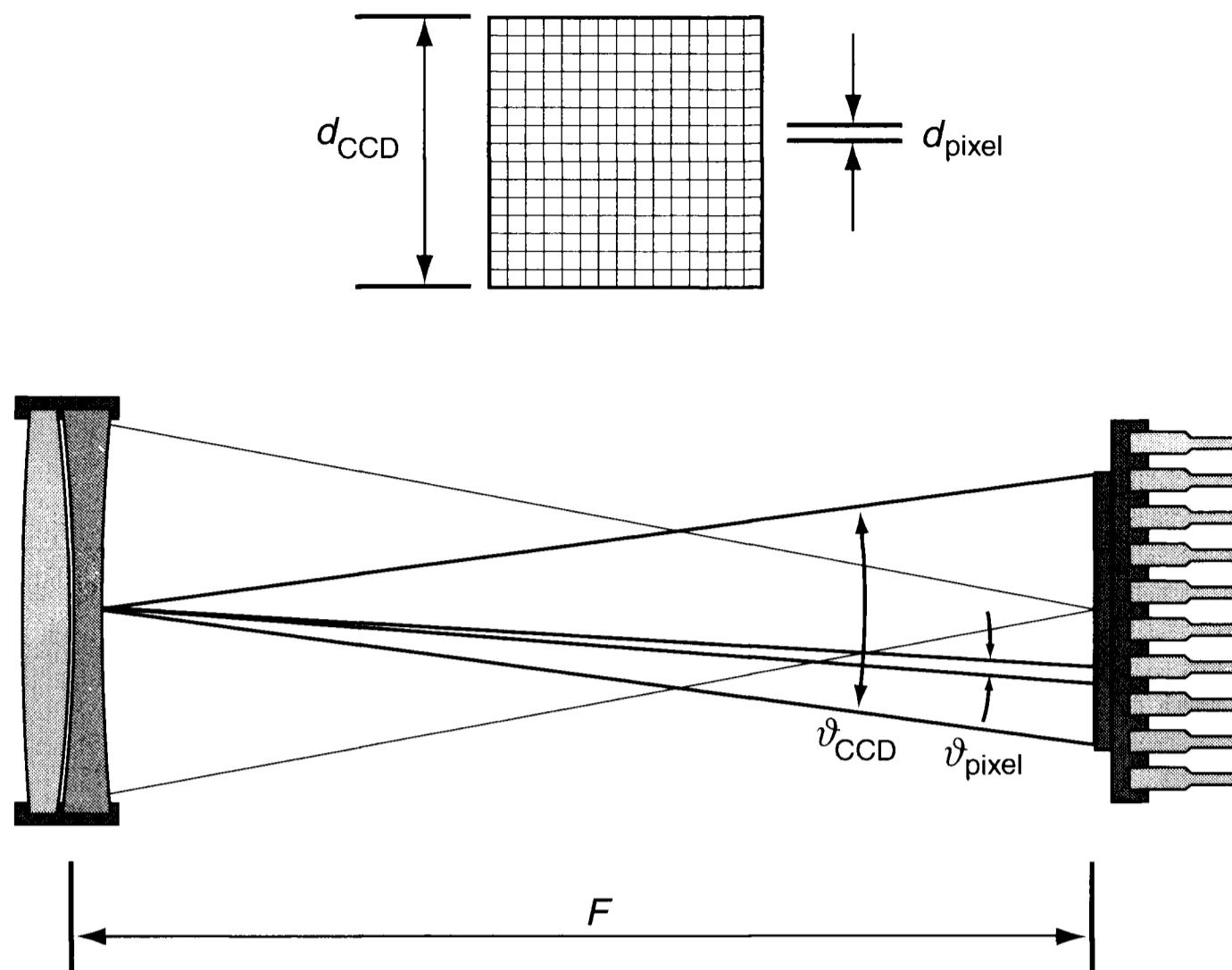
Field of view is the most important single factor in digital imaging—it determines how much sky your camera covers. The field of view depends on the focal length of the optical system and the physical size of the imaging area on the CCD chip. By putting different optics in front of your CCD camera, you can capture a wide-angle view or the tiny disk of a planet.

Calculate the field of view,  $\vartheta_{\text{CCD}}$ , produced by a CCD camera used on an optical system from the width or height dimension of the CCD,  $d_{\text{CCD}}$ , and  $F$ , the focal length of the optical system using:

$$\vartheta_{\text{CCD}} \cong 57.3 \times \frac{d_{\text{CCD}}}{F} \text{ [degrees]} \quad (\text{Equ. 4.1})$$

or:

$$\vartheta_{\text{CCD}} \cong 3439 \times \frac{d_{\text{CCD}}}{F} \text{ [minutes of arc].} \quad (\text{Equ. 4.2})$$



**Figure 4.1** The angular field of view of an imaging system depends on both the focal length of the optical system and the physical size of the detector. The angular size of a single pixel likewise depends on both the focal length of the optical system and the physical size of the pixel.

This formula is an approximation (that is what “~” above the equal sign means) that is accurate for long-focus systems such as telescopes and telephoto lenses. Be sure to use the same units—\_inches, millimeters, whichever you prefer—for the chip size and the focal length.

Here are some examples.

**To capture deep-sky objects.** All but a few of the Messier objects will fit comfortably into a 15-minutes-of-arc field of view, so that figure should be your minimum acceptable field size. Will a sensor that is 6.4 mm wide by 4.8 mm high be right for Messier deep-sky objects if you use it on a telescope with a focal length of 1,000 mm? From the equations, the field of view is 22 minutes of arc by 16.5 minutes of arc, so the combination has a field size large enough for making your own Messier Gallery.

**To capture a wide field of view.** Use optics with a short focal length. Suppose you want to make a CCD movie of a comet with a  $10^\circ$  long tail using the same imaging sensor—the 6.4 mm by 4.8 mm array; what lens do you need? Substituting  $10^\circ$  and the 6.4 mm dimension of the detector into the equation gives a focal length of 36.7 mm. A standard 35-mm focal length lens for a 35-mm camera should serve admirably.

**To capture the planets.** At the opposite extreme, imaging Jupiter—which

**Table 4.1 Angular Field of View for CCDs**

F mm	CCD Chip Dimension (millimeters)								
	2.5	3.2	4.8	5.6	6.4	8.0	12.0	16.0	24.0
100	1°26'	1°50'	2°45'	3°12'	3°40'	4°35'	6°52'	9°09'	13°41'
150	57'	1°13'	1°50'	2°08'	2°27'	3°03'	4°34'	6°06'	9°09'
200	42'	55'	1°23'	1°36'	1°50'	2°17'	3°26'	4°35'	6°52'
250	34'	44'	1°06'	1°17'	1°28'	1°50'	2°45'	3°40'	5°30'
400	21'	27'	41'	48'	55'	1°08'	1°43'	2°17'	3°26'
500	17'	22'	33'	38'	44'	55'	1°22'	1°50'	2°45'
700	12'	16'	24'	28'	31'	39'	59'	1°19'	1°58'
1,000	8'36"	11'	17'	19'	22'	28'	41'	55'	1°23'
1,250	6'52	8'48"	13'	15'	18'	22'	33'	44'	1°06'
1,400	6'08"	7'51"	11'	13'	16'	20'	29'	39'	59'
1,600	5'22"	6'53"	10'	12'	14'	17'	26'	34'	52'
2,000	4'18"	5'30"	8'15"	9'38"	11'	14'	21'	28'	41'
2,400	3'35"	4'35"	6'53"	8'01"	9'10"	11'	17'	23'	34'
3,600	2'23"	3'03"	4'35"	5'21"	6'07"	7'39"	11'	15'	23'
5,000	1'43"	2'12"	3'18"	3'51"	4'24"	5'30"	8'15"	11'	17'

**Table 4.2 Angular Pixel Size for CCDs**

F mm	CCD Pixel Dimension (microns)								
	8.0	9.0	10.0	12.0	13.75	16	18.0	20.0	24.0
100	16.5"	18.6"	20.6"	24.8"	28.4"	33.0"	37.1"	41.3"	49.5"
150	11.0"	12.4"	13.8"	16.5"	18.9"	22.0"	24.8"	27.5"	33.0"
200	8.25"	9.28"	10.3"	12.4"	14.2"	16.5"	18.6"	20.6"	24.8"
250	6.60"	7.43"	8.25"	9.90"	11.3"	13.2"	14.9"	16.5"	19.8"
400	4.13"	4.64"	5.16"	6.19"	7.09"	8.25"	9.28"	10.3"	12.4"
500	3.30"	3.71"	4.13"	4.95"	5.67"	6.60"	7.42"	8.25"	9.90"
700	2.36"	2.65"	2.95"	3.54"	4.05"	4.71"	5.30"	5.89"	7.07"
1,000	1.65"	1.86"	2.06"	2.48"	2.84"	3.30"	3.71"	4.13"	4.95"
1,250	1.32"	1.49"	1.65"	1.98"	2.27"	2.64"	2.97"	3.30"	3.96"
1,400	1.17"	1.33"	1.47"	1.77"	2.03"	2.36"	2.65"	2.95"	3.54"
1,600	1.03"	1.16"	1.29"	1.55"	1.77"	2.06"	2.32"	2.58"	3.09"
2,000	0.83"	0.92"	1.03"	1.24"	1.42"	1.65"	1.86"	2.06"	2.48"
2,400	0.69"	0.77"	0.85"	1.03"	1.18"	1.38"	1.55"	1.72"	2.06"
3,600	0.46"	0.51"	0.57"	0.69"	0.79"	0.92"	1.03"	1.15"	1.38"
5,000	0.33"	0.37"	0.41"	0.50"	0.57"	0.66"	0.74"	0.83"	0.99"

## Chapter 4: Imaging Tools

**Table 4.3 Typical Astronomical CCD Camera Specifications**

Camera	Cols	Rows	Pixel Size		Gain	RON	Full-Well
<b>Apogee</b>							
<b>AP1E</b>	768	512	9	9	n/a	15	100
<b>AP2E</b>	1536	1024	9	9	n/a	15	100
<b>AP4</b>	2048	2048	9	9	n/a	15	100
<b>AP6E</b>	1024	1024	24	24	n/a	15	200
<b>AP7</b>	512	512	24	24	n/a	n/a	n/a
<b>AP8</b>	1024	1024	24	24	n/a	n/a	n/a
<b>AP9E</b>	3072	2048	9	9	n/a	15	100
<b>AP10</b>	2048	2048	14	14	n/a	15	200
<b>AP16E</b>	4096	4096	9	9	n/a	15	100
<b>AP32ME</b>	2148	1472	6	8	n/a	12	50
<b>AP260E</b>	512	512	20	20	n/a	15	200
<b>Cookbook</b>							
<b>CB211</b>	192	165	13.75	16	40	30	150
<b>CB245 (252-wide)</b>	252	242	25.5	19.75	34	30	80
<b>CB245 (378-wide)</b>	378	242	17	19.75	27	20	80
<b>CB245 (756-wide)</b>	756	242	8.5	19.75	27	20	80
<b>Starlight Xpress</b>							
<b>HX516</b>	660	494	7.4	7.4	0.4	11	30
<b>HX916</b>	1300	1030	6.7	6.7	0.35	12	30
<b>MX5/SXV-M5</b>	510	290	9.8	12.6	n/a	11	60
<b>MX5C/SXV-M5C</b>	512	290	9.8	12.6	n/a	11	60
<b>MX716/SXV-M7</b>	752	580	8.6	8.3	1.3	10	70
<b>MX7C/SXV-M7C</b>	752	580	8.6	8.3	1.3	10	70
<b>MX916/SXV-M9</b>	752	580	11.6	11.2	2	12	100
<b>SX_500x256</b>	500	256	12.4	16.6	n/a	n/a	n/a
<b>SX_500x291</b>	500	291	12.4	16.6	n/a	n/a	n/a
<b>SXV-H9</b>	1392	1040	6.45	6.45	0.36	6.77	27
<b>SXV-H9C</b>	1392	1040	6.45	6.45	0.36	6.77	27
<b>SXV-M25C</b>	3024	2016	7.8	7.8	0.4	12	25
Table continued on next page...							

averages about 40 arcseconds in equatorial diameter—so that it spans half the image width, you need a field of view of about 80 arcseconds (1.33 minutes of arc). Substitute these values into the equation and find that you need a focal length of 16,500 mm. With a 10-inch (250mm) telescope, eyepiece projection to a focal ratio of  $f/66$  gives the desired image size. This calculation only takes the size of the image into account, and not the ability of the telescope to show detail—that is covered in the next section.

**Table 4.3 Typical Astronomical CCD Camera Specifications (Cont.)**

Camera	Cols	Rows	Pixel Size	Gain	RON	Full-Well
<b>Santa Barbara Instruments Group</b>						
<b>ST4</b>	192	164	13.75	16	n/a	n/a
<b>ST5</b>	320	240	10	10	n/a	25
<b>ST6</b>	375	242	23	27	n/a	n/a
<b>ST7 (ABG)</b>	765	510	9	9	2.3	15
<b>ST7 (NABG)</b>	765	510	9	9	2.3	15
<b>ST8 (ABG)</b>	1530	1020	9	9	2.3	15
<b>ST8 (NABG)</b>	1530	1020	9	9	2.3	100
<b>ST9</b>	512	512	20	20	2.8	13
<b>ST10</b>	2148	1472	6.8	6.8	1.5	9
<b>ST10 HF</b>	1092	736	6.8	6.8	1.5	9
<b>ST10 QF</b>	548	370	6.8	6.8	1.5	9
<b>ST237</b>	657	495	7.4	7.4	0.72	14
<b>ST2000</b>	1600	1200	11.8	11.9	0.72	15
<b>STV</b>	640	480	7.4	7.4	n/a	n/a
<b>STL-4020M/CM</b>	2048	2048	7.4	7.4	0.6	7.8
<b>STL-1301E</b>	1280	1024	16	16	1.6	18
<b>STL-1001E</b>	1024	1024	24	24	2	15
<b>STL-11000M/CM</b>	4008	2672	9	9	0.8	13
<b>STL-6303E</b>	3072	2048	9	9	2.4	15
<b>Southwest Cryostatics</b>						
<b>SWCryoPV</b>	565	512	27	27	n/a	n/a
<b>Finger Lakes Instruments</b>						
<b>IMG1300</b>	1280	1024	16	16	n/a	n/a

Pixel Size given in microns.  
Gain = Electrons per ADU.  
RON = Readout noise in electrons r.m.s.  
Full-Well = Full-well capacity in electrons.

Knowing the chip size in your camera gives you the ability to think creatively about many different imaging projects. Finding an optical system that gives your camera the desired field of view is all that stands between you and interesting new images.

#### 4.1.2 Pixel Size and Resolution

The size of the photosites in a sensor array determines the angular size of the smallest visible details in your images. For maximum resolution, the pixel should be small enough that the smallest angular details present in the image cover two or more pixels side-by-side. Remember, however, that high resolution may not al-

## Chapter 4: Imaging Tools

ways be desirable or even possible, especially if the field of view is an important consideration.

To compute the angular size of a pixel,  $\vartheta_{\text{pixel}}$ , use this equation:

$$\vartheta_{\text{pixel}} = 206265 \times \frac{d_{\text{pixel}}}{F} \text{ [arcseconds]}, \quad (\text{Equ. 4.3})$$

where  $d_{\text{pixel}}$  is the size of the pixel and  $F$  is the focal length of the optical system. Be sure to convert all measurements to the same units. To convert, recall that one micrometer (or micron) equals  $1/1000$  of a millimeter, and 1,000 micrometers (microns) equal one millimeter.

What is the smallest angular feature that a sensor with 7.5-micron pixels can distinguish on a telescope with a focal length of 1,000 millimeters? Substituting a pixel size of 7.5 microns ( $= 0.0075$  millimeters) and a focal length of 1,000 millimeters into Equation 4.3 gives the angular pixel size as 1.55 arcseconds. During exposures lasting more than a few seconds, telescope tremor, small tracking errors, and slight residual defocusing can enlarge star images from a tiny diffraction disk to 2 or 3 arcseconds. If the seeing is around 3 arcseconds (that is, rather poor), the match is excellent—but if the seeing settles down to 1.5 arcseconds, a longer focal length would be a better match for 7.5-micron pixels.

However, it is important to remember that in deep-sky imaging especially, resolution is not everything; many times, the field of view you need to cover an object will dictate the focal length of the optical system. This is a judgment call which you will need to make when you select equipment.

For planetary imaging, resolution becomes the *primary* consideration. If the seeing is good, the size of the diffraction disk is the limiting factor. To capture all the detail present in the telescopic image, the sensor must be able to fully resolve the diffraction disk.

Given a sensor with 7.5-micron pixels, what focal length is necessary to capture all of the detail on the planet Jupiter visible with a 10-inch telescope? The first step is to compute the angular diameter of the bright central region of the diffraction disk,  $\vartheta_{\text{HWHM}}$  (half-width at half-maximum), from:

$$\vartheta_{\text{HWHM}} \approx 206265 \times \frac{1.02\lambda}{A} \text{ [arcseconds]}, \quad (\text{Equ. 4.4})$$

where  $\lambda$  is the wavelength of the light forming the image and  $A$  is the aperture of the telescope. (For more detail, see Section 1.2.3.) The telescope aperture is 250 mm, and the effective wavelength for a typical sensor is 600 nanometers ( $600 \times 10^{-6}$  mm), yielding a diameter for the core of the diffraction disk of 0.51 arcseconds. To capture diffraction-limited detail, the Nyquist sampling theorem states that it takes two pixels to sample the core of the diffraction disk. One pixel should therefore fill an angle of 0.25 arcseconds at the focal plane.

Knowing that essential fact, you solve Equation 4.3 for the focal length;

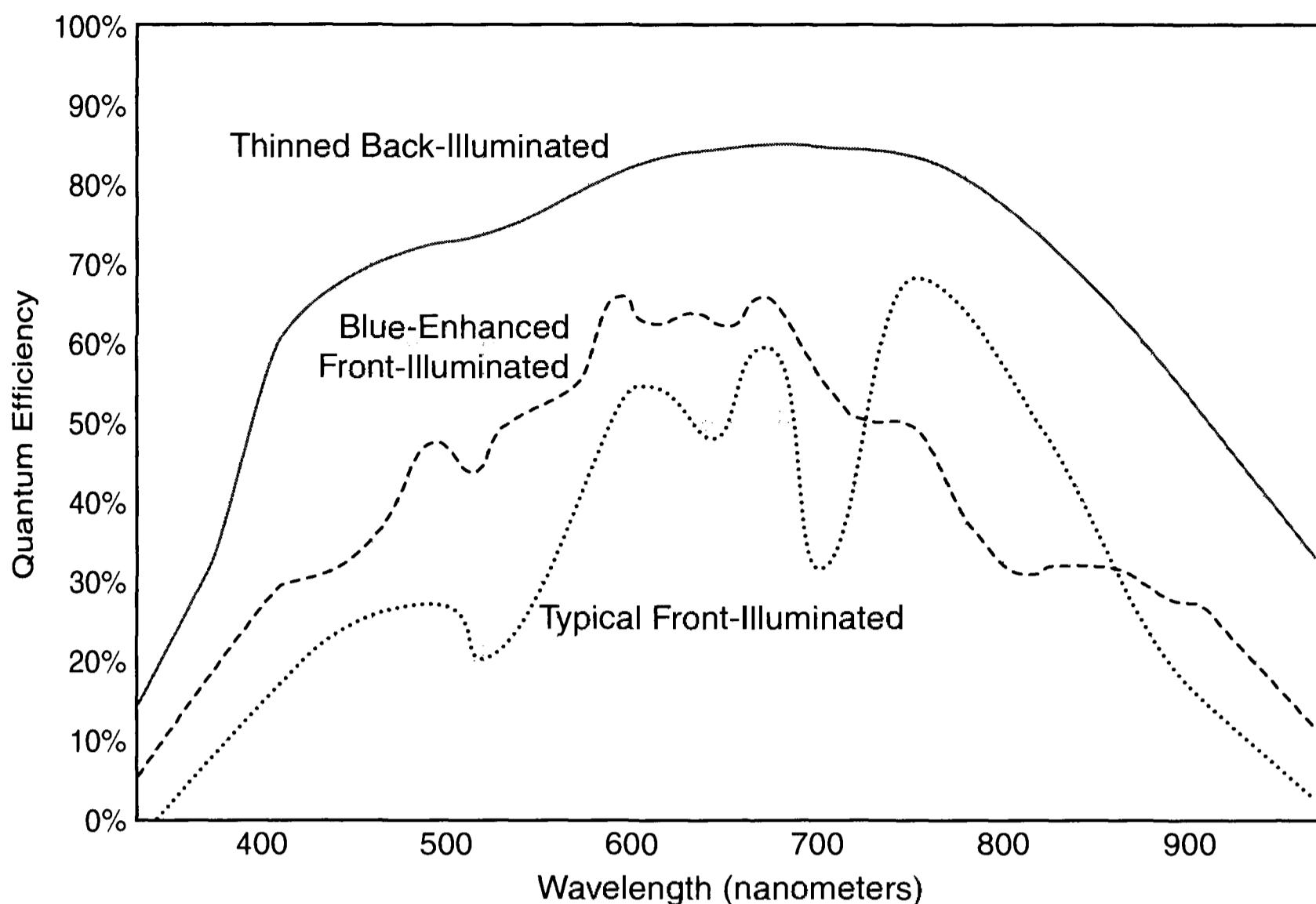


Figure 4.2 The spectral sensitivity of detectors varies considerably. Thinned back-illuminated CCDs offer high quantum efficiency over a wide range of wavelengths, but the front-illuminated detectors used by most amateur astronomers are considerably more rugged and much less expensive.

$$F = \frac{206265 \times d_{\text{pixel}}}{\vartheta_{\text{pixel}}} \text{ [millimeters]}, \quad (\text{Equ. 4.5})$$

and then for 7.5-micron pixels and 0.25-arcsecond resolution, you need a focal length of about 6,180 mm, corresponding to a focal ratio of  $f/25$  for the 250-mm aperture. At that focal length, the image of Jupiter will be about 160 pixels wide, so you don't need a very large sensor—in fact, a good-quality  $640 \times 480$  webcam would probably do quite nicely.

### 4.1.3 Spectral Sensitivity

Spectral sensitivity is the third important factor that determines what digital cameras capture in their images, and the one most often ignored because accurate spectral sensitivity curves have traditionally been reported in engineering units that are difficult to interpret. The information you need is the quantum efficiency of the image sensor as a function of wavelength. This measures what percentage of the photons (also called “quanta”) falling on the sensor get converted into useful signal. The highest possible value is 100%, implying that the sensor absorbs every photon and converts it into signal.

Specialized CCDs used by professional astronomers usually have a peak quantum efficiency between 70% and 90% that peaks at the red end of the spectrum (compare this to quantum efficiencies of around 1% for fast photographic films), and drops to around 20% for blue light. Typical *amateur* CCD cameras are

## Chapter 4: Imaging Tools

based on digital camera sensors designed to have a peak sensitivity of about 60% in the green or yellow-green at around 580 nanometers wavelength.

Any disparity between the red end and blue end of the quantum efficiency curve causes difficulties in color imaging—which is why amateurs prefer CCDs that peak in the visual part of the spectrum. Spectral sensitivity plays a key role not only in making color images—where you need good sensitivity in blue, green, and red—but also in determining the faintest objects that a CCD can image. At the best dark-sky sites, the sky is darker for blue wavelengths than it is for the red and near-infrared, because molecules in the upper atmosphere fluoresce in the red and near-infrared part of the spectrum. For the ultimate in deep-sky penetration from a dark-sky site, you need a CCD camera with a high quantum efficiency in blue light. However, in areas with lots of light pollution, or on nights when the Moon is bright, the sky is brightest in the blue, green, and yellow regions of the spectrum, and darkest in the red and near-infrared. For imaging from suburban sites, a red-sensitive CCD is best.

### 4.2 Optics for Imaging

Choosing the best telescope for digital imaging is worth an entire book, but the basics can be summarized in a few paragraphs. For any type of imaging, the choice of telescope is a trade-off among the desired field of view, the desired resolution, and the desired length of exposures. For a given aperture, a short focal length means a wide field of view and shorter exposures (or more data in same-length exposures) at some cost in resolution. Longer focal lengths mean more resolution, but a smaller field of view and longer exposures or fewer photons per pixel. To make the optimum trade-off, you must be willing to make trades among the “hard” factors—aperture, focal length, resolution, exposure time, tracking error, seeing conditions—discussed in the preceding sections, and also among the “soft” factors such as the cost of the desired telescope.

For more details, you may wish to consult a very useful book by Harrie Rutten and Martin van Venrooij, *Telescope Optics: Evaluation and Design*.

**Newtonian Reflectors.** The concave paraboloid of the Newtonian reflector remains the best bargain in photon collection available today, but aggravating factors such as tube length and field flooding often conspire to make the Newtonian less desirable than it could be. The simple paraboloid produces perfect diffraction-limited images on the optical axis. Away from the field center, images become increasingly comatic, meaning that star images grow short triangular tails. The faster the paraboloid, the greater the coma. The formula for coma is simple:

$$d_{\text{coma}} = \frac{3}{16} \frac{h}{F^2} \quad [\text{millimeters}], \quad (\text{Equ. 4.6})$$

where  $d_{\text{coma}}$  is the length of the coma blur,  $h$  is the off-axis distance in millimeters, and  $F$  is the focal ratio. To find what focal ratio gives coma-free images edge-to-edge with a  $24 \times 16$  millimeter sensor, begin by calculating the center-to-

edge distance. The distance from center to edge on the long dimension is 12 millimeters. For an  $f/8$  Newtonian, the length of the edge blur is 35 microns—but most of the light is concentrated in the pointed end, so the actual coma blur is more like 12 microns diameter. Given well-mounted optics and a strong tube, this Newtonian would be acceptable for most types of imaging.

However, it is possible to correct the paraboloid's coma using an auxiliary optical system a short distance ahead of focus. We describe a coma corrector in Section 4.3.4. A coma-corrected Newtonian telescope at  $f/4$  has approximately the same residual aberration blur as an uncorrected Newtonian at  $f/8$ .

**Achromatic Refractors.** Short-tube achromatic refractors look attractive for imaging because they are compact, offer a wide field of view, and are inexpensive. However, achromatic refractors suffer from longitudinal chromatic aberration; that is, light of different colors comes to focus at different distances from the lens. Although one wavelength is in perfect focus, other wavelengths are out of focus. For achromats of standard design, the formula for the chromic blur is:

$$d_{\text{chromatic}} = \frac{D}{1700} \text{ [millimeters]}, \quad (\text{Eq. 4.7})$$

where  $D$  is the diameter of the objective lens in millimeters. The focal ratio does not matter; therefore, for a 100 millimeter aperture achromat, the blur will be 59 microns in diameter. Unfortunately, this formula only takes colors from blue-green to red into account; deep blue and ultraviolet light will form an even larger blur. This means that images of bright stars are surrounded by a blue halo, colloquially known as “blue bloat.”

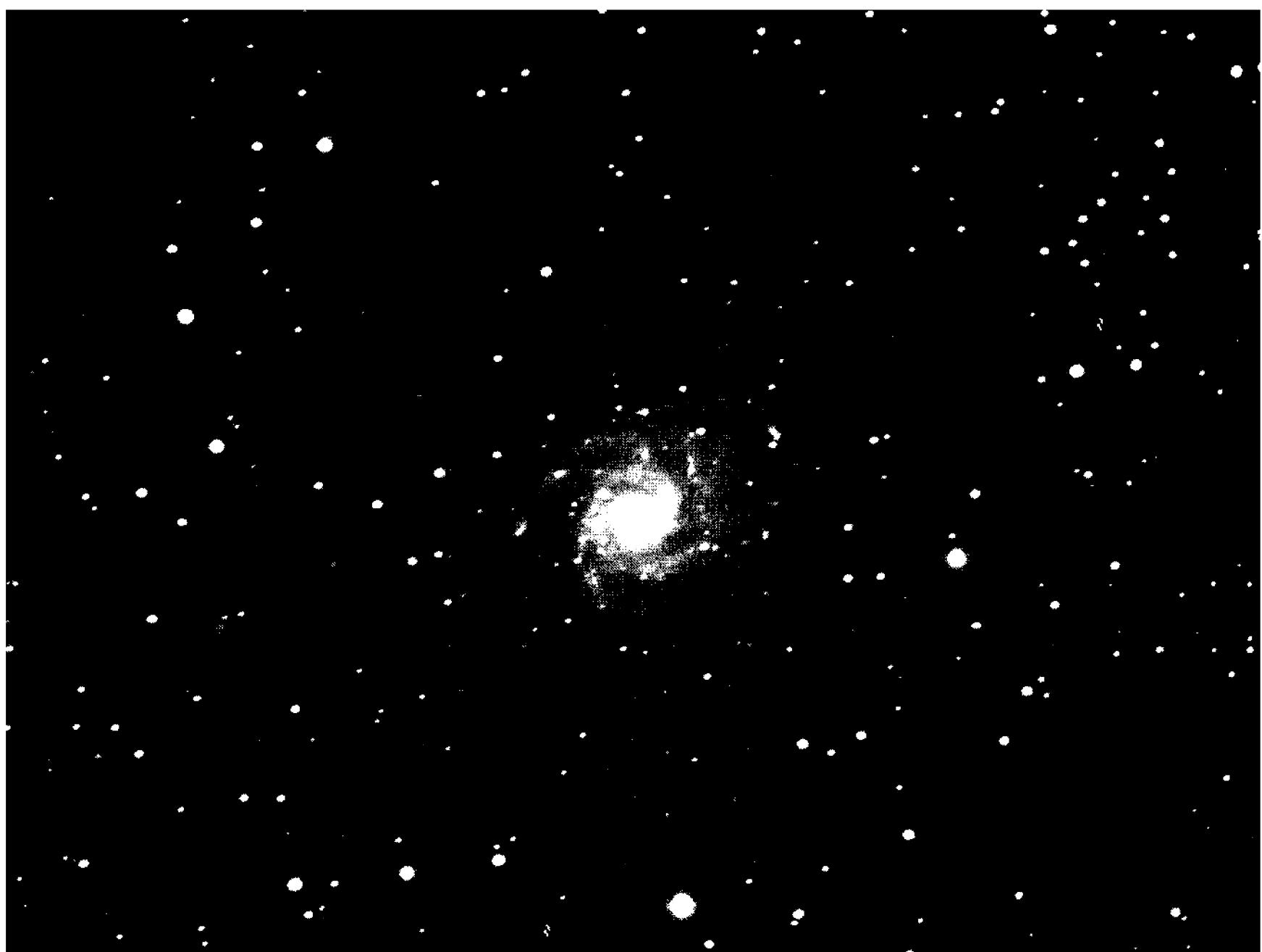
**Apochromatic Refractors.** Apochromatic refractors are made with special glasses that correct chromatic aberration more effectively than do achromats. While there are so many designs that it is impossible to treat them as a category, you can expect chromatic blurs one-third to one-fourth as large as those in an achromatic refractor—something in the range of 15 to 20 microns, with a smaller, bright core in the center of the blur. For most astronomical CCDs and digital SLR cameras, this much aberration is entirely acceptable.

Off-axis aberrations are another consideration. Apochromats may have coma, astigmatism, or field curvature. In field curvature, when the center of the field is in focus, the edges may be out of focus. Firms that manufacture apochromats intended for digital imaging may offer auxiliary field-flattening lenses to correct this aberration so that stars are in sharp focus across the entire field of view.

Hybrid refractor designs based on the Petzval lens—a system consisting of two widely separated doublet lenses—offer freedom from field curvature and astigmatism, but chromatic aberration remains a problem for such designs that use only “normal” glasses. By incorporating glasses with abnormal dispersion curves (the so-called “ED” glasses), chromatic aberration is controlled.

**Cassegrain Telescopes.** Two Cassegrain designs matter in digital imaging: the standard, or classical Cassegrain, and the field-widened Ritchey-Chrétien. Both are two-mirror systems. In the classical Cassegrain, the primary mirror is a

## Chapter 4: Imaging Tools



**Figure 4.3** The trio of images of M101 on this and the opposite page were made with the same astronomical CCD camera on optical systems of differing focal length. Made with an inexpensive 500 mm  $f/8$  mirror-lens, this image spans  $1^\circ$ , comparable to the deep-sky observer's view in a low-power eyepiece.

paraboloid and the secondary is a hyperboloid. The bulk of classical designs have a focal ratio between  $f/8$  and  $f/16$ . Although the classical Cassegrain suffers from coma, the blur size resembles that of a Newtonian of the same focal ratio, and is therefore often negligible. In the Ritchey-Chrétien, the curves on the two mirrors have been chosen to be free of coma, but astigmatism remains uncorrected. Both forms of the Cassegrain can be used with auxiliary lenses to yield wide, well-corrected fields.

**Catadioptric Telescopes.** Catadioptric instruments are primarily reflecting telescopes that incorporate at least one major refracting element. In these designs, the reflective elements perform the bulk work of bringing light to focus; the refracting element corrects one or more residual optical aberrations. The correcting element may be a thin, nearly flat, aspheric plate (in Schmidt forms) or a relatively thick meniscus lens (in Maksutov forms). In digital imaging, many catadioptric systems suffer from spherochromatism; *i.e.*, spherical aberration that is corrected for only one wavelength. Detailed examination of specific designs is required to determine their suitability for digital imaging; and, unfortunately, design details are usually considered proprietary.

**Camera Lenses.** For wide-field imaging, camera lenses offer a tempting and readily available source of optics. Many, however, offer performance that

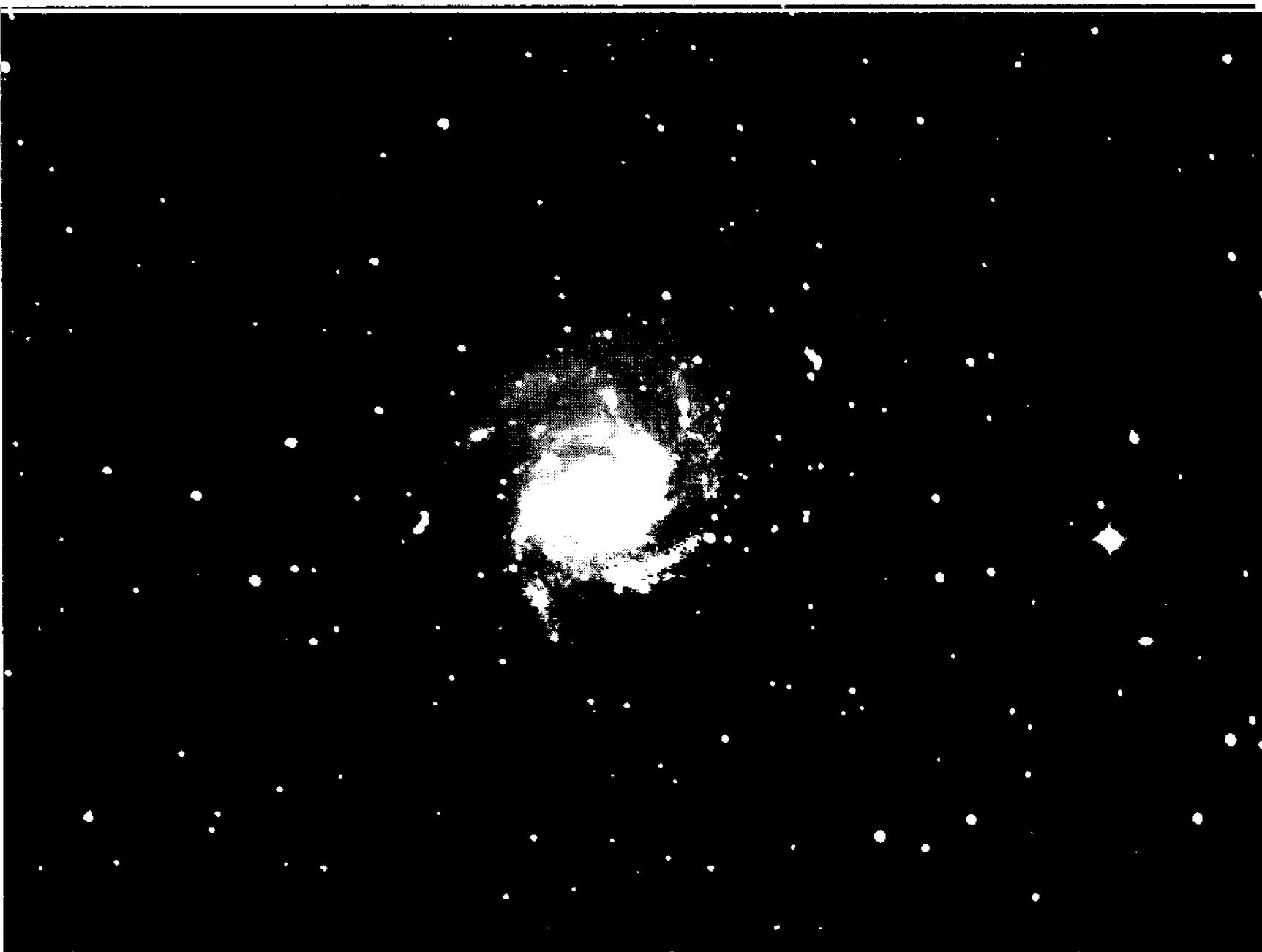
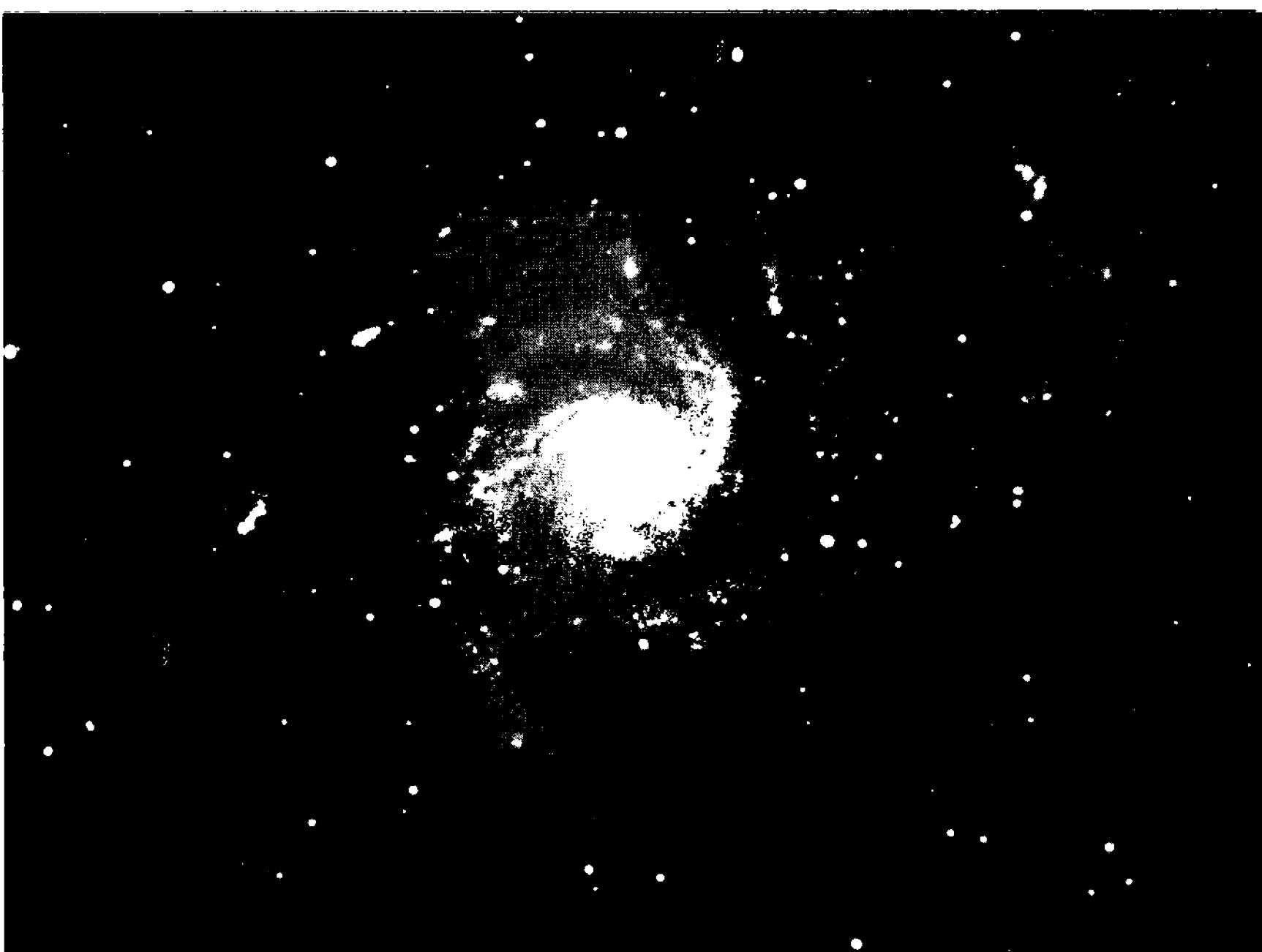


Figure 4.4 Increasing the focal length enlarges the size of the image but decreases the angular field of view; the decision is a trade-off. The image above was made with a 6-inch  $f/5$  Newtonian (750 mm focal length), while that below was made using a 10-inch  $f/6$  Newtonian (1500 mm focal length).



## Chapter 4: Imaging Tools

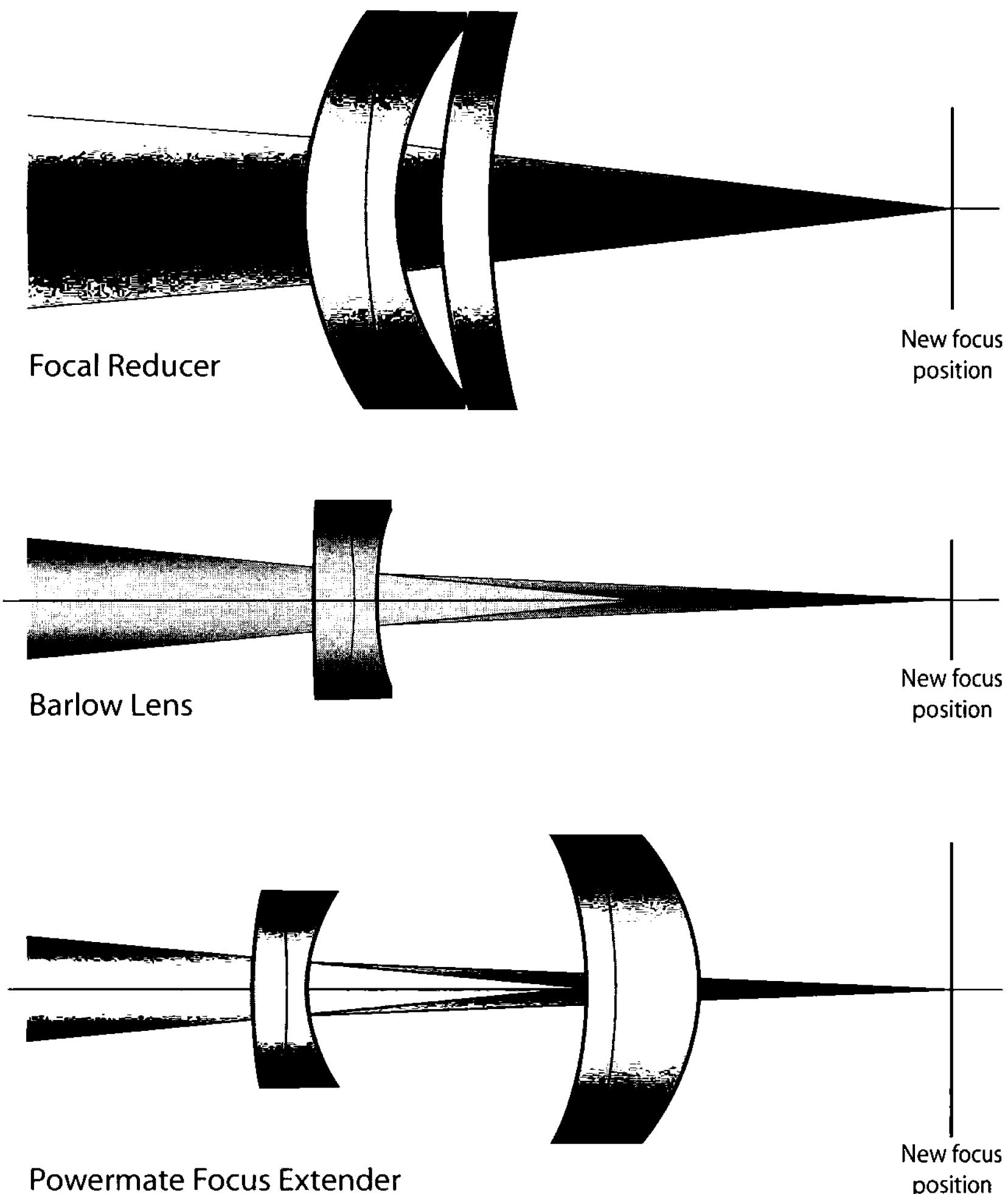


Figure 4.5 Focal reducers and focal extenders allow observers to alter the focal length of their telescopes. A focal reducer can compress a slow  $f/7$  focal ratio to  $f/5$  for deep-sky imaging, and a Barlow can lengthen a too-short 1,500 mm focal length to 7,500 mm for high-resolution planetary imaging.

leaves much to be desired. Lenses designed for film cameras seldom offer adequate control of chromatic aberration. At short wavelengths they suffer from enlarged images (the classic “blue bloat” problem with CCD cameras), and at the near-infrared wavelengths that astronomical CCDs respond to most strongly, from defocused images.

Camera lenses designed for use with digital SLRs can and do offer excellent performance, producing small, clean star images free of blue haloing. Not only do

these lenses incorporate ED glasses to cover the extended range of wavelengths that the digital sensors detect, but many have aspheric lens surfaces to correct residual spherical aberration. However, such lenses should be evaluated on a case-by-case basis.

## 4.3 Auxiliary Optics

Telescopes generally have a fixed focal length, but for your imaging projects, you may want to have more or less sky coverage, or to increase the resolution (sampling) of the optical system, or you may need to correct an aberration such as coma in the optics of the telescope. These goals are often accomplished through the use of some kind of auxiliary optics placed in the optical train between the telescope and the camera.

### 4.3.1 Reducing Focal Length

A focal reducer is a positive (i.e., converging) optical assembly that makes the focal length of a telescope shorter, condenses the image at the focal plane, and potentially increases the field of view seen by the sensor. A focal reducer provides a convenient way to shoot wide-field images with optical systems such as the standard  $f/10$  Schmidt-Cassegrain telescope.

Intercepting the converging light cone before focus, the focal reducer brings it to a shorter focus. They are rated either by the factor by which the original focal ratio is multiplied, or by the final focal ratio produced with an  $f/10$  Schmidt-Cassegrain. Focal reducers with  $0.8\times$  and  $0.5\times$  magnifications, and others rated to produce focal ratios  $f/6.3$ ,  $f/5$ , and  $f/3.3$  on  $f/10$  SCTs, are available commercially.

High-quality models are designed for use with a specific optical system, and deliver their rated focal reduction for a single spacing between the focal reducer and the focal plane. Increasing the nominal spacing lowers the reduction, and in some designs will cause significant vignetting and/or poor star images around the edge of the frame.

Focal reducing optics are sold in a variety of physical configurations: with 1.25-inch and 2-inch diameter barrels, with T-threads, with SCT-rear-cell threads, and with custom fittings for specific telescopes.

All focal reducers with T-threads share a common back-focus distance of 56 mm. The back-focus distance is measured from the shoulder of the T-thread to the focal plane, where the CCD chip must be located. In other models, the back-focus distance varies; and if you do not find this distance in the accompanying specification sheet, make every effort to obtain this important information from the manufacturer.

Because they shorten the converging light beam, images formed by focal reducers may suffer from vignetting. With small CCDs, vignetting may not be a problem, but at short focal ratios and with large detectors (such as those in digital SLR cameras) vignetting is a common problem. If your images show the characteristic dark corners due to vignetting, making flat field frames is essential.

## Chapter 4: Imaging Tools

### 4.3.2 Increasing Focal Length

To increase the focal length and angular sampling of the optical system, a negative (i.e., diverging) optical assembly called a *telenegative amplifier* or *Barlow lens* is placed in the beam of light converging to focus. These optics increase the focal length of the system, enlarging the image at the focal plane. The end result is that although the CCD covers a smaller field of view, each pixel samples a smaller patch of sky, potentially enabling it to record more fine detail.

This enhanced sampling may exceed the actual resolving power of the telescope, and be limited by the seeing conditions, but can be very useful in lunar and planetary imaging, especially when capturing fleeting moments of steady seeing. For high-resolution imaging with video cameras and webcams, oversampling the seeing works because among the hundreds or thousands of images captured, the very best moments of seeing can be identified and combined.

Barlow lenses are generally rated by the magnification factor they give the optical system, with factors of  $1.5\times$  through  $3\times$  being most common. However, the magnification of standard Barlows increases as the distance between the Barlow and the focal plane increases. For example, if you insert a filter holder between the Barlow and your camera, the magnification will be greater than the specified magnification.

Unlike most Barlow lenses, the TeleVue Powermate series was designed to provide nearly constant magnification regardless of the distance to the focal plane. The series provides  $2\times$ ,  $2.5\times$ ,  $4\times$  and  $5\times$ , magnifications well suited to planetary and lunar imaging. In addition, they offer the deep-sky observer interested in high-resolution images of bright objects (such as the Trapezium region of the Orion Nebula and many planetary nebulae) the opportunity to produce striking large-scale images.

Barlow lenses are manufactured with 1.25-inch and 2-inch barrels, and accept attachments with 1.25-inch and 2-inch barrels, as well as the 48-mm and T-thread attachments found on some cameras.

Although they provide much-needed ability to magnify the image, there are some drawbacks. As the spacing between it and the focal plane is increased, the Barlow lens itself must be moved further back, so it becomes necessary to add an extension tube to bring the image to focus. These optical constraints can result in unwieldy configurations.

Another difficulty you may encounter is optical vignetting, the loss of light at the edges of the field of view. With webcams, video cameras, and CCD cameras with small chips, a Barlow lens mounted in 1.25-inch barrel should pose no vignetting problem. However, for large-format CCD cameras and digital SLR cameras, a Barlow lens with a large clear aperture and mounted in a 2-inch barrel may be required to avoid this drawback.

### 4.3.3 Correcting Field Curvature

Ideally, your telescope produces its sharpest images on a plane surface, but in

## Section 4.4: Finding Celestial Objects

practice many produce their best images on a curved surface. Field curvature seldom hampers visual observing because your eyes refocus as they rove around the field of view. For imaging, field curvature is a problem because when the center is in focus, the edges of the field are out of focus, and *vice-versa*. With small image sensors this is not a problem because the zone of acceptable focus is larger than the sensor. However, large sensors reveal the out-of-focus stars at the edge of their large field of view. For example, a telescope that has served faithfully with small astronomical CCD cameras may suddenly show bloated star images with a digital single-lens-reflex camera.

In addition to their primary function, focal reducers sold for use with  $f/10$  Schmidt-Cassegrains may also correct the curved focal surface produced by these telescopes.

The manufacturers of high-end refractors usually offer field flatteners designed for specific models. These devices work well only with the telescope they were designed for, and are not usable on others. Since these field flatteners are photographic accessories, they are usually designed to provide full illumination over a 35-mm film frame, although some are designed to illuminate the field of a medium-format camera.

### 4.3.4 Correcting Coma

Coma is an optical aberration that causes stars at the edges of the field to appear as tiny “comet” shapes. The parabolic mirrors in Newtonian telescopes exhibit significant coma at fast focal ratios, as do many catadioptric optical systems. In Newtonians, the size of the coma blur is proportional to the inverse square of the focal ratio of the mirror—so an  $f/4$  mirror exhibits four times the coma of an  $f/8$ . In long-focus parabolic mirrors the effect is minor, but in short-focus paraboloids, coma can be significant.

A coma corrector is a lens assembly that corrects this aberration, providing a wide field for visual observation or imaging. Coma correctors are available in 2-inch barrel mounts, 48-mm threaded mounts, and custom-threaded units that fit specific telescope models.

As with most of the auxiliary optics described above, small CCD chips image only the central portion of the field; and the use of a coma corrector, particularly on a mirror  $f/5$  or slower, will probably not be necessary. But for large CCDs and the sensors found in digital SLR cameras used on  $f/5$  and faster paraboloids, a coma corrector will almost certainly be required.

## 4.4 Finding Celestial Objects

Three essentials for CCD imaging are finding, focusing, and following: you can’t make images if you can’t find celestial objects; images have little value if they are not in focus; and your telescope must follow an object for a reasonable length of time for a high-quality image. Equipment that makes these tasks easy is fundamental to long-exposure imaging.

## Chapter 4: Imaging Tools

### 4.4.1 Flip Mirrors, Finder Scopes, and Go-To Mounts

Many observers report some degree of difficulty finding celestial objects when they begin using an astronomical CCD or DSLR camera. The skills needed to find objects for imaging are no different than those needed for visual finding; the difference lies entirely in the accuracy required. While pointing within  $\frac{1}{2}^\circ$  is adequate for most visual observers, digital imaging is most efficient when you can accurately point the telescope within a few minutes of arc; pointing errors of  $\frac{1}{2}^\circ$  are frustrating, if not unacceptable.

Broadly speaking, telescope pointing aids fall into three classes: CCD camera finder functions, optical finding accessories, and accurate telescope mountings. The first of these should be built into the CCD camera acquisition software. The software finder mode usually consists of a high-speed CCD readout that puts a fresh short-exposure image on the screen every second or two. The value of a fast finder display cannot be overstated; instead of exposing for 30 seconds and then waiting another 15 to 45 seconds for the screen display to appear, a fast readout mode gives the observer real-time feedback on what the CCD camera sees. If the object is in the field of view of the camera, centering takes just a minute or less.

Given a CCD camera with a fast finder mode, an optical finder and an accurate telescope mounting insure that the object will remain in the field of view.

**Big Finders.** The finders that come as standard equipment on many telescopes provide too little light grasp and magnification to center deep-sky objects accurately. A large-aperture finder telescope allows the observer to see and center fairly faint objects well enough that a flip-mirror system may not be needed. Many observers own an old 60-mm or 80-mm refractor; these are obvious candidates to become auxiliary high-power finders. Depending on the size of the telescope, a re-tired 4-inch to 6-inch reflector also makes an excellent high-power finder.

Mount the auxiliary finder in ring brackets on the main telescope. If necessary, replace its original low-quality focuser with a standard  $1\frac{1}{4}$ -inch focuser, and add an illuminated cross-hair eyepiece that provides between 20x and 50x magnification. Once the new finder has been aligned, it should be easy to center any visible target within 1 minute of arc. A big finder is usually the most cost-effective way to locate and accurately center deep-sky objects.

**Unity-Power Finders.** Although Telrad-type finders offer neither magnification nor added light grasp, many observers swear by them. With the aid of a good star atlas, a skilled Telrad user can often point a telescope within  $0.2^\circ$  of the true location of a celestial object. This places the object close enough that the observer can then locate and center it with a big finder or a flip-mirror system.

**Flip-Mirror Systems.** These systems fit between the telescope and the CCD camera. They contain a swiveling or sliding mirror that directs the image from the telescope to a cross-hair eyepiece, allowing the observer to accurately center any object bright enough to be seen visually. Their great advantage is that they deliver all the light gathered by the telescope to the observer's eye.

Before you order a flip-mirror, check that your telescope has enough back

## Section 4.5: Telescope Mountings

focus to accommodate the extra length of the device. You should be aware that with fast optical systems especially, the flip-mirror housing may block parts of the converging beam, and cause unwanted vignetting.

In addition to its primary function as a finder telescope, when the image is sent to the eyepiece, the flip-mirror should block all light from the CCD camera, either with a separate dark slide or as part of the mechanism that moves the flip mirror. This allows the observer to use the flip-mirror to cap the telescope for taking dark frames, and also allows making dark frames while the observer finds and centers the next target object.

**Sky Atlas Software.** Bulky books and charts pose a significant obstacle to efficient imaging, especially when the wind is blowing and the temperature has fallen through the dew point. Since any observer using a CCD camera is already operating a computer, sky atlas software such as *MegaStar*, *TheSky*, *Earth-Centered Universe*, and *Guide* is a godsend. Not only do these programs display thousands of celestial objects, but they also allow the observer to set the magnitude limit and set the on-screen image orientation to match the eyeball view. Combined with a big finder, sky atlas software is an outstanding observing aid.

**Digital Setting Circles.** Although a quality set of analog setting circles can serve admirably, digital setting circles make locating objects easier with any telescope. These systems consist of digital shaft encoders and a dedicated computer unit to convert signals from the encoders into right ascension and declination. The computer unit can also store a database of hundreds or thousands of popular sky targets, and then signal when the telescope's position matches the coordinates stored in the unit's database.

**Go-To Mountings.** Computer-driven telescope mountings can make CCD observing extremely efficient. Today's market supports two classes of computer-controlled mountings: those designed for visual observers (typically accurate to about  $\frac{1}{4}^\circ$ ), and those that attain the 1-minute-of-arc accuracy that is desirable in CCD imaging. Before buying, an observer who is considering the purchase of an automated telescope should consult observers who already own the product for an honest assessment of its performance.

## 4.5 Telescope Mountings

The accuracy of your telescope's clock drive has a profound effect on how you shoot images with a CCD camera. The longer you integrate, the deeper your images. However, background sky brightness, blooming of bright stars, dark current in the CCD, those pesky "hot" pixels, and—most importantly—the accuracy of the telescope's drive limit the length of integrations.

### 4.5.1 Tracking Sensitivity

Unfortunately for CCD users, digital imaging requires tracking about 60 times more accurate than is needed for satisfactory performance for visual observing

## Chapter 4: Imaging Tools

and, except for CCDs with very large pixels, about three times better than is needed for deep-sky photography. Thus, a telescope drive that performs to your entire satisfaction for visual use may turn in a lackluster performance in digital imaging.

Poor tracking shows up as trailed, elongated, or enlarged star images. Although poor seeing and image pixilation tend to disguise tracking errors, when the latter reaches about half the width of a pixel, the problem becomes obvious. Not until the elongation of a sharply focused star image becomes less than about one-quarter pixel will star images pass muster under the demanding eye of a long-time observer.

### 4.5.2 Rate Errors and Periodic Errors

Drive errors fall into three broad categories: rate errors, periodic errors, and erratic errors. Rate errors occur because the drive motor is running either too slow or too fast, periodic errors occur because the telescope advances at a changing rate as the worm wobbles against the drive gear, and erratic errors occur because components in the drive are loose or separated.

**Rate Errors.** These are fixed by adjusting the speed of the drive to match the motion of the sky. If the gears in your telescope are driven by a synchronous 120-volt AC motor—the tip-off is that the motor can be plugged directly into a wall socket—you can build or buy a drive rate corrector or variable frequency oscillator to generate the required frequency. Drive correctors run from 12-volt DC or 120-volt AC inputs, and generate 120-volts AC current at a frequency that you can set, usually providing  $\pm 20\%$  above and below the base rate of 60 Hz. Telescopes driven by small DC pulse motors or stepping motors (recognizable by their characteristic “dut-dut-dut” sound) almost always allow the observer to change the basic rate, typically giving a choice of lunar, solar, sidereal, and variable rates. If the sidereal setting does not move the telescope at the correct rate, you will have to set the variable rate by trial and error.

**Periodic Errors.** Periodic errors mean that the average drive rate is correct, but the drive alternately runs slow and fast. A perfect worm and drive gear will exhibit periodic error if the axis of the worm does not lie in the plane of the drive gear, if the worm is not centered on its shaft, or if the worm shaft is bent. Commercial telescope systems made for visual observing often appear to suffer from poorly assembled gear systems, generating quite large periodic errors.

By definition, periodic errors repeat in a predictable manner, with the telescope usually running fast and then slow in a smooth sinusoidal or a lopsided sinusoidal pattern. The period of the cycle is almost always equal to the time it takes the worm to make one full turn against the drive gear. If the amplitude of periodic error in a drive is no more than three or four times the error sensitivity of the camera, periodic errors are more an annoyance than anything else because they can be corrected by one of the techniques described in Section 4.5.4.

**Erratic Errors.** If the worm does not press firmly against the drive gear, the mount may track erratically. The telescope may suddenly jump forward, track for

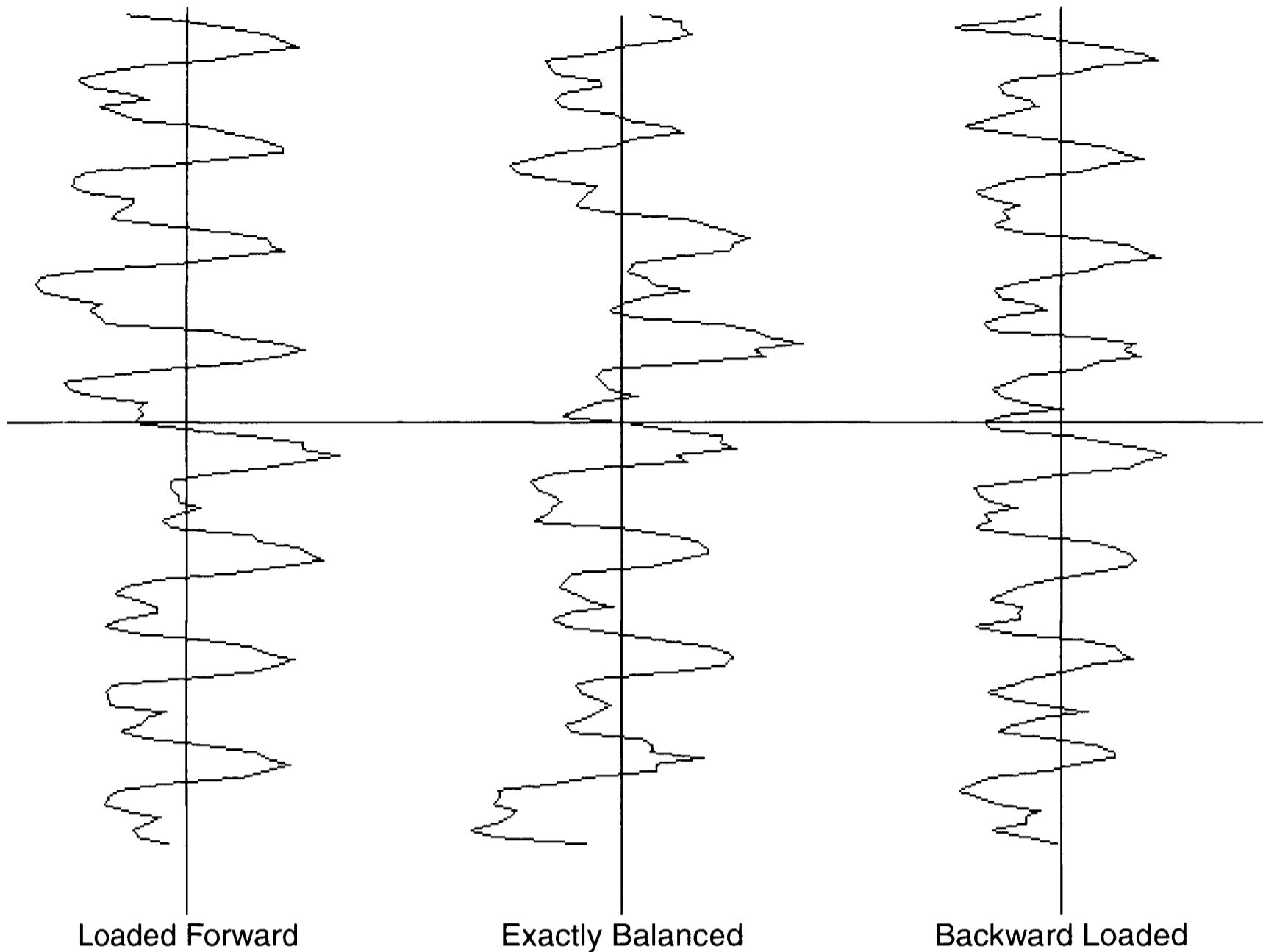


Figure 4.6 To test a drive, take many short integrations and then plot the position of a star on the images. This tracking test revealed that loading the drive so that it had to “lift” the telescope improved its performance. The period for this drive is 4 minutes, and the error is roughly 8 arcseconds peak-to-peak.

a minute or two, stop, and then repeat an approximation of the same cycle. Errors of this type are difficult or impossible to correct because the worm gear is not in continuous contact with the main drive gear, so the telescope does not respond to compensatory changes in the speed of the drive motor. The only recourse is to adjust or reassemble the mechanical parts of the drive system.

#### 4.5.3 Testing and Tuning a Clock Drive

CCDs make it possible to evaluate a clock drive’s tracking with remarkable accuracy. When you create a track-and-stack image, most software packages create a file containing a record of the shifts necessary to register the images. By plotting the shift values against time, you can see how well the drive is tracking, and if necessary, figure out ways to improve it.

On a night when the sky is less than ideal for normal imaging, shoot a sequence of images of a star field on the meridian and near the celestial equator. Make sure you have several reasonably bright stars in the field. Be sure to orient the camera so that north is at the top of the frame. If the camera supports different readouts, use a close-up or planetary mode to obtain the best resolution.

Shoot enough images for the worm gear to make several turns. If the image acquisition software for your camera supports an “autograb” or multiple imaging

## Chapter 4: Imaging Tools

mode, this will be easy; but if it does not, shoot the sequence manually. Make the integrations 5 seconds each at intervals of 15 seconds; this is short enough to give nice round star images but long enough to reach reasonably faint stars. Shoot a good set of dark frames to complete the data set.

Track-and-stack the images. As it operates, the software will save the  $(x, y)$  displacements measured in pixels. A good centroid algorithm is accurate to about 0.05 pixels. Here is a short excerpt from a typical tracking data file:

```
TRACK055.FTS, -4.584, -2.206
TRACK056.FTS, -4.214, -2.301
TRACK057.FTS, -3.809, -2.207
TRACK058.FTS, -3.901, -2.293
TRACK059.FTS, -4.170, -2.332
TRACK060.FTS, -4.738, -2.389
```

This particular sequence was part of a test to check the drive sector of a Byers 812 mounting. The full sequence of 337 images spans 84 minutes. The first datum is the name of the file, the second the shift in the  $x$  axis, and the third the shift in the  $y$  axis. You can analyze these measurement data with the aid of spreadsheet software or write a program to read the data and plot them.

- **Tip:** With **AIP4Win**, you can export tracking data to a spreadsheet such as Excel for analysis of your telescope's tracking errors. Use the Multi-Image function to measure the centroid of a star in a test set of images automatically.

The first step is to convert the tracking errors, which were measured in pixels, into arcseconds. Recall from Equ. 4.3 that the conversion factor is  $\vartheta_{\text{pixel}}$ , the width of a pixel in arcseconds:

$$\vartheta_{\text{pixel}} = 206265 \times \frac{d_{\text{pixel}}}{F} [\text{arcseconds}].$$

In the example, the focal length is 768 mm and the width of a pixel is 17 microns (0.017 mm), so for this particular system,  $\vartheta_{\text{pixel}} = 4.57 \text{ arc-seconds}$ .

In addition to any periodic error present, the tracking data will almost certainly contain an offset value,  $x_0$ , a rate of change,  $v$ , and an acceleration,  $a$ , or the rate of change of the rate of change. The rate of change is probably an error in the basic tracking rate, and the acceleration will probably be caused by atmospheric refraction changing the apparent motion of the star. By trial and error, you can plug in different values for the offset, rate error, and acceleration and plot the corrected position of the star,  $x$ , as a function of time:

$$x = x_0 + vt + \frac{1}{2}at^2. \quad (\text{Equ. 4.8})$$

For the mount in the example above, a value of  $x_0 = -5$  arcseconds placed the curve in the center of the graph, a value of  $v = 0.45$  arcseconds per image corrected the slow drift of the star, and a value of  $a = 0.0008$  arcseconds per image removed the slowly increasing drift rate. Since images were taken every 15 sec-

onds, the  $\nu$  value shows that during the test the drive was falling behind the sky at an average rate of 0.03 arcseconds per second, or 1 arcsecond every 33 seconds.

However, the really interesting result is the periodic error with an amplitude of 10 arcseconds. The drive was tested under three conditions: with the telescope perfectly balanced, loaded so that the imbalance pushed the telescope in the direction the drive was running, and loaded so that the imbalance opposed the drive, retarding it. You will probably find that when you run your drive against a light load, you get significantly better and more consistent tracking. This is particularly important with periodic error correction, which relies on a consistent periodic error.

If your telescope has a fork mounting with a short polar axis, examine the declination error by plotting the y-axis data. In these instruments, the pressure of the worm on the drive gear can force the polar axis to wobble in declination so that the star traces a cyclical path across the CCD. A mounting with a long polar axis shaft resists wobbling more effectively than a short-axis one.

If unbalancing the telescope does not improve drive performance, examine its mechanical parts. It is not uncommon to find that the worm assembly fits loosely or is lightly held in place by weak springs. Check that none of the components have worn or slipped out of place. Consultation with the dealer or manufacturer may help and should precede any work, to avoid invalidating the warranty. If the drive is supposed to track within 5 arcseconds and you measure a peak-to-peak tracking error of 30 seconds, at the very least the dealer owes you a polite response.

If you suspect that your telescope tracks poorly, your CCD can help you assess the performance of the drive. You may find that what you *thought* the drive was doing and what it *really* does are not the same. Armed with information instead of hunches, you can optimize the drive and your tracking strategy.

#### 4.5.4 Periodic Error Correction (PEC) Drives

Commercial telescopes may have periodic error correction (PEC) circuits in their drive electronics; that is, a programmable microprocessor that the observer can “train” to run at a rate calculated to offset a periodic error in the drive system. Typically, after training the PEC, the circuit eliminates about 90% of the periodic error. Although PEC is often used as an inexpensive way to produce marginally acceptable tracking from a poor drive gear, when PEC is applied to a drive that is already good, it produces superb correction.

## 4.6 Filters

Filters control the wavelength of light that reach a detector. In astronomy, there are many different uses for filters:

- color imaging at standard red, green, and blue wavelengths;
- isolating wavelength ranges for accurate scientific photometry;

## Chapter 4: Imaging Tools

- isolating specific spectral lines from astronomical sources;
- cutting light from mercury, sodium, and neon lamps; and
- reducing chromatic aberration in telescopes and lenses.

Filters are colored or coated glass plates that transmit some wavelengths of light while blocking other wavelengths; their role is to select what you see. To make color images, for example, we divide the spectrum into the same three ranges of wavelength that cone cells in the human eye sense using red, green, and blue filters. The red filter passes light from about 600 nm to 700 nm in wavelength and (ideally) blocks all other wavelengths. The green and blue filters likewise pass the 500- to 600-nm band (green) and 400- to 500-nm band (blue). From three separate filtered images, we can reconstruct full natural-color ones.

For the urban observer, light-pollution rejection (LPR) and short-cutoff filters perform yeoman duty blocking light pollution and improving the quality of star images in telephoto lenses and refracting telescopes.

The filters that astronomers use for scientific photometry resemble those used for color images, differing mainly because they block and pass somewhat different wavelengths. Filters that isolate spectral lines have long been used by professional astronomers, but since the advent of astronomical CCDs, amateur astronomers have discovered the beauty and power of hydrogen-alpha to reveal marvelous nebular details even on Moon-bright nights!

### 4.6.1 Filter Types

There are two basic types of filters: *dyed-glass*, and *interference*. In a dyed-glass filter, one or more metal ions are dissolved in the glass; they absorb some wavelengths while passing others. Interference filters are also called dichroic filters, and work by constructive and destructive interference in multiple thin layers deposited on the filter surface.

**Dyed-glass Filters.** There are three types of dyed-glass filters: short cut, long cut, and passband. Short-cut filters block short wavelengths below a cutoff wavelength while passing longer wavelengths. Long-cut filters block wavelengths above a cutoff wavelength while passing shorter wavelengths. Bandpass filters transmit a band of wavelengths and block shorter and longer wavelengths.

Short-cut dyed-glass filters generally exhibit sharp cut, high long-wavelength transmittance, and excellent short-wavelength blocking. The classic Wratten #25 red filter is a fine example and a fine filter. Long-cut dyed-glass filters are almost always problematic, with a broad cutoff, poor blocking, and low short-wavelength transmittance; the classic deep-blue Wratten #47 is typical in passing long-wavelength infrared better than they pass blue light!

Bandpass dyed-glass filters seldom exhibit sharp wavelength cutoffs, so they have broad passbands and tend to “leak” a fair amount of energy outside the passband. Even in the main passband, dyed-glass bandpass filters seldom transmit more than 50% of the incident photons, so much precious light never reaches the CCD. The classic green Wratten #58 is a good example. Before interference filters

became popular, the Wratten #25, #58, and #47 were the standard filter set used for color-separation imaging.

Dyed-glass filters are available as glass disks, or mounted in 1.25-inch and 2-inch threaded filter mounts.

**Interference Filters.** Dichroic filters are made by evaporating many fine layers of different dielectric materials (such as magnesium fluoride) onto the surface of a disk of optical glass. Interference filters are painstakingly designed, and they work by selectively reflecting photons having wavelengths that the designer does not want passed. The design is realized by adjusting the thicknesses of the dielectric layers. Dichroic filters have a characteristic “shiny” appearance because they reflect the wavelengths that do not pass.

Interference filters often have steep passband profiles, high transmittance in the desired passband—and wavelengths outside the design passband can be almost entirely blocked. A skilled designer can produce a wide range of filter transmittance curves, from wide-band filters well suited to color imaging to narrow-band filters suitable for H $\alpha$  imaging. Compared to dyed glass, interference filters are amazing—a dyed-glass filter will have a bandpass of 100 nm with a peak transmittance of 50%, whereas an interference filter will transmit 90% of the incident photons in a passband less than 10 nm wide.

For astronomical imaging, an interference filter with its passband centered on the wavelength of an astrophysically important element (such as hydrogen) yields a picture in the light of just that one type of atom. Meanwhile, only the tiniest sliver of broadband sources (such as Moonlight) gets through the filter, and the wavelengths of city streetlights are fully blocked.

Despite their technical elegance, two characteristics of interference filters can and do cause some problems for the digital imager:

1. Tilting an interference filter causes the wavelength passed by it to shift to longer wavelengths than the design wavelength.
2. With fast optical systems, off-axis light impinging on the filter at a sufficiently steep angle may lie outside the filter passband.
3. The reflective surface of the filter facing the reflective surface of the CCD chip may cause ghost images and haloes around bright objects.

These problems must be taken into account, but they are not severe. The first only requires that you take care to mount the filter perpendicular to the incoming light. The second is an issue only with optical systems  $f/4$  and faster, and can be handled by using a filter with sufficiently wide passband to include the wavelengths of interest. The third is the most difficult to cure, but it occurs only when you try to image faint objects that are near very bright ones.

Interference filters are available in 1.25-inch and 2-inch threaded mounts. Bear in mind that interference filters do have delicate surfaces, and are subject to attack by moisture and chemicals. With care, however, they can and do render excellent service for years.

## Chapter 4: Imaging Tools

### 4.6.2 Filters for H-Alpha and Other Emission Lines

When an atom is struck by a photon with sufficient energy, its outmost electron jumps to a higher energy state, and then eventually falls back. When it does, it emits a photon at one of several wavelengths. These wavelengths are characteristic for each atom. Filters that selectively pass only the strongest of these spectral lines for certain atoms are called *emission line filters*.

Emission line interference filters are available for many of the bright nebular emission lines. These have exceptionally narrow passbands, ranging from 10 nm down to around 4 nm. In the visual spectrum, nebular emission lines include:

- doubly-ionized oxygen (OIII) at 500.7 nm wavelength,
- hydrogen-alpha ( $H\alpha$ ) at 656.3 nm wavelength,
- ionized sulfur (SII) at 672.6 nm wavelength.

Because they block all other wavelengths, narrow-band filters increase contrast, bring out fine detail in objects where it is present, and allow deep imaging from bright-sky locations. It is important to find a good compromise between competing needs. Rejecting unwanted light to get maximum contrast calls for a narrow bandpass, yet accepting all of the light from a fast optical system calls for a wider bandpass.

It is possible to make vivid pseudocolor images by joining OIII,  $H\alpha$ , and SII images into a color image, using the shortest of the wavelengths (OIII) as blue, the middle wavelength ( $H\alpha$ ) as green, and the longest (SII) as red. Many of the Hubble Space Telescope's nebula images use this particular color-coding scheme—and it works equally well for amateur astronomers.

### 4.6.3 Light Pollution Rejection (LPR) Filters

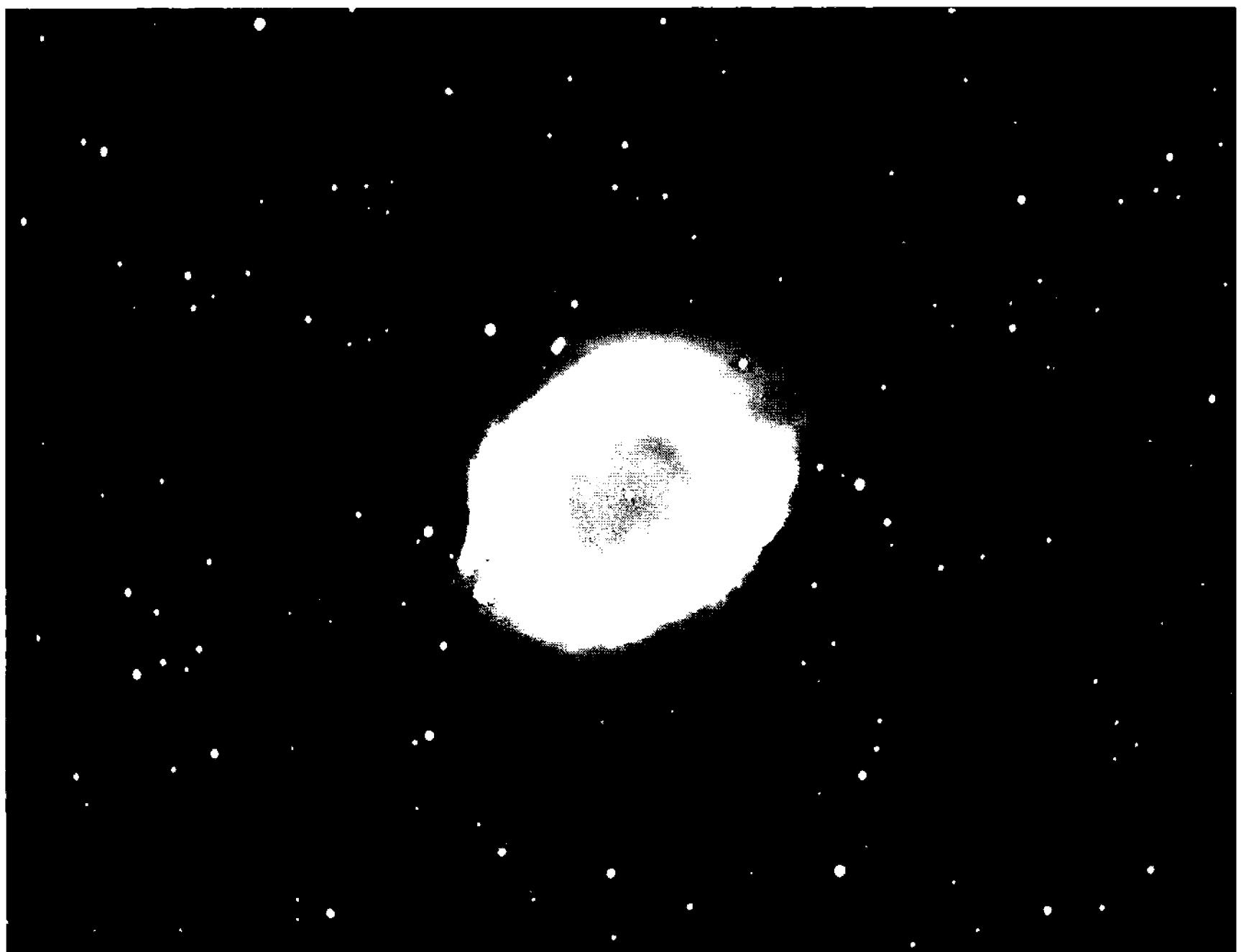
Filters used to block the emissions of low-pressure and high-pressure sodium- and mercury-vapor lamps are called light-pollution-rejection (LPR) filters. Visual observers have long known that viewing through LPR filters enhances the contrast of faint objects, helping them stand out against a black sky background. LPR filters are usually designed to pass the  $H\alpha$  and OIII emission lines that are strong in emission nebulae. LPR filters designed for the needs of CCD imaging can really improve the images taken under moderately light-polluted skies.

LPR filters are effective from urban areas because they cut background sky light. The high background light level limits exposure times by creating a “plateau” on which the signal from the target sits. As a result, stars, nebulae and galactic cores reach saturation during relatively short exposures. The LPR filter attenuates the wavelengths from man-made sources, dropping the background and leaving more “headroom” for the celestial object being imaged.

They are available in 1.25-inch and 2-inch-barrel threaded sizes, SCT rear-cell threads, and threads for standard camera lenses.



**Figure 4.7** Filters are an essential component in the CCD imager's toolbox. The image above shows the Helix nebula imaged with a wide-band red filter; the image below shows the same object imaged with a narrow-band H $\alpha$  filter. By suppressing sky light, the Helix stands out against a dark background.



## Chapter 4: Imaging Tools

### 4.6.4 Blue-Block and Violet-Block Filters

Many refractive optical systems suffer from chromatic aberration, in which all wavelengths do not focus at the same point. This is true of simple achromatic objectives, ED-glass objectives, and apochromatic objectives to varying degrees. Refractors that are designed for visual observing can seldom accommodate the range of wavelengths that the astronomical CCD “sees” clearly.

In imaging with refractors, when an observer focuses using short exposures, the image appears sharp because the user is focusing primarily on the electrons from the red photons to which the camera is more sensitive. Not seen are the electrons generated from blue and violet photons at the short end of the spectrum. After a long exposure, however, every bright star is surrounded by a halo of out-of-focus blue and violet light. For color imaging this is especially troublesome because every bright star ends up with an obvious blue halo.

One solution is to add a filter that selectively blocks photons from the short end of the spectrum, usually below 420 nm wavelength (violet light). With the out-of-focus light excluded from the optical system, star images look small and sharp, and don’t have the tell-tale “blue-bloat” halo. Minus-violet and minus-blue filters are available in 1.25-inch and 2-inch threaded mounts as well as cells threaded to fit many camera lenses.

## 4.7 Recognizing and Correcting Equipment Problems

All images suffer from a certain number of residual defects—streaks, black blobs, dark corners—the sorts of things that flat-fielding is supposed to eliminate, but does not always remove entirely. On astrophotographs, the same defects are often present, but they are small and hard to see. Because we inspect digital images so closely and apply powerful image-processing routines to enhance contrast, these same defects stand out with great clarity.

Common types of image defects are hot spots, field flooding, vignetting, and dust donuts. Field flooding and hot spots occur when light that should not reach the sensor falls on it anyway. Vignetting and dust donuts result when light that should fall on the sensor is blocked and does not reach it.

When reflections from focus tubes, flip mirrors, and focal-reducer lenses cause unwanted light to reach the focal plane, you get hot spots. You cure them with baffles and paint. An important variation on hot spots is field flooding: it occurs when the whole focal plane (rather than the center only) is awash in stray light. Internal baffling and a long “snoot”-type light shield on the front of the telescope tube greatly reduce or entirely cure field flooding.

Vignetting results from things that block light on its way to the focal plane; you deal with it by removing the obstructions. Too-small focusers are the most common cause. Dirt on the optics and dust on the CCD cause dust donuts; cleanliness is the fix. After removing light blockers and cleaning, what remains of vignetting and dust donuts can be cured by shooting good dark frames and flat fields so you can calibrate them out.

## Section 4.7: Recognizing and Correcting Equipment Problems

It is easy to get upset about image defects, but getting upset doesn't help much. Take an analytical approach to image defects: figure out what causes them so you can eliminate, greatly reduce, or calibrate them out of your images.

### 4.7.1 Hot Spots

Hot spots usually show as a bright region near the mechanical or optical axis of the telescope, which should be close to the center of the CCD chip. They may be large or small, sharp or diffuse, irregular or round. If your CCD is offset from the center of the camera body, the hot spot may be off-center in the image; but it probably lies close to the optical axis.

Hot spots result from excess light—non-image light—reaching the center of the sensor. Vignetting, which is caused by too little light reaching the edges of the sensor, looks a lot like a hot spot, but their causes are quite different.

If you were to look into the back end of an ideal optical system, you would see nothing but the objective. Everything else would be black. If you see hot spots in an image, it is very likely that when you look through the optics, you will see other sources of light. The best test for hot spots is to remove the camera and place your eye where the sensor normally goes—at the focal plane. As you look up the light path, any light not coming from the objective may be causing the problem.

In a system with a hot spot, light may reach the sensor by a variety of other routes. Once you have identified its source, you can probably eliminate it. Below are some likely hot-spot sources.

**Cylinder Focusing.** To check for cylinder focusing, place whatever adapter tubes you use with your camera in the eyepiece holder, but remove the camera body. With a deep-sky imaging system, the adapter may be a simple 1 $\frac{1}{4}$ -inch-to-T adapter; with eyepiece projection, the adapter may be a system of tubes and a projection eyepiece. In either case, set up the optical system as you would normally use it. Point the telescope at the bright daytime sky or at a bright, uniformly illuminated surface.

In most astronomical CCD cameras, the CCD is slightly behind the front flange of the camera. If possible, place your eye about the same distance behind the adapter tubes. You will see the objective filled with light; all else should be dark. If the shiny metal interior of the focuser tube lights up when you place your eye at the normal location of the CCD—bingo!—that's the cause of the hot spot. If it is difficult to place your eye at the focal plane, place your head behind the open tube and look in. Move your head from side to side and up and down, looking for reflections from the interior surfaces of the tube. These reflections direct light toward the CCD.

The cures for hot spots are baffles and black paint. Baffles are thin rings of metal shim stock, thin plastic painted black, or blackened cardboard lining the inside surfaces of the tubes. The length and size of the adapter tube determine how high the baffles should stand, and how many you will need. You can cut baffles with sharp scissors, a razor knife, or turn them on a lathe to fit snugly into the tube.

## Chapter 4: Imaging Tools

Slide the baffles into the tube and secure them with a dab of acetone-based solvent glue (such as Duco Cement). Paint all surfaces flat black. An inexpensive option to enhance ordinary black paint is to mix fine sawdust into Rustoleum flat black. Dab on this concoction with a cotton swab. It will dry to a dull, rough surface.

An alternative is to line the adaptor tube with flock paper (which has a velvet-like surface) or with black velvet. The nap of the flock paper and velvet acts like thousands of tiny baffles, breaking up and absorbing the reflections. Velvet linings are especially effective for short lengths of tubing; baffles are more effective for long ones.

**Internal Lens Reflections.** The lenses in a focal reducer, focal extender, or projection eyepiece sometimes conspire to focus an image of the objective near the focal plane. Place your eye at the focus, but this time point the telescope at a dark target silhouetted against bright sky. The chimney on a nearby house usually works well for this purpose.

With your eye at the focus, the objective should appear fairly dark because you see it filled with light from the dark target. Move your eye around the focal plane; if the lens fills with light at the center of the field and becomes partially illuminated away from the axis, reflections are contributing to the hot spot. With focal reducers and extenders, you can switch to a different lens, change the spacing so the reflected light forms a larger, dimmer hot spot, or stop using the lens. With eyepiece projection systems, switch to another eyepiece.

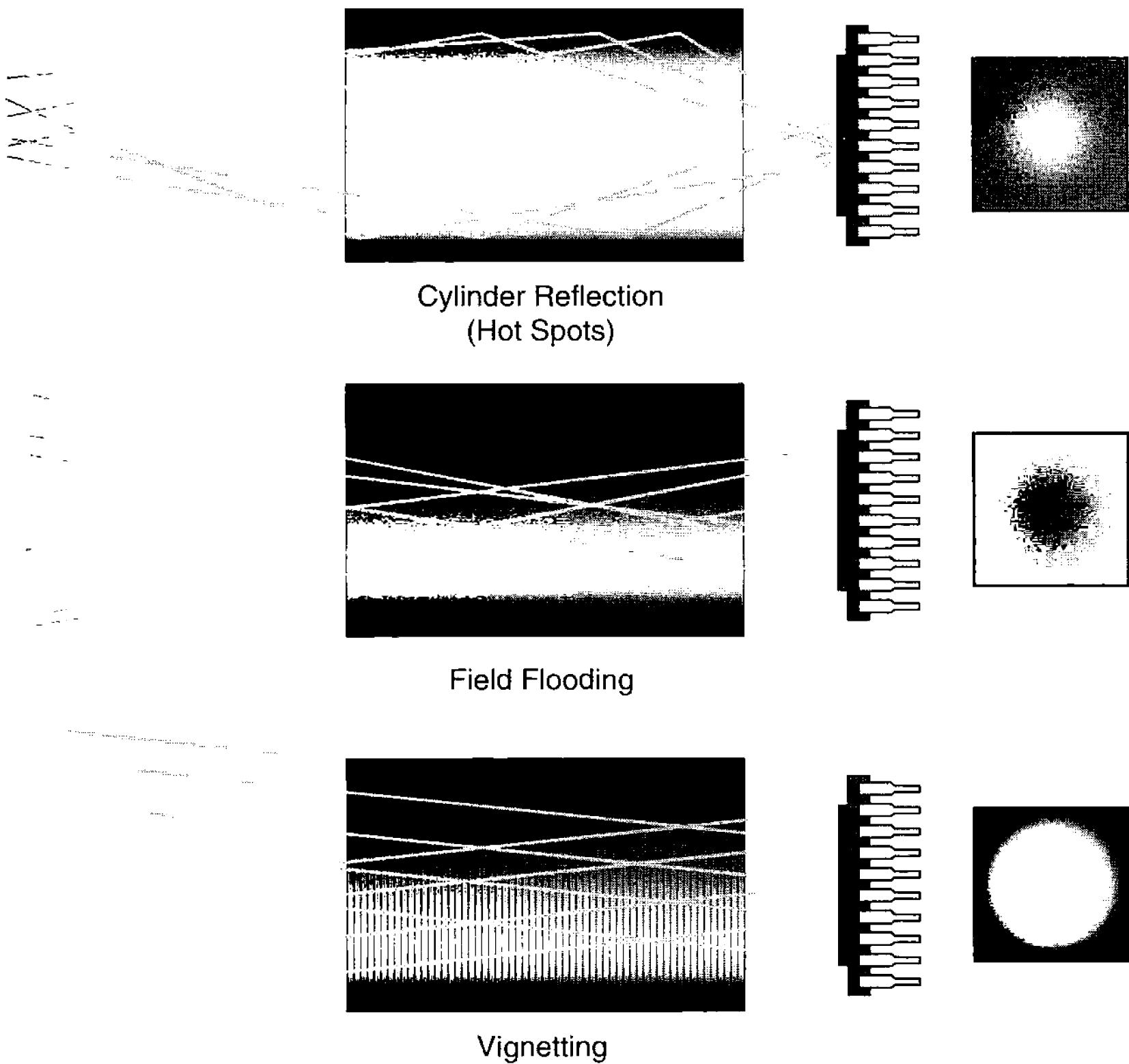
**CCD and Window Reflections.** Your CCD reflects a significant amount of the light that strikes it; this light does not just disappear, but instead travels to the window of the camera where some of it is reflected again, and returns to the CCD. Another reflection comes from the glass window mounted on the CCD itself. It is rare for either reflection to cause serious problems, but when you are troubleshooting hot spots, you should keep it in mind lest you mistake it for a more serious problem. The easiest way to observe these reflections is to point the telescope at a bright star and make an integration of about 10 seconds. Upon processing the image, you should be able to find a large, faint, out-of-focus outline of the telescope objective.

A much smaller ghost image results from reflections between the CCD and the window covering it. The distance is shorter, so the spot will be brighter and smaller. This spot may be hidden by the glare of any star bright enough to make the reflection visible.

### 4.7.2 Field Flooding

When you carry out the hot-spot check, you may see sources of light that are continuously visible; these cause field flooding. In Newtonians, for example, a significant amount of light may reach the focus from the inside wall of the telescope tube opposite the focuser and from the open space around the mirror at the bottom end of the tube. Plug these leaks—field flooding increases noise, reduces contrast, whittles away at the limiting magnitude, and introduces systematic errors in flat-

## Section 4.7: Recognizing and Correcting Equipment Problems



**Figure 4.8** Hot spots occur when stray light reflects from shiny interior surfaces of focus tubes to the center of the detector. Field flooding arises from stray light illuminating all or part of the detector. Vignetting occurs when focus tubes or filter holders clip the edge of the cone of light converging to focus.

fielding.

A ring-shaped baffle inside the bottom of the tube will stop light from coming around the mirror. You can also add baffles cut from thin cardboard or blackened brass shim stock to the front ends of tubes near the camera. When the opening in a baffle is between about 0.1 and 0.4 inches smaller than the inside diameter of the tube, light does not reach the inside surfaces. Install baffles temporarily with tape to check that you are not introducing any vignetting. Once you know they block reflections without vignetting, paint the baffles flat black, and attach them securely.

To stop field flooding from the upper end of the tube, extend it with a “super-baffle” twice the mirror diameter in length. This reduces the amount of stray light falling on the tube walls and diagonal and scattering into the CCD. The super-baffle should be larger in diameter than the rest of the tube, painted flat black on the

## Chapter 4: Imaging Tools

inside, and lined with internal baffles of its own.

While it may seem like a hassle to eliminate field flooding and hot spots in your telescope, you cannot exploit the full power of your CCD as long as stray light reaches it. Proper calibration of images requires that the amount of light falling on each pixel of the CCD be proportional to the amount of light coming from the corresponding place in the sky.

Hot spots and field flooding disturb the linearity of the digital image: the anomalous light means that the amount of light reaching a pixel is not proportional to the intensity of the celestial source. When this happens, the flat-fielding magic does not work, and the observer wonders how he failed.

### 4.7.3 Vignetting

Vignetting results when some of the light from the objective cannot reach the CCD. The light loss is usually at the edges of the image. It can result from a too-small diagonal mirror, focus tube, focal reducer or focal extender, or from misalignment. All of these cut off light from the outer parts of the objective.

**Diagnostic 1: View the Detector in the Objective.** In this simple test, you view the reflection of the CCD in the telescope objective. Move your head back and forth until you have seen the reflection of the CCD in every part of the objective. You should always see the entire CCD reflected in the primary mirror. If you cannot see the whole CCD, light is getting lost. You will also be able to see what part of the telescope—be it an undersized diagonal mirror or the base of the focuser—is causing the vignetting.

The advantage of this test is that the CCD is in place when you conduct it; the disadvantage is that the CCD appears small and distant. Since the image of the CCD appears at infinity, you can inspect it with a finder telescope or binoculars. Once you have identified the source of vignetting, you should fix it.

**Diagnostic 2: Look for Blockage.** Another way to find vignetting is just to look for clipping. Set up the optics for testing for internal reflections; that is, complete but with no camera. Place your eyeball exactly where the CCD would normally go and look at the objective. You should be able to see it in its entirety. As you move your eye away from the optical axis, you will see the edge of the objective blocked when vignetting is present.

Because CCDs are such small detectors, you might expect vignetting to be rare, but this is not the case. It often results from the simple mistake of using a tall 1½-inch focuser on a telescope with a fast mirror, from an earnest attempt to prevent field flooding, or from an overzealous effort to keep the central obstruction small. Vignetting also occurs with focal reducers and focal extenders when the lens elements are too small to illuminate the whole CCD.

You may find vignetting and field-flooding in the same telescope. In Cassegrain, Schmidt-Cassegrain, and Maksutov-Cassegrain systems especially, the baffles that should block light coming around the edge of the secondary mirror may be too small, resulting in field flooding; and the baffle that protrudes through the

## Section 4.7: Recognizing and Correcting Equipment Problems

primary mirror may also be too small, resulting in vignetting.

The eyeball test is not easy to make because it is difficult to place your eye exactly where the CCD would be. If your eye is too far back, the objective may appear to be vigneted when it is really not. If you have difficulty placing your eye in the focal plane, put a diaphragm with a  $\frac{1}{8}$ -inch hole at the location of the CCD; you can then inspect the objective by peering through the hole.

**Diagnostic 3: The Extrafocal Star Test.** The idea in this test is to take two out-of-focus pictures of a field of reasonably bright stars, one on each side of focus. The stars should be far enough out of focus that the objective appears between 20 and 30 pixels across. With an integration time of 60 seconds, shoot several different fields so that you have star images in every part of the images.

If there is no vignetting, the star images will be complete donut images of the objective over the entire field of view in both images, with the diagonal mirror and spider vanes showing in Newtonians, the secondary showing in Cassegrains and SCTs, and complete unobstructed disks showing in refractors.

In severe cases of vignetting, half of the light from the objective may be clipped off. When vignetting is slight, it may be difficult to tell whether the image is complete. Newtonians built for visual observing are primary offenders when it comes to vignetting. Tall focusers, undersized diagonals, and misalignment are the main problems. Eliminating bad vignetting may require significant reworking of your telescope—think of it as “customizing” the instrument for CCD imaging and it won’t hurt so much. With Cassegrains and SCTs, focal reducers often create severe vignetting—but they are compromises anyway. The best solution is to test several different focal reducers in hopes of finding one with better vignetting characteristics.

Unless vignetting is severe, though, there’s no need to despair. Although losing 5 to 10 percent of the incident light at the edge of the field is hardly desirable, with proper flat-fielding, this loss can be calibrated out.

### 4.7.4 Dust Donuts

After a long night of imaging, you get up the next morning, turn on the computer, and there’s an ugly dark donut in half the pictures. Relax—you’re in good company. If you inspect the Voyager images of Jupiter, Saturn, Uranus, and Neptune, you’ll see lots of dust donuts, shadows of dust particles in the optical system. The advantage you have over the Voyager Imaging Team is that you can clean your camera. They couldn’t!

Dust donuts are shadows. It may seem counterintuitive that a small, round speck of dust will cast a ring-shaped shadow, but in Newtonians and Cassegrainians, where the light comes from a source with a central obstruction, that’s what happens. One way to imagine it is to think of a dust particle as an anti-pinhole camera, with the anti-hole giving you an anti-image of the objective. In refractors and unobstructed systems, dust donuts have no holes.

Dust donuts are sneaky. A speck that is only 100 microns across (4 thou-

## Chapter 4: Imaging Tools

sandths of an inch) doesn't block much light when it is on the window 12 mm ahead of the CCD. Assuming an  $f/10$  optical system, the light blocked by the speck spreads to 1,200 microns diameter at the CCD, casting a shadow that is 99.4% as bright as the surrounding sky. The sneaky part is that you don't see this exceedingly pale shadow when you shoot the images. Only after processing and contrast stretching does that minuscule 0.6% drop in light stand out as a dark donut. Fortunately, they are easy to diagnose because the diameter of a dust donut is directly proportional to the distance,  $D$ , between the dust speck and the CCD:

$$D = Pdf \quad (\text{Equ. 4.9})$$

where  $P$  is the width of the dust donut in pixels,  $d$  is the width of a single pixel in the units you used for  $D$ , and  $f$  is the focal ratio of the optical system.

Consider the following example: You have taken a night's worth of lunar images that show dust donuts, and you know that you need to clean something to get rid of them—but what should you clean? One particularly clear donut is 13 pixels in width by 15 lines high. The images were taken with a Cookbook 245 in 252-wide mode using eyepiece projection with an effective focal ratio of  $f/44.4$ . Since the images were taken in 252-wide mode, the pixels are 0.0255 mm wide by 0.01975 mm high. You compute their location:

$$D_{\text{width}} = 13 \times 0.0255 \times 44.4 = 14.7 \text{ mm} \quad (\text{Equ. 4.10})$$

$$D_{\text{height}} = 15 \times 0.01975 \times 44.4 = 13.2 \text{ mm}. \quad (\text{Equ. 4.11})$$

Given the difficulty of estimating the size of the dust donuts, the agreement is pretty good. The estimates place the dust roughly 14 mm ahead of the CCD chip, which pretty clearly puts the offending speck on the window of the camera. With eyepiece projection, small shadows are caused by dust on the cover glass of the CCD itself.

In deep-sky images taken at  $f/5$ , dust donuts on the camera window cast such dilute shadows that they are invisible. However, dust on the front of the CCD cover glass is a candidate for causing problems. The glass cover on a typical CCD is 1 mm thick, located about 1 mm ahead of the chip, and has an index of refraction of 1.53. Refraction decreases the effective air thickness of the glass to 0.65 mm, so the total distance from the front of the cover glass to the CCD is 1.65 mm. Solving the distance equation for the diameter of the shadow:

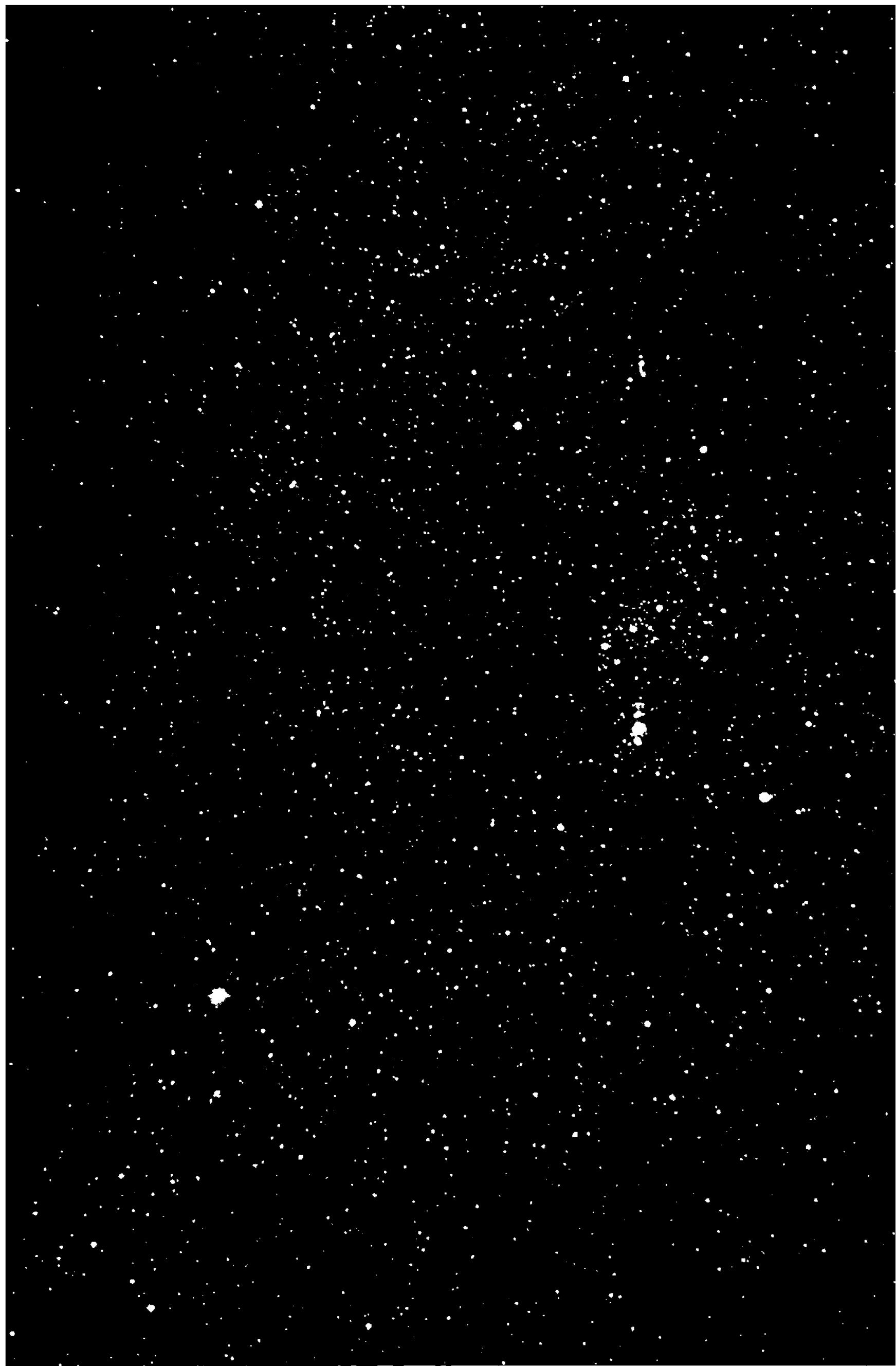
$$P = \frac{D}{df}, \quad (\text{Equ. 4.12})$$

where again  $P$  is the width of the dust donut in pixels,  $D$  is the distance from the focal plane,  $d$  is the width of a single pixel,  $f$ , the focal ratio of the optical system.

$$P = \frac{1.65}{0.0255 \times 5} = 13 \text{ pixels}. \quad (\text{Equ. 4.13})$$

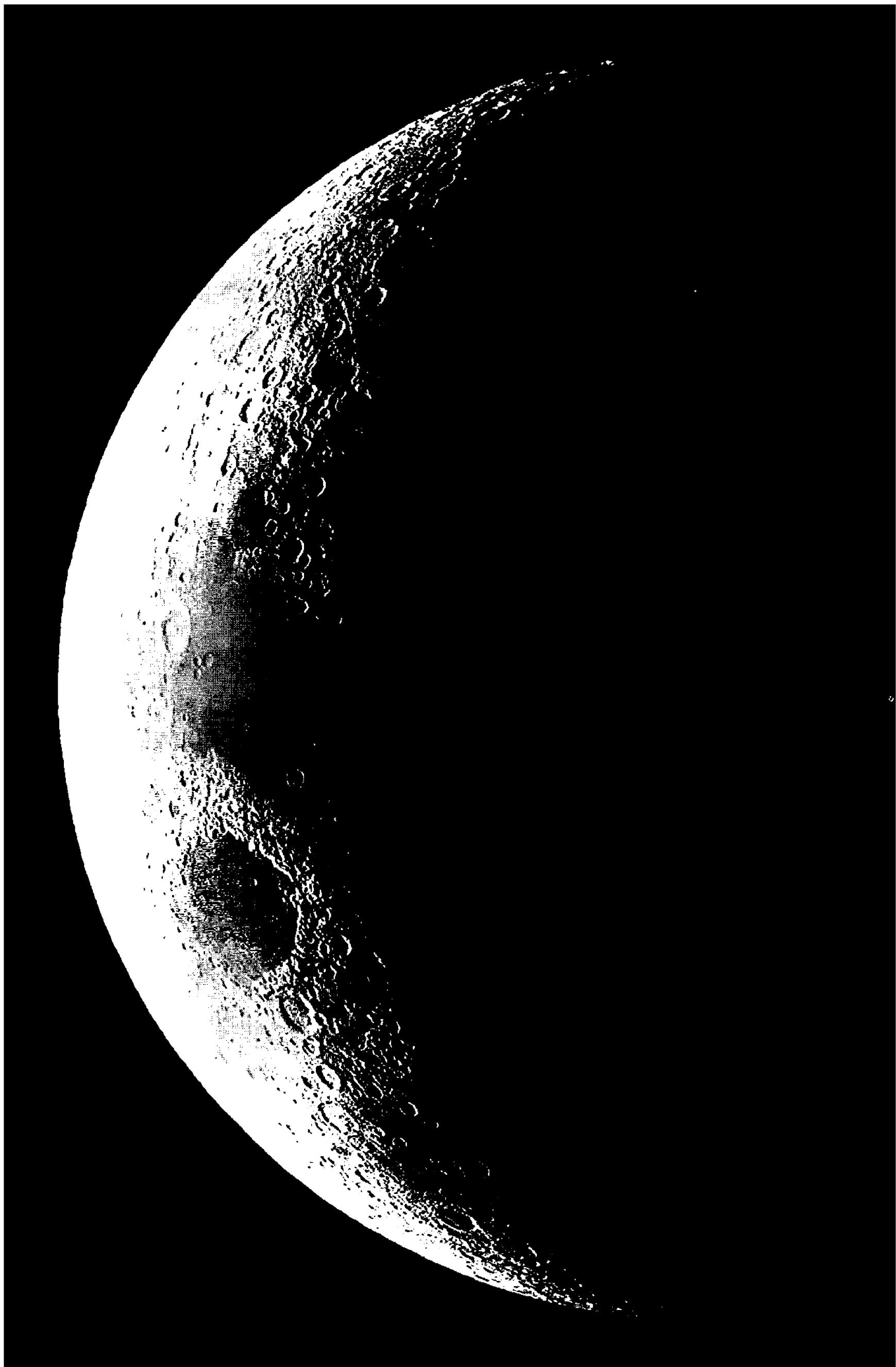
In this example, an image taken with an  $f/5$  optical system using a Cookbook 245

## Section 4.7: Recognizing and Correcting Equipment Problems



**Figure 4.9** Select imaging tools to produce the results you want. To obtain a wide field of view, use a short focal length and/or a large detector. Today's digital SLRs combine quality short-focus lenses with large CCD or CMOS sensors, and they provide constellation-spanning fields of view.

## Chapter 4: Imaging Tools



**Figure 4.10** The right imaging tools will give you the results you want. To hold the Moon comfortably, the focal length of the optical system should be 100 times the size of the detector. This focal length comfortably accommodates Messier objects and most of the better-known NGC and IC objects.

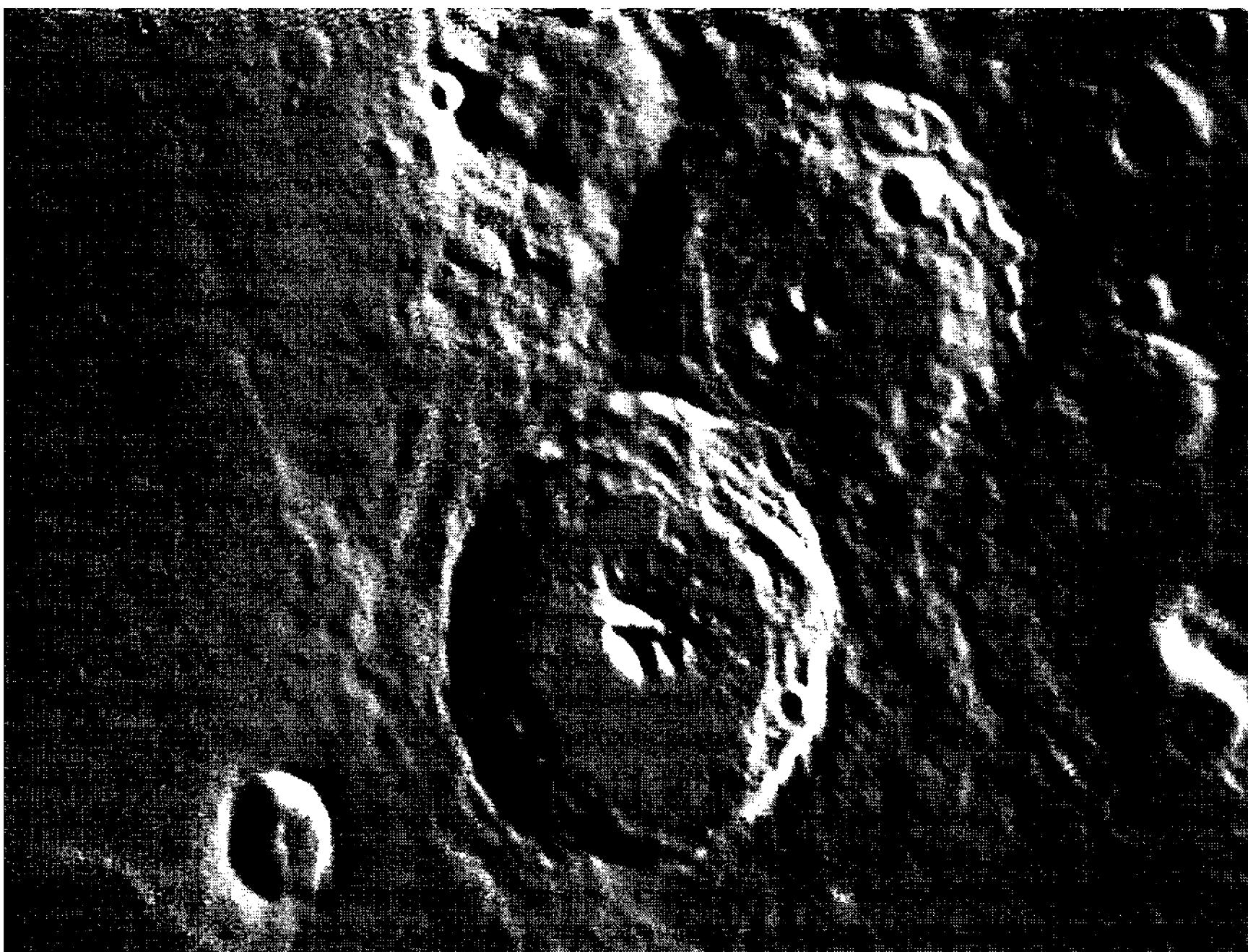


Figure 4.11 Webcams and small-chip astronomical CCDs are great for capturing diffraction-limited detail on the planets and Moon. For such objects, a field of view no more than a few arcminutes wide is all that's needed. The key to successful imaging is to match the tools you use to the imaging task.

in its 252-wide mode (i.e., 25.5 micron pixels), dust on the CCD cover glass will show up as dust donuts 13 pixels wide. When you are confronted with dust donuts, measure their size, calculate their location, and then clean the appropriate surface with alcohol on a swab.

## 4.8 Reaping the Benefits

Shooting digital images is easy, but shooting *great* images takes effort. Great images reflect not only careful attention to hardware and techniques, but also identification of and devotion to specific imaging goals.

Begin by determining your imaging goals. On a clean sheet of paper, write down what you expect to accomplish with your digital images. Reduce your goals to basic facts and figures—fields of view, limiting magnitudes, star-image size—and then select equipment that can and will meet those goals. Every minute that you invest in learning what equipment can meet your goals will be time well spent.

Be prepared to discover that you cannot attain your initial set of goals. It may be necessary to tailor your choice of sensor to an existing optical system—or to tailor the optics to a specific sensor system. You must decide whether field-of-view trumps resolution, or whether resolution trumps field-of-view. This depends

## Chapter 4: Imaging Tools

on your goals and the celestial bodies that you are imaging.

The three F's of imaging—finding, focusing, and following—are prerequisites that your equipment must satisfy for you to achieve success. Select equipment that will make it easy and efficient to locate, sharply focus, and accurately track the objects that you wish to image. It does not much matter how you accomplish these goals, it only matters that you *do* accomplish them.

Once you have selected and installed your imaging equipment, it is still important to remain goal-oriented. It is easy and tempting to expect “the right equipment” to solve all of your imaging problems. If you see hot spots or dark corners in your images, the equipment is trying to tell you that something is not right. This does not mean that the equipment is bad, defective, or broken; it may simply be that you have put the components together incorrectly. Do not blame equipment for problems, but rather look into the matter, identify the source of the problem, and with the aid of the insight gained, cure it. Your reward will be better images.

Finally, learn all you can about your equipment. In digital imaging, your camera, your telescope, the weather, and astronomical objects interact with one another. The more you learn about esoteric subjects such as the spectra of galaxies and the wavelength sensitivity of your camera’s sensor, the more things will make sense to you, and the more focused and better directed your activities will become. Setting broad goals for your imaging and paying attention to the nitpicky details pays off handsomely in results that make you feel proud.

---

# 5 Imaging Techniques

---

In digital imaging, good equipment is only half of the story. The other half of is employing good *technique* in using that equipment. In this chapter, we use the word *technique* in two senses: its general meaning, “the manner in which technical details are treated;” and its specific meaning, “a body of technical methods (as in a craft or scientific research).”

In astronomical imaging, good technique embraces:

- accurate polar alignment,
- good guiding,
- critical focus,
- correct exposure, and
- making calibration frames.

Knowing how to learn from experience is the foundation of all good imaging no matter what technology you are using; it is the essence of developing good *technique*. The items on the list are *techniques*, specific methods that you can apply to image making. These occupy the first half of the chapter.

In the second half, we review techniques for deep-sky imaging and lunar/planetary imaging. Deep-sky images by amateurs now rival the best work professional astronomers could produce a few decades ago. The famous Palomar Observatory Sky Survey carried out with the 48-inch Oschin Schmidt camera using what was then the world’s most advanced photographic emulsion is now routinely bettered by amateurs with 8- and 10-inch telescopes. Also, a few years ago, CCD users stunned the world of amateur astronomy by proving that CCDs (and now webcams) can record more detail than skilled visual planetary observers. We’ll help you see what it takes to push your results into the realm of high-resolution lunar and planetary imaging.

## 5.1 Accurate Polar Alignment

Equatorial mountings should be aligned so that the polar axis of the telescope is parallel with the rotation axis of the Earth. Alignment errors larger than a fraction of a degree quickly reveal themselves in star images that drift north or south even during exposures of one or two minutes.

## Chapter 5: Imaging Techniques

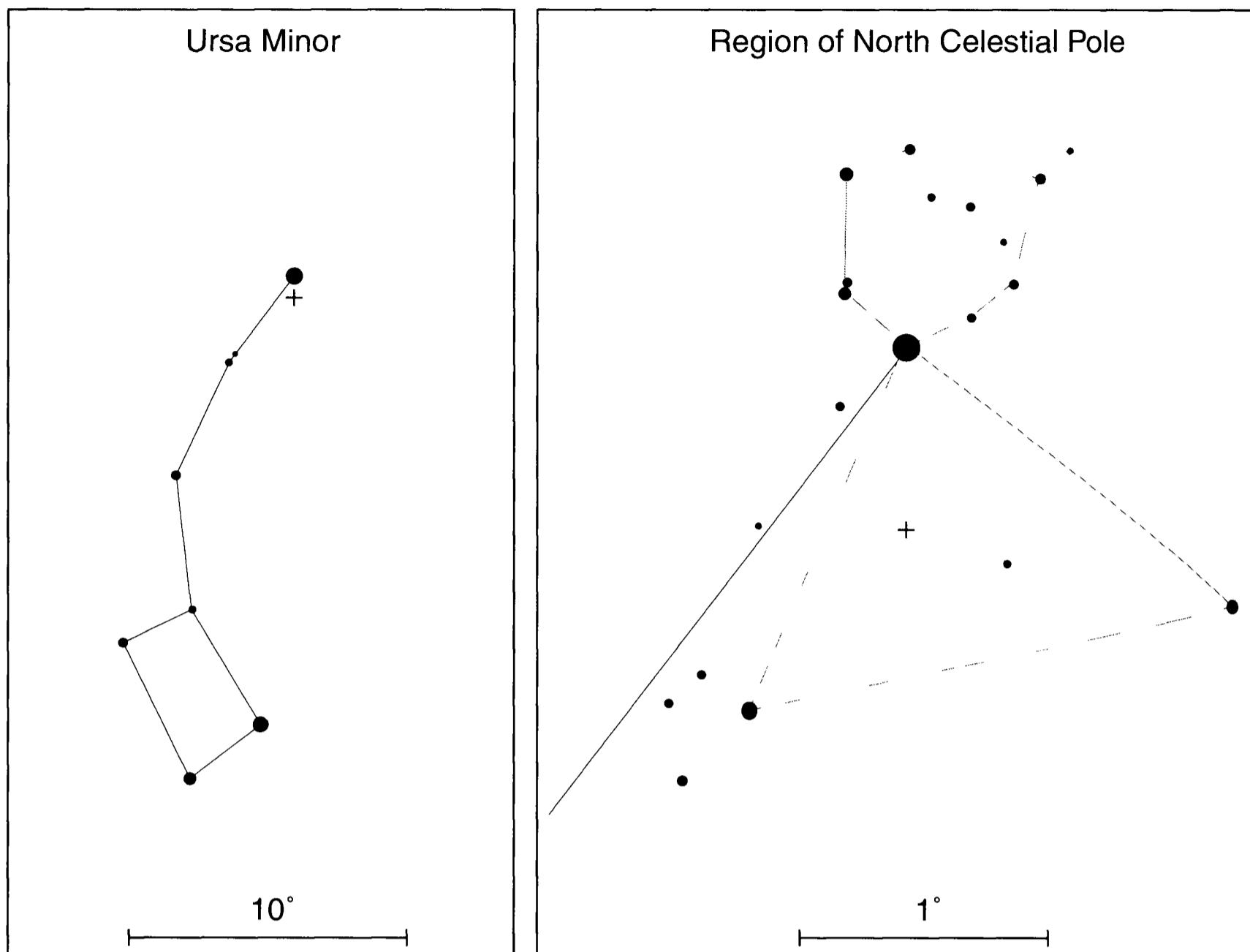


Figure 5.1 Portable telescope mountings often include a polar-axis telescope. The north celestial pole is easy to locate some  $0.73^\circ$  from second-magnitude Polaris, opposite a “diamond ring” asterism composed of 8<sup>th</sup> and 9<sup>th</sup> magnitude stars. The small cross in each chart marks the location of the celestial pole.

If you have a portable telescope with a built-in polar axis alignment telescope, you can simply point that at the current location of the pole, and alignment is taken care of. For short exposures or for short-focus optics, a polar-alignment telescope makes this task quick and easy.

If, on the other hand, you have a telescope that is permanently mounted in a backyard observatory, you can adjust the polar alignment to within a gnat’s whisker of perfection and then forget about it—only atmospheric refraction will disturb your tracking accuracy.

Observers with portable rigs that lack polar-alignment telescopes need to develop an efficient technique for setting up and checking alignment. Here is a procedure (for Northern Hemisphere observers) that works well with any mounting with setting circles:

1. Set the telescope so that the declination circle reads exactly  $+90^\circ$ .
2. Adjust the polar alignment to center Polaris in the field of view.
3. While looking through a low-power eyepiece, rotate the telescope in right ascension. All objects in the field of view will appear to circle around some point. That point may be outside the field of view.

## Section 5.1: Accurate Polar Alignment

4. Move the telescope in declination only until that center of rotation is centered in the field of view. (Note: if the declination and polar axes of your telescope are not perpendicular, you will be able to center the rotation in declination but not in right ascension.)
5. Adjust the declination setting circle so that it reads exactly  $90^\circ$ . This calibrates the declination circle. Lock the setting circle adjustment so that it cannot change.
6. Point the telescope at a bright star roughly overhead. Depending on the date and time, convenient stars are:
  - $0^{\text{h}} 57^{\text{m}}$  — Cih ( $\gamma$  Cas)
  - $3^{\text{h}} 24^{\text{m}}$  — Mirfak ( $\alpha$  Per)
  - $5^{\text{h}} 17^{\text{m}}$  — Capella ( $\alpha$  Aur)
  - $8^{\text{h}} 30^{\text{m}}$  — Muscida ( $\sigma$  UMa)
  - $11^{\text{h}} 04^{\text{m}}$  — Dubhe ( $\alpha$  UMa)
  - $13^{\text{h}} 24^{\text{m}}$  — Mizar ( $\zeta$  UMa)
  - $17^{\text{h}} 56^{\text{m}}$  — Eltanin ( $\gamma$  Dra)
  - $21^{\text{h}} 19^{\text{m}}$  — Alderamin ( $\alpha$  Cep).
7. Set the polar circle to read the right ascension of the star. This calibrates that circle. Lock the circle adjustment so that it cannot change.
8. Set the telescope to the coordinates of Polaris ( $2^{\text{h}} 32^{\text{m}} +89^\circ.26$ ).
9. Adjust the location of the polar axis to center Polaris in the field of view.

If the mount's setting circles are accurate, the polar axis is now aligned within perhaps  $0.1^\circ$ . Next, you must refine the alignment to eliminate trailing during long integrations. Use your CCD camera in its focus mode so that the image scale is large. Be sure that you know the orientation of images on the computer screen. Now use the drift method to fine-tune the polar alignment:

1. Center the telescope on a star on the celestial equator near the meridian.
2. Observe the motion of the star for several minutes. If the CCD software reads out the position to subpixel accuracy, so much the better.
3. **If the star drifts south, move the north end of the polar axis west.  
If the star drifts north, move the north end of the polar axis east.**  
After several repetitions, you will get a sense of how much correction it takes to change the drift. Quit when you have reduced the drift enough that it is difficult to measure.
4. Center the telescope on a star on the celestial equator rising in the east.
5. Observe the motion of the star for several minutes.
6. **If the star drifts south, raise the north end of the polar axis.  
If the star drifts north, lower the north end of the polar axis.**  
After several repetitions, you will get a sense of how much correction it takes to change the drift. Quit when you have reduced the drift to a level that is difficult to measure.

## Chapter 5: Imaging Techniques

7. Repeat fine-alignment steps 1 through 3 and 4 through 6 until the telescope tracks well enough that the drift is less than one-fourth of a pixel during your longest integration.

This technique calls for small adjustments in the polar axis alignment. Anything you can do to make the adjustments reliable and repeatable—such as attaching protractor scales on the adjustment knobs—will greatly speed the process.

### 5.2 Good Guiding

Guiding long photographic exposures brought CCD cameras to the world of amateur astronomy. When the Santa Barbara Instrument Group (SBIG) introduced their ST-4 camera, it was intended as a guider that would automatically sense tracking errors and restore a guide star to its correct location. It didn't take long before observers realized that a tiny CCD (such as the 165×192-pixel TC-211 in the SpectraSource Lynxx) was capable of imaging faint stars and galaxies. CCD imaging took off, and the rest is history.

There are three basic methods of guiding:

- Off-Axis Guiding
- Auxiliary Telescope Guiding
- Software Virtual Guiding

The first two methods are employed by autoguiders and by observers who guide manually. The third method is unique to astronomical CCD cameras.

In the old days, astrophotographers stared into an eyepiece for hours on end to guide long exposures on film. Today many observers use an autoguider to do the work; but for others, even the simplest autoguider is a budget-buster. For these observers, manual guiding is still an option, and because CCDs and digital cameras do not require hour-long exposure times, this option is relatively easy compared to the old days.

Either guiding process exacts a significant toll in observing time, because after acquiring the object, it is necessary to locate a guide star and to designate it in the guiding camera or center it in the crosshair eyepiece. Whether you guide manually or automatically, you will need to develop efficient techniques for the task.

#### 5.2.1 Off-Axis Guiding

CCDs were introduced to amateur astronomy as autoguiders to relieve the tedium of guiding long exposures. With a second CCD to detect the motion of a guide star and automatically command the slow-motion motors on the telescope, an observer can make long exposures without suffering from a stiff neck for days afterward.

Many self-guiding CCD cameras are off-axis guiders; that is, a second CCD in the camera head picks up a star image that is off the optical axis. This guiding method eliminates potential problems with shifting optics and telescope tube flexure.

The operating technique in off-axis guiding is to locate the target object in the imaging CCD, and then find a guide star. Because the guider CCD is more sensitive than the eye, often there are several stars bright enough to serve as the guide star. After selecting one and starting the autoguider, the observer can make one or more exposures with the imaging camera.

Manual guiding works much the same way—except instead of a CCD watching the guide star, an eyepiece equipped with illuminated crosshairs receives the off-axis image. The observer’s job is to monitor the position of the star and when it drifts, to press control buttons that recenter the star. Manual guiding is tedious—there is no other word for it. However, it is a skill that can be satisfying to master, especially when the result is small, perfectly round star images.

### **5.2.2 Auxiliary Telescope Guiding**

Using an auxiliary telescope to guide is the principal “real-time” alternative to the off-axis method, and the only practical technique for use with digital cameras. The principal requirement is a second telescope solidly mounted on the imaging telescope. In practice, you place the autoguider head at the focus of the auxiliary telescope and focus the image. After you center your target object in the imaging telescope, locate a suitable guide star with the auxiliary telescope, designate the star to the autoguider software, allow the guider to self-calibrate, and then relax while the autoguider keeps the star centered within a fraction of a pixel.

For guiding manually, first acquire the target object in the imaging telescope, place a high-power eyepiece with an illuminated cross-hair reticle at the focus of the guide telescope, align the latter on a star bright enough to see easily, and then, after starting the exposure, use the telescope’s slow-motion controls to keep the star centered precisely in the reticle. It is easy to take five-minute exposures providing you have a telescope that can track more than 30 seconds without trailing.

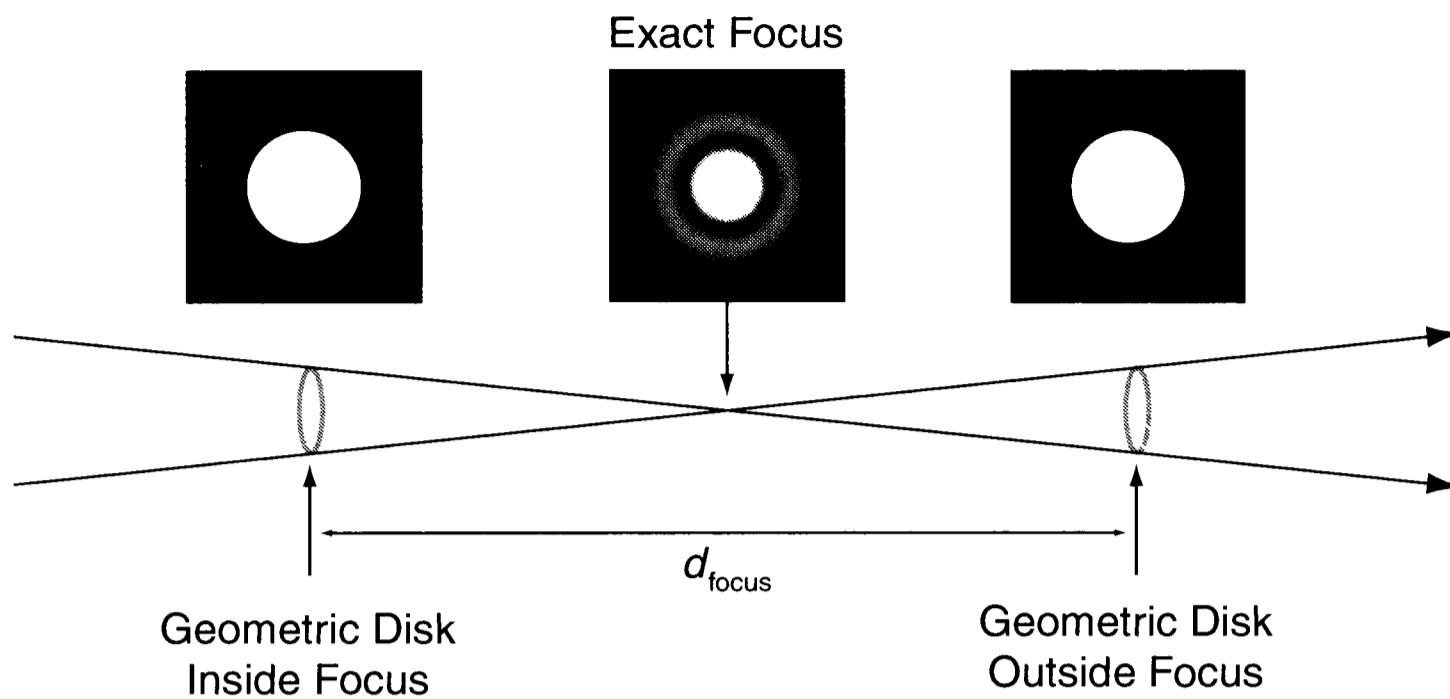
Although it looks a bit odd, the auxiliary telescope technique works especially well when the guiding instrument is larger than the imaging one. The number of potential guide stars is much larger, and of course, using the small telescope for imaging gives you a system that is more forgiving of guiding errors.

### **5.2.3 Software Virtual Guiding**

Few telescope mounts satisfy a “less-than-one-pixel” tracking standard for exposure times of five to ten minutes, but many mounts track adequately during exposures of one or two minutes. With a sinusoidal periodic error, the drive rate is nearly correct for about one-fourth of each rotation of the worm. With a standard 360-tooth drive gear, the worm rotates once in four minutes. Without guiding at all, you will get two slightly trailed 60-second images and two reasonably well-tracked 60-second images during each rotation of the worm. Furthermore, in a 60-second exposure, less-than-perfect polar alignment tracking errors are usually not large enough to matter.

CCD cameras can exploit short exposures using the track-and-stack tech-

## Chapter 5: Imaging Techniques



**Figure 5.2** At best focus, star images are maximally concentrated and compact, limited in size by diffraction. On either side of focus, the image expands. Depth of focus is defined as the distance over which a star image remains sufficiently small that it is effectively still in perfect focus.

niques; instead of one ten-minute exposure, program the camera software to make twenty one-minute exposures. Typically, somewhere between half and two-thirds of the exposures will have acceptable tracking, and those that don't can be discarded. The good images are registered in software and stacked to form an image with the signal-to-noise as good as a single long exposure.

Although the track-and-stack method requires lots of space on the imaging computer's hard disk, if you cannot afford a high-quality mounting and an autoguider, it is a practical and cost-effective way to accumulate enough photons to make an outstanding image.

### 5.3 Critical Focus

To all appearances, focusing an astronomical CCD camera or digital camera ought to be simple, but often it is not. Observers report spending ten to fifteen minutes trying to obtain accurate focus, and then complain that they are not sure that they have succeeded.

The first step in getting sharp images is to be sure that the telescope optics are accurately aligned and produce good images. SCTs are prone to getting bounced out of alignment on the bumpy back roads that lead to good observing sites. Newtonians are prone to slip out of alignment at the slightest provocation. Make sure the optics are good, and check the alignment of your telescope with collimation tools or a laser before every imaging session. Only well-made refractors seem to hold their alignment year after year.

In astronomical CCD cameras, the focus problem seems to be that the blocky star images on a computer screen do not look at all like the familiar pinpoints in an eyepiece, so the observer cannot decide which blocky blob is the "best" blocky blob. With digital cameras, the image in the reflex viewfinder is too small to judge

whether stars are in sharp focus, so focusing must be done by taking an image, assessing its sharpness, adjusting the focus knob, and taking another image.

This process sounds easy, but downloading and displaying the last focus image takes some length of time. Since telescope shakes and atmospheric turbulence change star images somewhat randomly, a slow readout can make judging and attaining critical focus frustrating even for a skilled observer.

Another focusing bugaboo occurs with telescopes that change focus when they are pointed at different parts of the sky. SCTs and Newtonians are prone to shift focus slightly when pointed in a new direction. To use a Newtonian telescope for imaging, it helps to stabilize the primary and secondary mirrors by applying aquarium cement or bathtub caulking at three points around the edge of each mirror. With SCTs, special mirror lock-down kits allow you to steady the optics.

### **5.3.1 The Focuser**

Focusers that stick, bind, or suddenly jump from one focus position to another simply will not do for digital imaging. Rack-and-pinion focusers are terrible in this regard; torque applied to the focus knob seldom results in predictable linear motion. Another offender is the SCT with a sloppy mirror-focus system; not only are focus changes unpredictable, but star images move every time the observer touches the focus knob. An add-on accessory that helps you deal with these problems is a focus counter that displays the exact focus position—but a solid, motorized focuser is a blessing that every serious imager should consider.

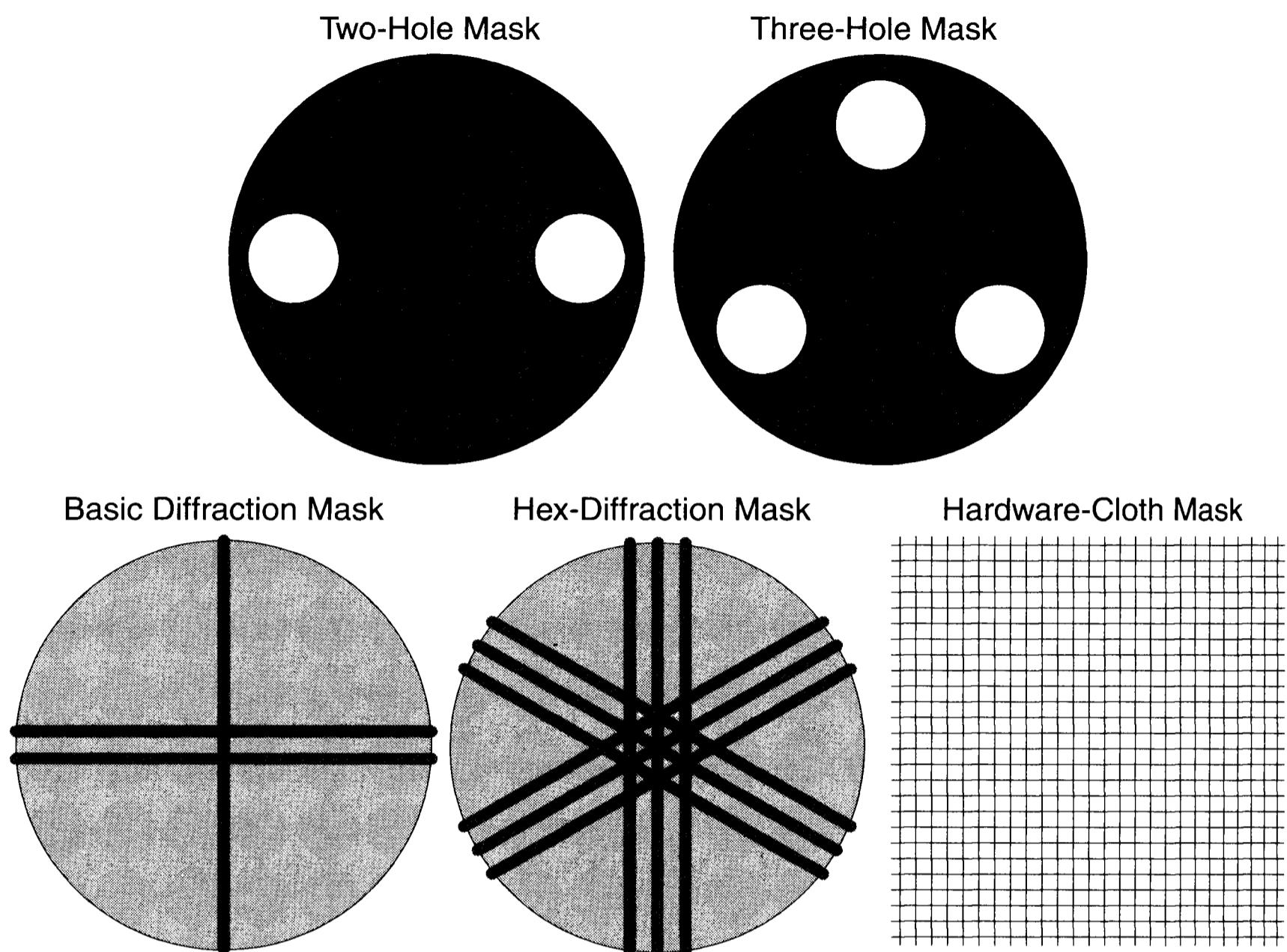
### **5.3.2 Focus Techniques**

Even with steady air, a sturdy mounting, a fast-reading CCD camera, a rock-solid focuser, and stable telescope optics of high quality, the greatest difficulty in focusing is knowing how to recognize a properly focused image. Here are some methods in wide use among skilled observers.

**The “Smallest Star Image” Technique.** Some observers are adept at judging when the turbulent star-image blob is smallest. To focus using this method, select a star that is bright enough to produce an easily observable disk when it is out of focus, but not so bright that it saturates the sensor when it is near focus. Move slowly through focus in about ten steps, then back again until you find positions on either side of focus that have the same size disk. Find two new positions that have smaller out-of-focus disks, and finally find two positions where the star disk is just barely enlarged. Best focus is midway between the final two just-barely-out-of-focus positions.

**The “Peak Pixel Value” Technique.** If you have an astronomical CCD camera that reports the peak pixel value in an image or in a focus box, find a moderately bright star and observe the peak value as you change the focus position. The peak will fluctuate randomly; however, if you mentally average three to five images, the numbers will rise and then begin to fall again. Pass through focus several times until the peak pixel values are highest.

## Chapter 5: Imaging Techniques



**Figure 5.3** Perfect focus often proves to be elusive. A variety of focus aids make it easier to determine the moment when the star image is smallest. With two-hole and three-hole masks, the star is deformed when it is slightly out of focus; with diffraction masks, you judge when an extended pattern is sharpest.

**The “Number of Faint Stars” Technique.** The image of a sharply focused star concentrates its light in the smallest possible area. When stars are even slightly out of focus, their light is spread over a larger area, so fewer faint background stars will appear in focus-mode images. Set the exposure time to several seconds, focus approximately, then select a field with no bright stars in it. Slowly change the focus. As you move away from best focus, faint stars will drop out of the image; and as you approach best focus, more faint stars will appear. Proceed in very small steps—this technique is quite sensitive. Since the faint star images are literally “buried in the noise,” their presence is subtle but easy enough to recognize. Even when you cannot see any change in the size of the star images, the number of faint stars increases sharply at best focus.

**The “Longest Blooming Trail” Technique.** The blooming trails that are so annoying in antiblooming CCD cameras can be harnessed as a focusing aid. Focus approximately, then center on a star that is bright enough to cause a readily visible blooming trail on the computer screen. When the star image is in the best focus, the concentrated starlight causes pixels to bloom; the more concentrated the light, the better the focus. An additional cue is that the better the focus, the thinner the blooming trail. Focus back and forth until the blooming trail is as long and thin as you can get it.

**The “Diffraction-Spike” Technique.** When the seeing is poor, star image

blobs look the same over a considerable focus range. At such times, the contrast and visibility of diffraction spikes protruding from a star image may be easier to judge. You can generate handy diffraction patterns by placing one or many straight obstructions, such as wooden dowels over the front of the telescope tube. The thinner and closer the dowels, the wider (and weaker) the diffraction pattern. Two  $\frac{1}{4}$ -inch dowels separated by two inches produce a useful spike, and an array of six dowels spaced an inch apart generates a big, bright spike. A dramatic alternative is to place a piece of hardware cloth (wire mesh with  $\frac{1}{4}$ -inch spacing) over the front of the tube. When the extensive diffraction pattern looks bright and sharp, the camera is in focus.

**“Holes-in-a-Mask” Technique.** When a star is in sharp focus, rays across the entire aperture converge in the star image. If you place a two-hole mask (a piece of cardboard with holes cut in it) over the telescope tube, light from the two openings will converge at best focus. In poor seeing, it is relatively easy to tell when the two images merge into one. The holes in the mask should be about one-fourth the aperture and should be placed at opposite sides of the aperture. As you approach focus, the two images merge into an elongated blob, become round at best focus, become elongated past best focus, and finally split into distinct beams. After you focus, remove the mask.

A variation of this method is the three-hole mask. Cut three holes in a sheet of cardboard; they should be about one-fourth the aperture in diameter and spaced  $120^\circ$  apart. With this mask in place, you will see triangular images except when the telescope is near perfect focus. When you get round star images, remove the mask.

### 5.3.3 Automated Focusing

Focusing can be automated providing you have a good-quality motorized focuser, a CCD with a reasonably fast image readout, and appropriate autofocus software. When it’s time to focus the image, simply identify a star and turn focusing over to the software.

The software usually runs the focuser well inside focus, and then systematically steps outward through focus, monitoring and measuring the size of the defocused image at each position. Plotted on a graph, the size is a “V”-shaped curve, with the smallest star image and best focus at the base of the “V.” The software solves for the best position and then moves the focuser to that point.

## 5.4 Correct Exposure

Astronomical CCD cameras produce acceptable images over an extremely wide range of exposure—but there are limits. In an underexposed image, noise dominates and obscures the target; in an overexposed image, the sensor is near saturation, or pixel values have gotten pegged at the maximum value of the camera’s output format.

During an exposure, photons rain down on the CCD and accumulate. If the

## Chapter 5: Imaging Techniques

**Table 5.1 Surface Brightness of Celestial Objects**

Object	B-Value	Object	B-Value
<b>The Sun</b>		<b>Diffuse Nebulae</b>	
Solar disk	10,000,000	M42, core	0.001
Prominences	100	M42, batwing	$150 \times 10^{-6}$
Inner Corona	50	M42, tendrils	$18 \times 10^{-6}$
Middle Corona	5	M42, outer loop	$6 \times 10^{-6}$
Outer Corona	0.5	Crab Nebula	$35 \times 10^{-6}$
<b>The Moon</b>		Omega Nebula	$15 \times 10^{-6}$
Thin Crescent	10	Rosette Nebula	$5 \times 10^{-6}$
Crescent	20	<b>Planetary Nebulae</b>	
Quarter	40	Ring Nebula	$60 \times 10^{-6}$
Gibbous	80	Owl Nebula	$15 \times 10^{-6}$
Full	200	<b>Galaxies</b>	
Earthshine	0.01	Nuclear Bulge	$40 \times 10^{-6}$
<b>Planets</b>		Inner Arms	$8 \times 10^{-6}$
Mercury	1,000	Outer Arms	$4 \times 10^{-6}$
Venus	2,000	Faint Galaxies	$2 \times 10^{-6}$
Mars	160	<b>Sky Background</b>	
Jupiter	30	Daytime Sky	1,000
Saturn	10	Urban Night Sky	$2,000 \times 10^{-6}$
Uranus	4	Suburban Night Sky	$150 \times 10^{-6}$
Neptune	1	Rural Night Sky	$10 \times 10^{-6}$

source of photons remains at a constant brightness, the accumulated signal is proportional to the integration time. If the pixel value of the sky background is 100 ADUs and the arms of a spiral galaxy are 150 ADUs after a 60-second integration in a calibrated image, then after a 120-second integration they will have doubled to 200 and 300 ADUs, respectively.

The fact that pixel values in an image increase linearly with integration time means that you don't have to make wild guesses to get reasonable integration times. You can calculate the exposure time to reach a given pixel value based on the surface brightness of the object that you are imaging.

To estimate the exposure time,  $e$ , that's needed to produce a given signal, evaluate the following equation:

$$e = \left( \frac{P_{\text{desired}}}{P_{\text{max}}} \right) \left( \frac{N^2}{SB} \right) \text{ [seconds]}, \quad (\text{Equ. 5.1})$$

where  $P_{\text{desired}}$  is the pixel value you want the object to have in the image,  $P_{\text{max}}$  is the maximum pixel value (255 for 8-bit, 4095 for 12-bit, and 65,535 for 16-bit cameras),  $N$  is the focal ratio of the imaging optics,  $S$  is the “ISO speed” of your astronomical CCD or digital camera, and  $B$  is the surface brightness for the object, given in Table 5.1.

*Note: These exposure times should serve as a rough guide only; you will need to refine them to match your CCD or digital camera by taking test integrations.*

## 5.5 Shooting Calibration Frames

Calibration plays an integral role in making top-notch CCD images; but in the excitement of imaging faint nebulae, changing filters, or battling intermittent clouds, it is tempting to skimp on shooting calibration frames. Unfortunately, skimping on calibration steals away the very qualities you want most in your celestial imagery. We discussed the basic idea behind accurate photon counting in Chapter 2, and cover calibration in full detail in Chapter 6. In this section, we focus on the practical details of making high-quality calibration frames.

### 5.5.1 Basic, Standard, and Advanced Calibration

Calibration comes in three “flavors;” basic, standard, and advanced. Each flavor is a method of performing calibration according to a standard technique, or protocol. Select the technique most appropriate for your imaging program and follow it each night that you make images.

**Basic calibration.** This technique meets the needs of beginners and observers who are searching for supernovae, comets, or asteroids. It is simple: before or during an observing session, make between five and sixteen dark frames using the same exposure time you use for your raw images. These dark frames are combined into a master dark frame which is subtracted from the raw images.

**Standard calibration.** The standard calibration technique produces high-quality monochrome or color images suitable for display or publication, or for precise astrometry or photometry. In addition to shooting a set of dark frames with the same exposure as the raw images, you make raw flat frames and flat darks. The total investment of time is ten to fifteen minutes. The darks are combined to make a master dark frame, and the raw flats and flat darks are combined to make a master flat. The master dark frame is subtracted from each raw image, and the result is divided by the master flat frame.

**Advanced Calibration.** Advanced calibration is used to produce high quality images. It is the most flexible protocol, but also the most involved. In addition to making dark frames (with exposure times longer than the raw images), raw flats, and dark flats, you must make bias frames. The bias frames allow you to adjust the master dark frame so that dark current can be removed from an exposure of any length.

## Chapter 5: Imaging Techniques

- **Tip:** *AIP4Win automates all three calibration protocols, and provides tools for user-controlled calibration procedures as well. Newcomers to CCD imaging may wish to use the basic calibration protocol until they feel ready for standard or advanced calibrations.*

### 5.5.2 Making Bias Frames

Bias frames are images with zero exposure time and no light. Making them is simple: cap the telescope, close the shutter, and make the shortest integrations your camera allows. Make and save five or more bias frames. During the calibration, you can combine these either by averaging them or by taking the median value to obtain a master bias frame.

Some CCD cameras automatically correct for changes in the bias level by clocking out extra lines after the CCD image has been read, to obtain an average bias. They then correct the image for any deviation from a fixed bias pixel value, such as 100 ADUs. If your camera has this feature, use the fixed bias value instead of a master bias frame.

- **Tip:** *In AIP4Win's advanced calibration, you have the option of creating a master bias frame or using a fixed bias value.*

Because they have little or no dark current and no photon-induced signal, bias frames are uniform, recording principally the random noise contribution from the sensor's on-chip amplifier. However, because they are so very bland, bias frames reveal noise that might otherwise escape detection. Horizontal or vertical stripes, herringbone patterns, wavy lines, or random bright or dark pixels are signs that noise is creeping into the images.

### 5.5.3 Making Dark Frames

High-quality dark frames play a key role in calibration not only because they remove dark current, but also because they place the zero-point of the numerical values in your image where it should be located, at zero. This is necessary for accurate photometry, precise astrometry, and true-color imaging. Furthermore, flat-fielding works well only with properly dark-subtracted images.

Dark frames have two components: a zero-point bias and a thermal signal. The thermal signal (also called dark current) accumulates at a rate that depends on the temperature of the image sensor and the duration of the integration. The bias should be constant, but the thermal signal grows linearly with integration time.

Making dark frames is easy: you cap the telescope and (if it has one) you close the shutter of your camera. No light should reach the sensor. You then make an exposure and save the resulting image. If you immediately make another integration of the same length and carefully compare the two, you will find that they differ slightly. The small difference is random noise. By making multiple dark frames and averaging them, you reinforce the constant thermal signal and at the same time average away the random variations.

If you are using the basic and standard calibration protocols, make five or

## Section 5.5: Shooting Calibration Frames

more dark frames using the same integration time you are using for your images. If you are making 60-second and 120-second integrations, you should make two sets of dark frames with matching integration times.

For the advanced calibration protocol, you can make one set of dark frames with an integration time that is longer than your longest image integration. For example, if you are making 30-second, 60-second, and 120-second raw images, you can make five or more 300-second dark frames. In the advanced calibration protocol, you also make a set of bias frames, which can be subtracted from the darks to obtain a direct measurement of the thermal signal. This can then be scaled to match the thermal signal that accumulated in the 30-, 60-, and 120-second integrations. Flexibility in exposure times is a strong incentive to adopt the advanced calibration protocol for all of your imaging.

- **Tip:** *AIP4Win supports basic, standard, and advanced calibration techniques, as well as being able to automatically match a dark-frame of one exposure time to an image with a different exposure time—a significant convenience for observers.*

### 5.5.4 Making Flat-Field Frames

Flat-field frames map the relationship between the intensity of light coming from the sky and the response of each photosite on the sensor. To make a flat frame, all you need to do is shoot an image of a featureless subject. If the light is uniformly distributed across the telescope aperture and spread uniformly in angle, the resulting image records not only how much light came through the optics and struck each photosite, but also how each photosite responded to the light falling on it.

Later, in software, the pixel values in the flat field are used to compensate for light lost in the optics and for the variations in the pixel-to-pixel sensitivity of the sensor. The key to making good flats is simple: the flat must be made using exactly the same setup as used for imaging. Do not remove and replace the camera; do not tighten, loosen, clamp, unclamp, or otherwise change anything about optical system. Ideally, do not even change focus. The flat must be an exact record of how light passed through the telescope when you were making images.

Good flat fields are not difficult to shoot, but they must be done correctly or they don't work. The key is to image a uniform low-level source of light that fills half the dynamic range of the sensor in a reasonable integration time, between approximately 2 and 20 seconds. The four basic options are twilight flats, sky flats, dome flats, and light-box flats. Twilight flats are images taken of the twilight sky some 30 to 45 minutes after sunset or before sunrise. Sky flats are made from images of the night sky itself. Dome flats are images taken of a large white screen attached to the inside of the observatory dome. Finally, light-box flats are images of a diffusing screen in a lightweight, internally-lit box placed over the end of the telescope. Of the four options, light-box flats are by far the best bet for amateur astronomers.

**Light-Box Flats.** An illuminated box is the most reliable way for amateurs

## Chapter 5: Imaging Techniques

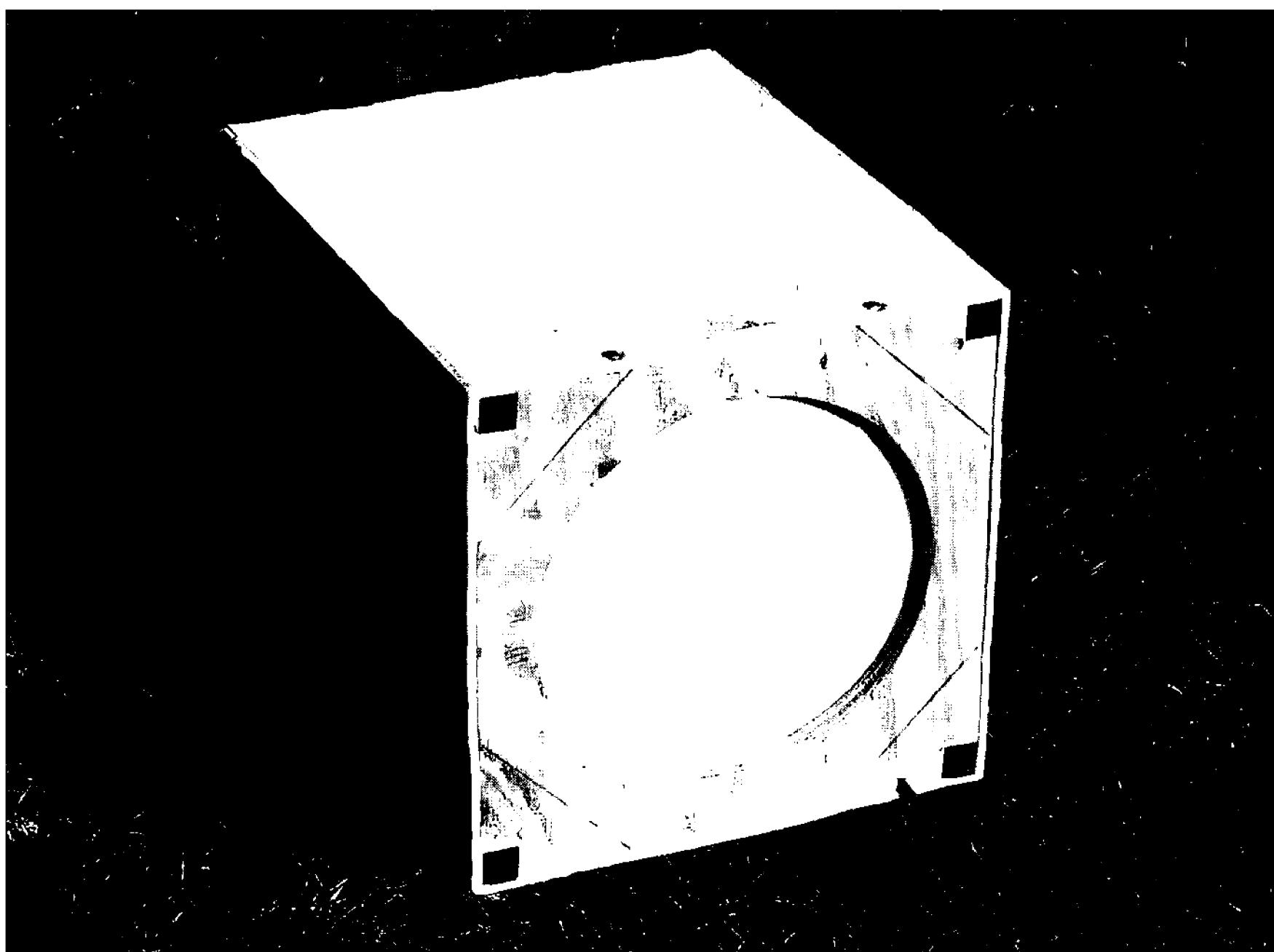


Figure 5.4 This simple light box for making flat-field frames slips over the front end of the telescope. When the internal lamps are on, the telescope “sees” a uniformly illuminated field of view. The circle is made of milk plastic, available from plastics suppliers and firms that sell outdoor advertising signs.

to make flats. It provides a uniform source that can be put on the telescope quickly and enables flats to be made easily with repeatable results.

The key component of a light box is a diffusing sheet of milk plastic that fits right against the front of the telescope. (Milk plastic is pure white material used in internally-lit signs and theater marquees; you can find a local supplier under “Plastic: Rods, Sheets, and Rolls.”) Illuminated from behind, a piece of milk plastic diffuser mimics a uniform section of sky.

In a typical home-built light box, the plastic is mounted directly on the end of the telescope. Over the milk plastic is a cube of foam core board or light plywood painted white inside. In the corners of the cube next to the telescope are four lamps. Baffles prevent them from shining directly on the milk plastic, so their light bounces around the interior of the box. By the time the light reaches the back side of the milk plastic, it is extremely uniform.

For a 6- to 8-inch telescope, the box should be about 12 inches on a side and 15 inches high, and proportionately larger for larger telescopes. You can build it from any material, but functor board (available from art-supply and craft stores) works well, because it is both lightweight and white. (If the light box will be handled roughly, use plywood instead.)

To flat-field lunar and planetary images where you’re using a long focal ra-

## Section 5.5: Shooting Calibration Frames

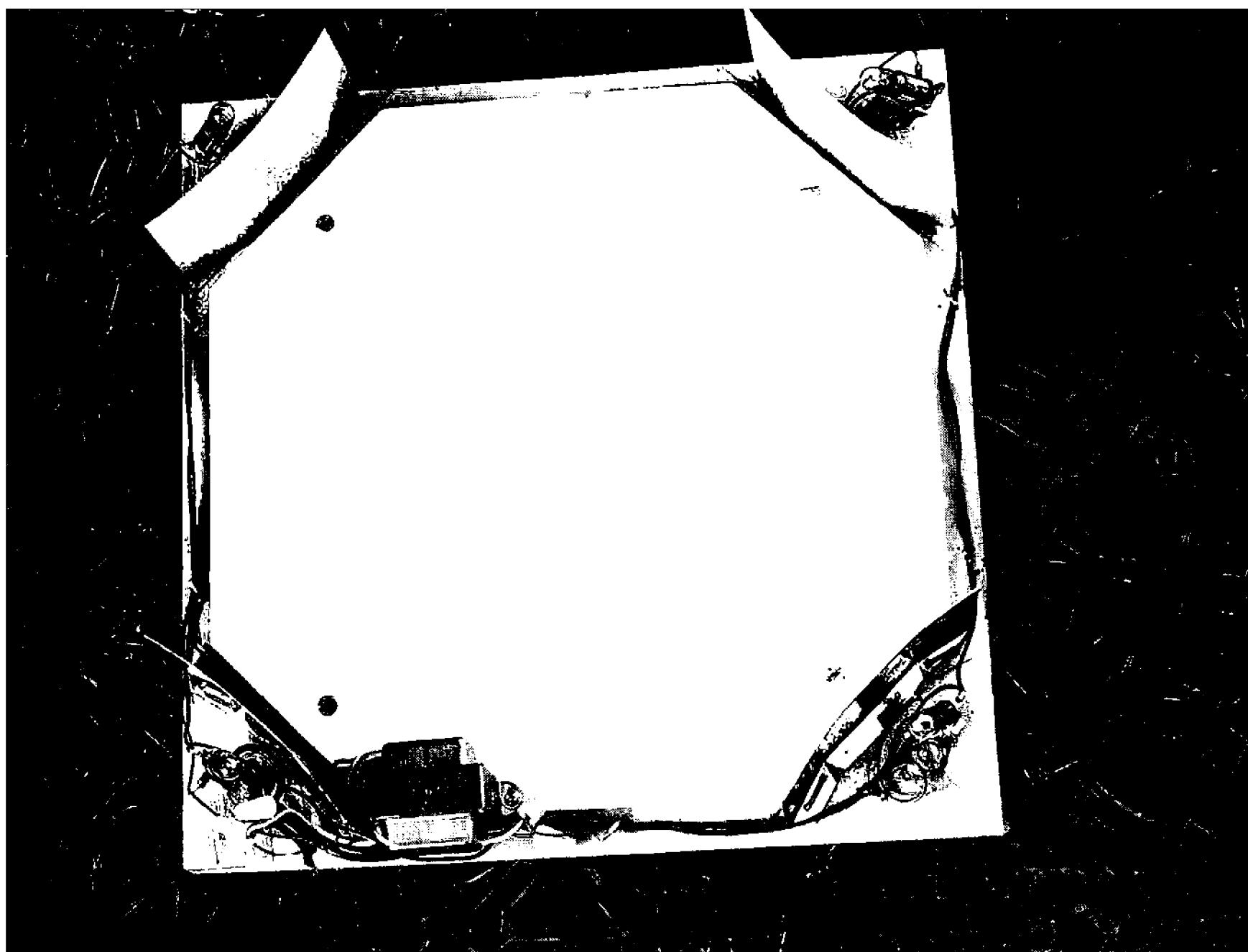


Figure 5.5 Four small halogen lamps send light into the box, where it is reflected by the white interior walls. After several reflections, the distribution of light reaching the milk plastic sheet from the box is almost perfectly uniform. Small shields prevent the lamps from illuminating the milk plastic directly.

tion, you will need a bright light box. Low-wattage halogen lamps operating on 120-volt house current provide ample light. However, for flat-fielding deep-sky images, you will need a considerably less bright light box. Small tungsten-halogen lamps (such as those made for mini-Maglights) can be run from a lantern battery and work well. Adjust the placement of the lamps and design of the box so that an exposure of 2 to 10 seconds is required to produce a half-full-well signal.

To make raw flats for a master flat-field frame, wait until you have everything focused and working smoothly. Place the box on the front of the telescope, turn on the lamps, and make test exposures to find the integration time that reaches half-full-well. Make enough raw flats to reduce noise to a negligible level when the flats are averaged—ten or more virtually guarantees a good result. Then without changing anything, using the same integration time, turn off the lamps and make the same number of flat dark frames. For example, you might standardize on 16 raw flats with the lamps turned on and 16 flat-darks with them off. For color images, you should (ideally) make a separate set of flats for each color filter—but in practice, you may find that “white light” flats are adequate.

Master flats made using the techniques described above easily attain a signal-to-noise ratio of 1000 or better, so you can be sure the flat field will add negligible noise to the final image.

## Chapter 5: Imaging Techniques

- **Tip:** *AIP4Win's automated calibration functions do the work of converting your flat-field data into a master flat frame. You need only to select the raw flats and the flat darks, and the rest is automatic.*

### 5.5.5 Defect Mapping

Although modern CCDs boast remarkably high quality, manufacturers offer them in several grades. Grading is usually based on the number of point defects (single-pixel defects), cluster defects (several contiguous defective pixels), and column defects (all or part of a column is dead). Regular “hot” pixels are not considered defects, because they can be corrected by dark subtraction and flat-fielding. In large arrays, even the highest grades may have a few point defects.

However, defects can be corrected by mapping them and replacing those pixels known to be defective. The defect map can be stored in a specially coded image in which pixel values act as instructions to apply a specific action, such as replacing a point defect with the median of the surrounding pixels. Defect mapping and correction need to be done only once for a given CCD; it can then be applied after calibration to all images taken with the camera.

- **Tip:** *Defect mapping is built into AIP4Win's calibration function. To make a defect map, make a uniform flat field and threshold it to reveal the non-responding pixels, clusters, and columns. AIP4Win can classify defects by type. After dark subtraction and flat-fielding, the defective pixels, rows, and columns are replaced with the median of nearby pixels.*

## 5.6 Imaging with Digital SLR Cameras

Digital cameras offer an attractive alternative to the astronomical CCD camera. Not only are they far less expensive (on a cost-per-megapixel basis), but they produce color images without the hassle of shooting separate filtered images. Digital SLRs offer the added advantage of being easy to use with a wide range of standard camera lenses, as well with telescopes.

They have excellent technical specifications. Readout noise is typically under 10 electrons r.m.s., and dark current is low enough to allow exposures of four minutes or longer, depending on the ambient temperature. The full-well capacity tends to be shallow, but it is fully utilized when the CCD or CMOS sensors are digitized and stored in a 12-bit “raw” format.

The greatest drawback from the standpoint of astronomical imaging with standard off-the-shelf digital SLRs is the integral color-balancing filter. This filter blocks the infrared and severely attenuates red light—however, special astronomical versions of digital SLRs are available without the filter. Because of the color-balancing and the integral Bayer array filters, pixels on the sensor receive only 10% to 20% of incident light, so to achieve the same signal-to-noise ratio as a cooled and unfiltered astronomical CCD camera, an off-the-shelf digital camera requires 25 times the exposure as the astronomical CCD. This means that to get a



Figure 5.6      Digital single-lens reflex cameras complement traditional astronomical CCD cameras by offering direct color imaging plus interchangeable lenses—and they are great for terrestrial photography, too! Furthermore, new models appear every few months with more and better features.

“solid” image, bright deep-sky objects require at least 5 to 10 minutes exposure, and faint ones need 60 minutes or more total exposure time.

To make effective use of digital SLRs, try these suggestions:

**Use a modest ISO setting.** In the camera’s raw format, an ISO setting of 200 to 400 fully samples the readout noise of the detector. Raising the ISO to high values multiplies both signal and noise—but does not capture additional information. At high ISO settings, output values reach saturation and reduce the dynamic range of the image data captured.

**Use the raw format.** The camera’s raw format captures more information from the detector than its other output options, so for astronomical imaging, its use is mandatory. If the camera offers the option of saving a raw image and a compressed JPEG image simultaneously, save both.

**Use fast optics.** You can offset the absorption of the white-balance filter and Bayer array by using digital SLRs with fast optical systems. An  $f/2.8$  camera lens is almost ideal; the best telescopes will have fast optics.

**Use long exposures.** Use the longest exposures possible consistent with keeping dark current reasonably low. Your goal is to accumulate enough photoelectrons that Poisson noise is substantially greater than readout noise.

## Chapter 5: Imaging Techniques

**Don't skimp on optical quality.** Digital SLRs have small pixels. Optical systems that produce soft images in terrestrial shooting produce bloated star images. Achromatic camera lenses and refractors generally produce star images surrounded with a blue halo of unfocused starlight.

**Infinity focus is not infinity.** With camera lenses, infinity focus usually does not yield the sharpest star images. It will almost certainly be necessary to make test images to determine the best focus for your camera's normal lenses.

**Focusing is critical.** Because the camera's focus screen is too small for critically sharp images, you will need to focus by attaching the camera to a computer, taking test images, and examining them until you achieve best focus.

**Accurate guiding/tracking is essential.** The small pixel size of the digital SLR reveals guiding or tracking errors. For effective long-exposure imaging, you will need either a very accurate drive system or some means of accurate guiding.

**Stack multiple images.** Reaching an aesthetically acceptable signal-to-noise ratio in the sky background requires collecting lots of photons, and this demands a lengthy total exposure. Take lots of images to register and stack.

**Use a “sunlight” white-balance setting.** As a general rule, natural sunlight is the default “white” for human vision. Settings such as “tungsten” and “fluorescent” yield unnaturally blue and green images.

## 5.7 Deep-Sky Imaging

For the most part, when amateurs think about CCD imaging, their goal is to image faint, fuzzy, deep-sky objects. (Operationally, the deep-sky list ought to include comets as well as nebulae and galaxies because comet are just another type of faint, extended object.) Astronomical CCD cameras are phenomenally successful at imaging deep-sky objects for three reasons: (1) their high quantum efficiency, (2) their linear response to low-level light, and (3) the ease with which you can correct their deficiencies with a computer.

### 5.7.1 Strategies for Deep-Sky Imaging

People have many different reasons for imaging deep-sky objects; your strategy should be governed by your imaging goals rather than a prescription of so many minutes for a globular, or so many minutes for a galaxy. After taking some test images, evaluate your results and *learn how to do better from what you did*. Nothing anyone can tell you is half as valuable as feedback from trying.

To get you started thinking about imaging strategies, we present three techniques—snapshot, guided, and track-and-stack—that satisfy different goals and require different equipment.

#### 5.7.1.1 Digital Snapshots

The sensitivity of the new breed of digital and astronomical CCD cameras allows an observer to capture fainter stars and more nebulosity in a 60-second exposure

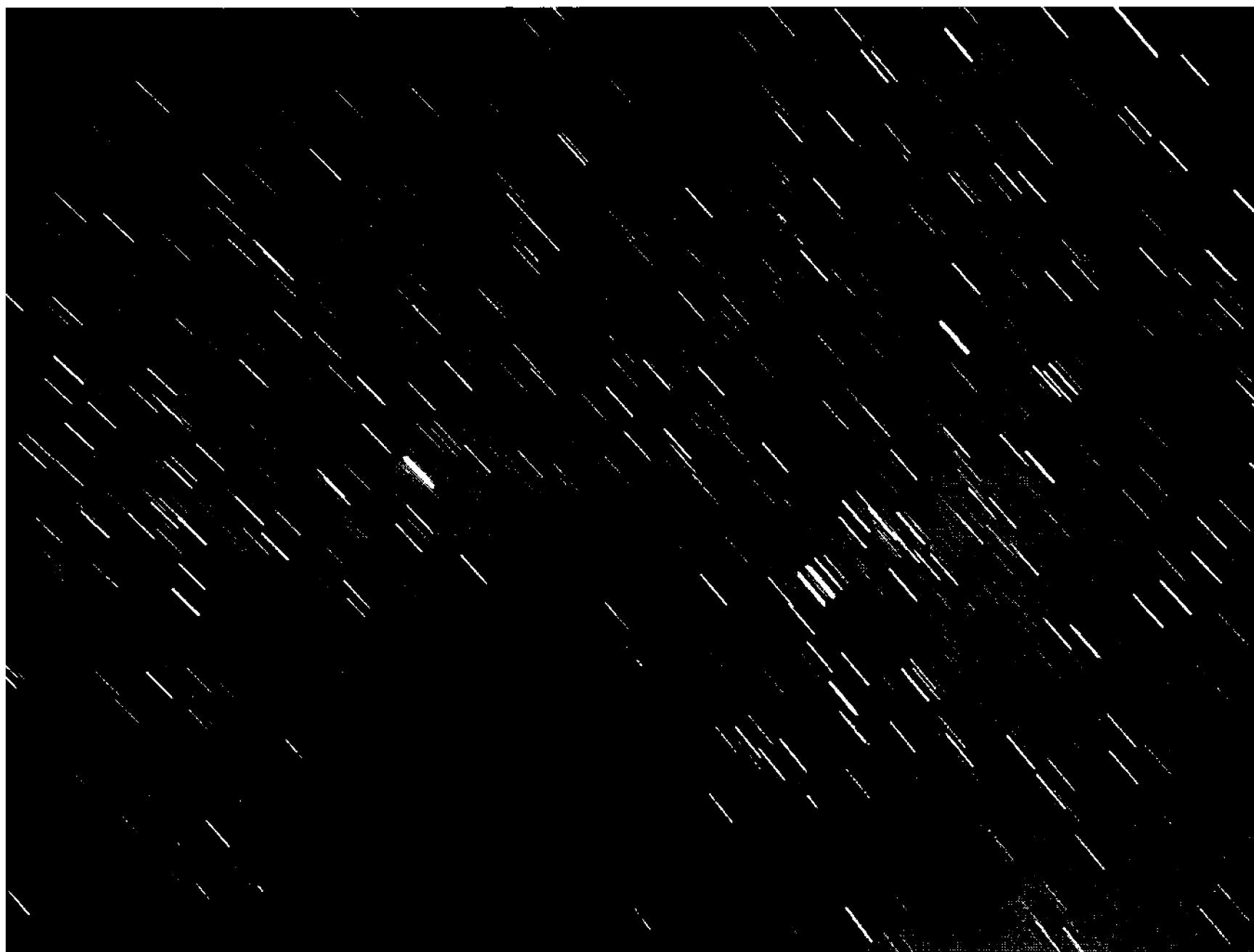


Figure 5.7    Snapshots with digital SLRs capture night scenes much as you see them. In this image, winter stars and the Milky Way trail behind a wind-blown palm tree. On a telescope, an astronomical CCD camera can capture more than you can see with the same telescope in exposures of a few seconds.

than their film-shooting counterparts of the previous decades could capture in an hour. Almost everyone who comes to CCD imaging goes a little nuts at first, banging off images of 50 nebulae one night and 100 galaxies the next.

With an astronomical CCD camera, 60-second snapshots require nothing more than a telescope with a reasonably fast focal ratio ( $f/8$  or faster) and a relatively accurate clock drive. Snapshots work because most astronomical CCD cameras have become sky-background limited in a 60-second exposure. Although longer times give you better signal-to-noise ratio, short exposures capture a nifty image of practically anything in the sky.

For snapshots, the best optical system is a medium-aperture telescope (4 to 12 inches) with a fast focal ratio ( $f/5$  or less). Newtonian reflectors, apochromatic refractors, and Schmidt-Cassegrains equipped with focal reducers all fill the bill. The short focal length provides a generous field of view and the low focal ratio provides a bright image at the focus. Light pollution and Moonlight are minor problems when the exposures are short and the CCD is red-sensitive: just ignore the streetlight and Moon and blast away every clear night.

Digital cameras are also good for making snapshots, and their versatile lenses make them ideal for night-time scenic shots. However, because the filters in their Bayer array reduce the amount of light reaching the sensor, good results re-

## Chapter 5: Imaging Techniques

quire: 1.) a fast camera lens (those that are  $f/2.8$  and faster), 2.) exposures of 4 minutes or longer, or 3.) sticking to the brightest deep-sky objects.

If the optics are free of significant vignetting, basic calibration is entirely adequate. By settling on a single universal exposure time on the order of 60 seconds, only one or two sets of dark frames are needed, so the overhead required for calibration is minimal.

Snapshots are actually very effective for surveying galaxies in search of supernovae or covering large patches of sky repeatedly in hopes of discovering asteroids. In a few months you could easily complete an album of all the Messier objects; or, if you're in a hurry, you could try a Messier Marathon with your CCD camera!

### 5.7.1.2 Guided One-Shot Imaging

With telescopes that have slow optics, snapshot exposures don't go deep enough to produce satisfying results. Longer times will get you down into the sky background and produce deeper images. Exposure times are limited only by the observer's stamina, CCD dark current, star image blooming, or background sky brightness.

Longer exposures mean more photons fall on each pixel, and the larger the number of photons, the stronger the image signal relative to image noise. Under a reasonably good sky, increasing the exposure from 60 seconds to 240 seconds adds 1.5 magnitudes and a huge increase in the number of stars recorded. Faint nebular details and outer spiral arms emerge from grainy indistinctness in the sky background to take on form and substance. The bottom line is that compared to snapshot imaging, long exposures reveal a lot more.

Making long exposures almost always requires guiding, either manual or automatic, but the results will repay the effort many times over. Single exposures of 5 to 10 minutes should result in excellent images that you can be proud of.

For digital cameras, guiding longer exposures pays big dividends. Very few clock drives will track unaided for four minutes or more and still give round star images, but guiding lets you push out to ten-minute exposures that capture the sky at a dark-sky site, with lots of stars, and of course, your deep-sky quarry.

### 5.7.1.3 Guided and Unguided Track-and-Stack Imaging

Accumulating dark current or blooming field stars usually limits CCD integrations to 20 minutes or less, and suburban skies seldom allow more than 10 minutes in a single integration. Worst of all, errors in the telescope clock drive often limit well-tracked CCD integrations to 60 seconds or less.

Track-and-stack imaging is an effective way to lengthen the total integration time by dividing a long exposure into many short ones. To collect an hour's worth of photons, you simply track (i.e., register) and stack (i.e., sum) 60 integrations of 1 minute each. Do this properly and you're well on your way to reaching 21<sup>st</sup> magnitude with a modest telescope and no guiding.

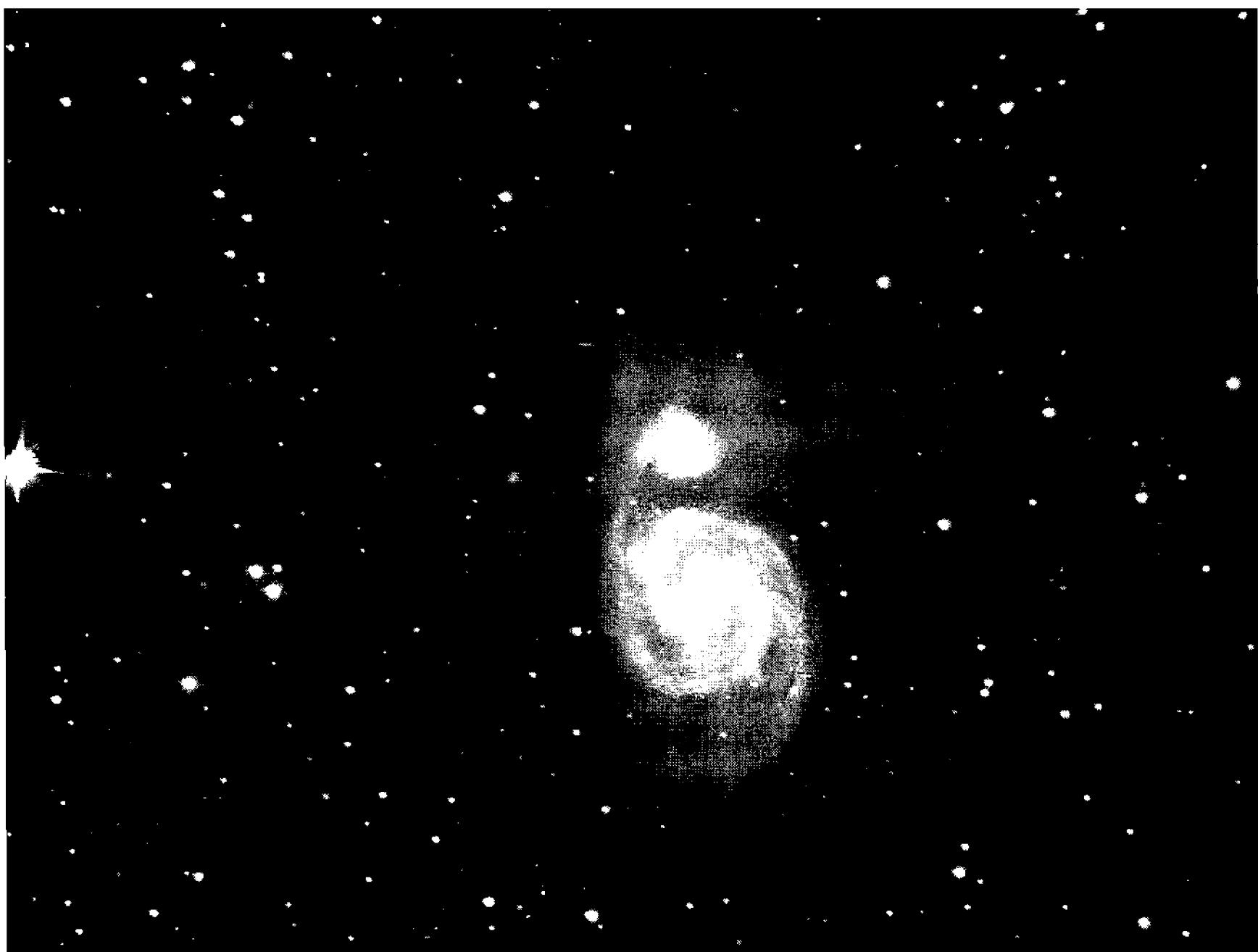


Figure 5.8 Stacking is the key to making very deep images of extended faint objects. For this image made with a 6-inch f/5 Newtonian, 64 60-second exposures were stacked to reveal the far-flung outermost parts of the Whirlpool galaxy, M51. In this image, the range of light has been drastically compressed.

Stacking images does exact a toll: to collect an hour's worth of photons in 60-second increments, you must read out the CCD 60 times. The photons don't care whether you gather them in one integration or 60—the Poisson statistics that govern their behavior work out the same. However, each time you read out an image from the CCD, the sensor's amplifier adds another dose of readout noise to the signal. With most astronomical CCDs and many digital cameras, however, Poisson noise becomes the dominant noise source and overwhelms the noise added by the multiple stack-and-track readouts.

- *Tip:* You can stack an unlimited number of images using **AIP4Win's Multi-Image Auto-Process Tool**. This tool automatically calibrates, registers, preprocesses, and stacks many integrations. You can use this tool equally well with astronomical CCD images, digital camera JPEG images, and digital camera raw images.

To accumulate the greatest number of photons with the least readout noise, your best strategy is to stack the longest exposures that you can do routinely. This implies that you should guide the exposures or use an autoguider. Under a dark sky, rather than shooting 60 integrations of 1 minute each, it would be better to take 10 integrations of 6 minutes each. The 6-minute sub-exposure is sufficiently long to collect lots of photons, but short enough to avoid the limitations of pro-

## Chapter 5: Imaging Techniques

tracted guiding, dark current, hot pixels, saturation, and blooming.

### 5.7.2 Making Good Deep-Sky Images

Through experience, every observer eventually develops a suite of personal techniques for making good deep-sky images. To help you get started up the learning curve, here are a few things to remember when you're shooting images under the night sky.

**Get a good drive for your telescope.** If your mounting has an inaccurate drive, everything you try to do with your CCD camera is much more difficult. To make good images, you will eventually have to replace it—the only question is when. The sooner the better.

**Get good polar alignment.** You need accurate polar alignment and a good drive system running at sidereal rate. Learn solid polar alignment skills; or better yet, put your telescope in a permanent shelter and align it for once and for all.

**Don't put up with shifting optics.** If your telescope's optics shift and you need to refocus for each new object, find and correct the problem. It is difficult or impossible to do good imaging with optics that don't stay put.

**Use one integration time.** Unless you are shooting one of the very bright deep-sky objects, it is best to use an exposure time that produces a good signal-to-noise ratio in the sky background. This means that unless you have compelling reasons to do otherwise, use the same exposure time for every object.

**Allow time for cooling.** Most telescopes don't give good images when they are warmer than the surrounding air. Be sure to set up the instrument, roll back the roof, or open the dome at least an hour before you plan to start taking images.

**Keep detailed records.** To help you learn from the hours you spend under the stars, it really helps to keep a written record of your imaging sessions. Note objects, file names, filters, exposure times, sky conditions, and anything else you might want to know later. When you get great images, all the information you need to repeat your success will be ready and waiting.

**Shoot dark frames.** Repeat this mantra: "I love dark frames. Dark frames are critical to making high-quality deep-sky images." As soon as the CCD camera has come to thermal equilibrium, cap the telescope and shoot a dozen dark frames at the exposure time you plan to use that night. (If you plan to calibrate using the Advanced calibration procedure, make the exposures two to five times longer than those you plan to make.)

Shoot a backup set of dark frames when you take your midnight cookie and coffee break. Just because you take a break doesn't mean that your CCD camera needs a break, too. Let it do something useful while you gobble a box of Oreos and guzzle caffeine.

**Shoot flat fields.** The best way to shoot flats for deep-sky images is with a light box. The light-box technique allows you to make the flats without pointing the telescope down, fussing with the dome, or trying to judge the moment when the twilight sky is at the right brightness. Determine the integration time necessary



Figure 5.9 The globular cluster M13 sprawls between its familiar twin guardian stars. Observed visually, the cluster is a delicate ball of stars; to the CCD observer, however, the cluster reveals a much greater extent. Taken together, visual observing and CCD imaging complement each other.

to produce a peak pixel value between half and two-thirds of the full-well capacity of the CCD chip. It is important to make sure that no pixels reach saturation. Save 8 to 10 integrations as well as an equal number of flat dark frames taken with the same integration time. Shooting flats takes some time and effort, but it's a good investment in getting top-notch results.

### 5.7.3 Imaging Deep-Sky Targets

Imaging strategies depend to some degree on the type of object that is the target of your efforts. Here is a rundown on objects that you may wish to image:

**Open Clusters.** The familiar examples in the Messier list consist of a few hundred fairly bright stars spread out over a fairly large angular diameter, lending themselves well to snapshot exposures with short-focus telescopes. There are literally thousands of faint open clusters scattered along the plane of the Milky Way, virtually ignored by amateurs. A technique that is effective in making these somewhat bland objects look more interesting is to place a mask made of three sets of three  $\frac{1}{8}$ -inch wooden dowels spaced 1 inch apart at  $60^\circ$  angles to one another over the aperture of the telescope. This mask produces six bright diffraction spikes that make the stars stand out clearly.

Young clusters such as the Pleiades still contain dusty remnants of the mo-

## Chapter 5: Imaging Techniques

lecular clouds that gave them birth, and many others are still embedded in clouds of gas and dust that are still actively forming new stars. NGC 6611 (in the Eagle Nebula) and NGC 2244 (in the Rosette Nebula) are examples. Most of the observational interest in these star-forming regions is directed to intricate tracery of the nebulosity. The clusters make recording the surrounding nebulosity difficult because their bright stellar constituents can cause blooming.

**Globular Clusters.** Roughly 160 of these objects orbit our Galaxy, all of them accessible to amateur CCD cameras. Although globulars are strongly concentrated toward the Sagittarius-Scorpius region of the sky, a few are found well away from the Milky Way.

Globulars vary in angular size, brightness, and degree of concentration. To resolve their stars, it is best to shoot them with a long-focus telescope having a fairly large aperture, since both scale and light-grasp are needed. Although short exposures easily resolve the dense central region of a globular, for really outstanding images you need to make long integrations to pick up the outlying halo of stars that surrounds the bright core. In a deep exposure, the angular diameter of a typical globular is four to six times larger than that recorded in a short integration.

Globulars are old objects, so their stars have evolved into a mixture of bright, white main-sequence stars and very luminous red giants. Color images made with CCD cameras show the disparate colors of the two varieties very nicely.

Many other galaxies are surrounded by a halo of globular clusters. Before the advent of CCDs, most of these objects were too faint for amateurs to observe, but current technology places the globular clusters around the Andromeda Galaxy well within amateur reach.

**Planetary Nebulae.** These are transient shells of gas thrown from the surface of a dying giant star. They are popular with visual observers because of their high surface brightness, due in part to the fact that the bulk of their emission falls at a wavelength of 500.7 nanometers, near the peak sensitivity of the dark adapted human eye. Most planetaries have a small angular diameter, but the high surface brightness means a telescope with a moderate to long focal length and a slow focal ratio—e.g., the standard 8-inch  $f/10$  Schmidt Cassegrain telescope—is well suited for them. What observer with a new CCD has not immediately set out to shoot an image of the Ring Nebula?

**HII Regions.** HII regions (read as “H-two”) are among the showiest celestial objects. These are fairly dense clouds of gas and dust that are soon to form, are now forming, or have just formed new stars; and as the new stars emit copious amounts of ultraviolet radiation, they ionize the gas, causing it to glow. The Lagoon Nebula (M 8), the Orion Nebula (M 42), the Omega Nebula (M 17), and the Eagle Nebula (M 16) are HII regions. Each of these objects has a clutch of hot, young stars providing the energy that makes it fluoresce.

The term “HII” means the glow comes from ionized hydrogen. Although the bright red  $H\alpha$  (hydrogen-alpha) line dominates the spectrum of many nebulae, the blue-green  $H\beta$  line and the deep blue  $H\gamma$  combine with the carmine red of  $H\alpha$  to give these nebulae a vivid “electric pink” hue in color images.

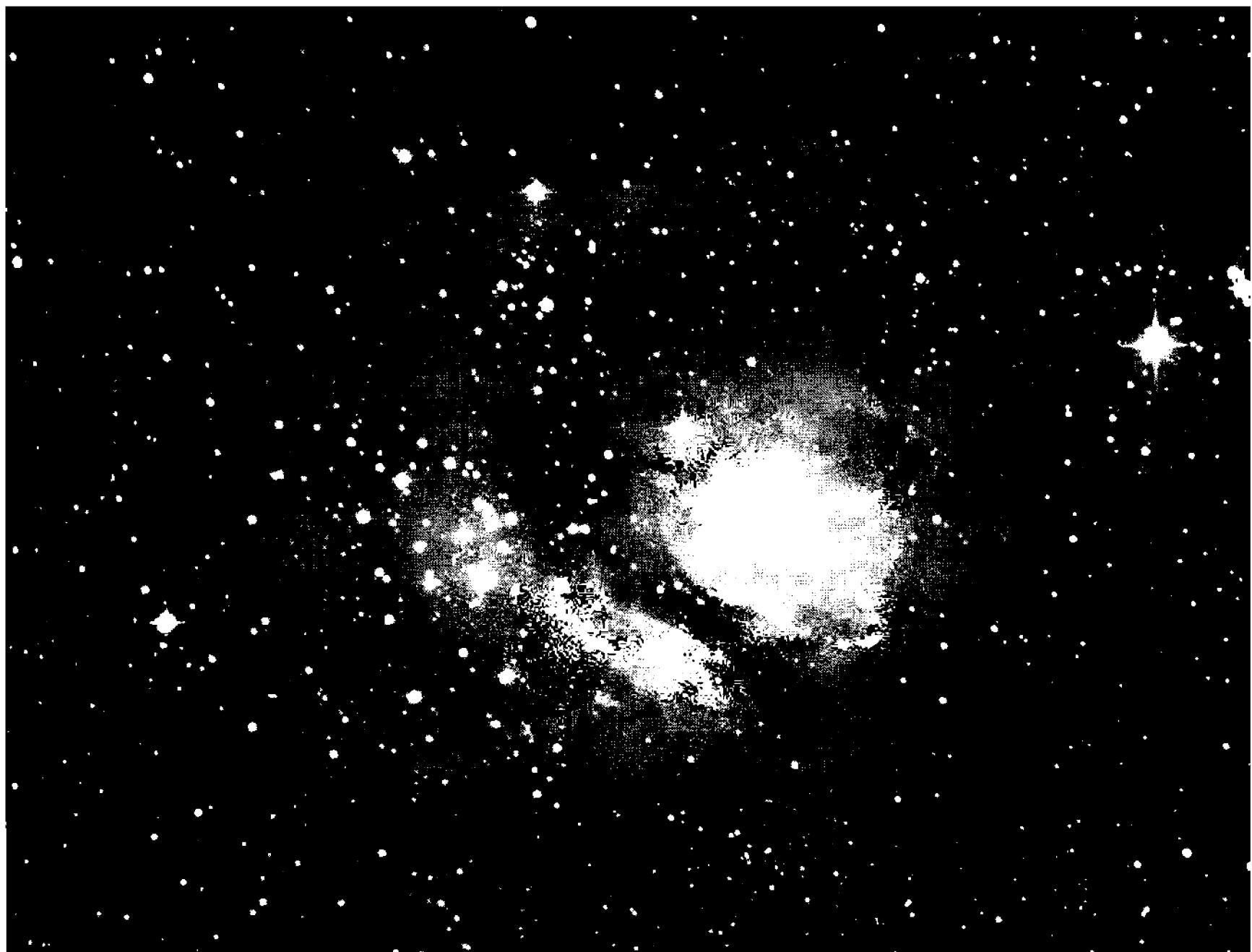


Figure 5.10 The Lagoon nebula is usually described as an HII region, but CCD images—even those made with modest telescopes—show it to be a much more complex object. The Lagoon holds a conspicuous open star cluster, numerous dark nebulae, reflection nebulosity, and elephant-trunk structures.

Red-sensitive CCD cameras are wonderfully sensitive to HII regions, especially when a narrow-band red filter is used to transmit H $\alpha$  light while blocking light from the sky background and the bulk of starlight. The strongest filters are interference filters with a passband of about 10 nanometers; HII regions imaged this way stand out vividly against a jet-black sky background.

**Reflection Nebulae.** Where clouds of interstellar material containing dust and gas aren't close enough to bright stars to be ionized, they still can reflect light and thus become visible. The Merope Nebula in the Pleiades is perhaps the best-known example, appearing blue because short-wavelength light is more efficiently scattered than redder light. Others include M 78 in Orion and the beautiful nebula surrounding R Coronae Australis.

Reflection nebulae reveal the incredible clumpy, wispy, stringy nature of interstellar gas and dust. Because they tend to be faint, a fast optical system, along with dark skies and long integration times, helps build the high signal-to-noise ratio needed to distinguish their faint light against the background sky.

**Dark Nebulae.** Imaging these objects requires above all else a dark sky. Dark nebulae are clouds, wisps, and strands of interstellar gas and dust that have no stars nearby to reflect light from, or to be ionized by. We see them because they block the passage of light. For the most part, these opaque veils cling to the plane

## Chapter 5: Imaging Techniques

of the Milky Way, with notable examples among the dark clouds forming a great rift that runs from Cygnus through Aquila, Scutum, Scorpius, and down into Lupus and Norma.

Dark nebulae become visible in three ways:

- as poorly defined regions with too few faint stars,
- as large, sharp-edged regions silhouetted against the pervasive background glow of the Milky Way, and
- as compact dark globules silhouetted against HII regions.

In all cases, the contrast between the dark nebula and the surrounding sky is very low. Good images require dark skies, a telescope or lens with a fast focal ratio, and long exposures. In combination, these yield images with high signal-to-noise ratios that reveal wonderful detail in these hard-to-see objects.

The most famous dark nebula is the Horsehead in Orion, a compact cloud silhouetted by the light of IC434, a diffuse HII region. The Milky Way in Sagittarius and Scorpius abounds in dark nebulae.

Like most particulate matter, dark nebulae absorb and scatter blue light more effectively than red; hence they are more opaque in the blue. Because at a dark-sky site the sky is darkest in the blue, a CCD with high blue sensitivity is very desirable.

**Galaxies.** Galaxies exert a powerful fascination for observers. They are huge and distant, and because some resemble our own Milky Way, we see in them a reflection of ourselves. Observationally, galaxies are wonderfully varied. Like snowflakes, no two are alike; yet the spirals all share a common master plan.

Well over 50,000 galaxies—those larger than 90 arcseconds and brighter than magnitude 15—lie within the grasp of amateur telescopes. The biggest and brightest spirals—most of the galaxies in the Messier catalog—fall easily to small telescopes with modest CCD cameras. Smaller and dimmer ones are best imaged with apertures in the 12- to 16-inch range, under dark skies, on nights of excellent seeing. This range of aperture can reach the limit of resolution imposed by seeing, yet still provide a reasonably wide field of view.

Galaxies offer a wide range of morphologies, from bland ellipticals through tight spirals to open and barred spirals. We may view these latter galaxies face-on, at some oblique angle, or edge-on. Irregular galaxies and dwarf ellipticals—especially the nearby specimens in the Local Group—are challenging to image, yet interesting because they are our neighbors.

Elliptical galaxies contain large numbers of older, evolved stars: they are basically yellow in color. Spirals have old stars in their cores, but their arms are defined by brilliant young blue stars and the HII regions where they are formed. Images of spiral galaxies taken with red-sensitive CCD cameras emphasize the underlying “smooth disk” population of the galaxy; in red light, spiral structure is weak.

A dramatic exception is imaging spirals through a narrow-band H $\alpha$  filter centered on 656 nanometers wavelength. The narrow bandwidth admits little light

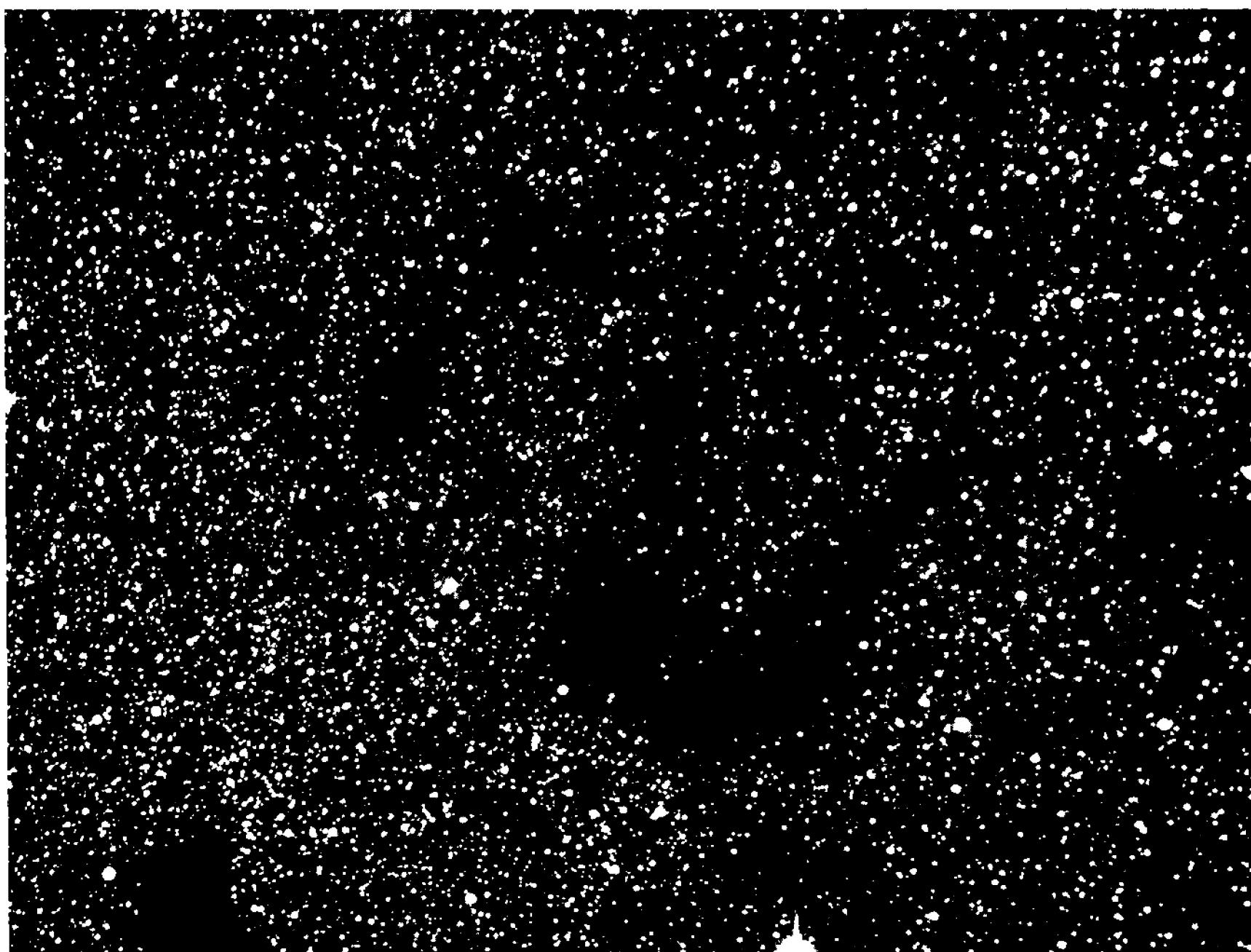


Figure 5.11 Silhouetted against the star-rich Milky Way, the Snake nebula (Barnard 72) is a tiny part of a much more extensive system of dark nebulæ that blankets much of Sagittarius, Ophiuchus, and Scorpius. Use your CCD camera to complement visual observations of these hard-to-see objects.

from the hordes of yellow and red stars in the disk; so most of the light is from the HII regions, which string out along the inner edges of the spiral arms like lights on a Christmas tree. With a blue-filtered blue-sensitive CCD camera, the arms appear prominently, outlined in young blue stars and HII regions.

Searching for supernovae is a popular sport among galaxy imagers. These occur in class Sb and Sc spirals at a rate of about one per galaxy per century. Since a supernova shines as brightly as the rest of its galaxy combined for several months, finding one is simply a matter of comparing tonight's image with an image taken at a different time.

On the average, you can expect to find one supernova for every 2500 galaxies that you examine. Such a discovery is, of course, a much-sought prize among serious deep-sky observers and CCD imagers.

**Clusters of Galaxies.** Traditionally reserved for professional astronomers and amateurs with enormous Dobsonians, distant clusters of galaxies add a new twist to deep-sky imaging. Everyone knows the Virgo cluster, and many observers have viewed the relatively nearby Perseus (Abell 426) and Coma (Abell 1656) Clusters, as well as the more distant Ursa Major (Abell 1377) and Hercules (Abell 2151) Clusters. These vast and distant agglomerations reveal themselves to a CCD on an 8-inch telescope. Within a few years, it's a good bet that amateurs will push

## Chapter 5: Imaging Techniques

the frontiers to clusters 1,000 megaparsecs distant.

Although a small telescope can detect galaxy clusters, imaging them really calls for a large telescope, a large image scale, and excellent seeing. With a short focal length or in poor seeing, the galaxies and stars will be indistinguishable. The value of the image depends on sufficient resolution to distinguish the tiny fuzzball galaxies that comprise the cluster from the scattering of point-like foreground stars.

**Quasars.** Quasars are thought to be very bright, active nuclei of galaxies; many have redshifts that place them a significant fraction of the distance across the Universe. The quasar which appears brightest is a 12th magnitude object in the constellation Virgo called 3C273; it has a relativistic jet that was discovered on photographic plates taken with the 200-inch telescope on Palomar Mountain. With an amateur telescope and a CCD camera, this jet appears as a short spike extending from the star-like core. Several hundred quasars are within the reach of amateur instruments.

Many quasars are variable on time-scales of a few days or weeks, an indication that their light comes from a region less than a few light-weeks across. Since their energy output typically flickers erratically by several tenths of a magnitude, an amateur astronomer with a small telescope can easily contribute photometric observations of quasars to organizations such as the American Association of Variable Star Observers (AAVSO).

**Comets.** The morphological properties of comets vary greatly, from 18th magnitude fuzzballs to great comets with tails that span huge arcs of sky. You can treat faint comets as you would any faint deep-sky object. An exposure of many minutes may reveal a faint wisp of coma surrounding a star-like nucleus. However, because of the comet's motion against the stars, you may need to guide on its nucleus or resort to making unguided images and then combining them with track-and-stack software. Between the faint fuzzballs and the great comets are objects of every size, brightness, and degree of activity.

Comets shine by sunlight reflected from fine particulates, and by fluorescence from cometary gases excited by sunlight. The more active a comet is, and the closer it is to the Sun, the greater the proportion of light that comes from fluorescence. Yellowish reflected sunlight is readily detected by typical amateur CCD cameras, but the fluorescent spectrum is dominated by bands of the CN molecule in the ultraviolet and the C<sub>2</sub> molecule in the blue-green, as well as weaker bands of CH, C<sub>3</sub>, and NH<sub>2</sub> spread across the spectrum from yellow to deep blue. With a blue-sensitive CCD camera, narrow-band filters can isolate the CN molecular bands for dramatic images of the cometary gas tail.

To make images of a large, bright comet, it is necessary to use an optical system such as a camera lens to get a wide field of view; for close-up shots, a wide range of telescopes is suitable. Telephoto lenses and short-focus telescopes will record structure in the outer coma and the inner degree or two of the tail; long-focus instruments can capture features of the inner coma and near-nuclear structures such as shells and jets.



Figure 5.12 CCD images allow you to reach deep into time and space from your backyard observatory. At first glance, you see a few dozen Hercules cluster galaxies; but look again and you'll see hundreds of them populating this rich cluster. Look yet again and you'll spot interacting pairs of galaxies.

Bright comets are remarkably dynamic, and you should plan your imaging accordingly. In the tail of Comet Hyakutake, for example, streamers within a few degrees of the nucleus changed from one minute to the next, and the shells and jets near the nucleus showed obvious expansion in a ten-minute interval. Exposure times should be short, and you should think in terms of movies or animations to show the changing structures.

## 5.8 Lunar, Planetary, and Solar Imaging Techniques

The Sun, Moon, and planets make tempting targets for digital camera, webcams, and astronomical CCD cameras. Once safely stored in your computer, the images you capture can be enhanced to reveal unprecedented detail. However, if you have actually tried it, you may have concluded that planetary imaging with digital cameras is not quite as easy as it's cracked up to be—and (of course) you are right. Outstanding planetary imaging requires strict attention to technique—but once you grasp the basic methods, it's remarkably straightforward.

Outstanding lunar and planetary images require a high-quality telescope that is well collimated, optics to match between the telescope and your digital camera, and moments of good seeing. If you are patient enough to wait for moments of

## Chapter 5: Imaging Techniques



Figure 5.13 As Comet Hale-Bopp approached us from the outer Solar System, we saw it nearly head-on, with six tails splayed around a compact coma. To see the tail structures, copies of the image were rotated and subtracted, suppressing star images and enhancing the tail. Image courtesy of Al Kelly.

good seeing and to take lots of images, the results will astound you.

### 5.8.1 Obtaining Excellent Images

Taking outstanding planetary images requires a telescope that has diffraction-limited optics. You must also collimate the optics precisely, employ high-quality enlarging optics, and be patient enough to take advantage of excellent seeing when it occurs at your observing site. It takes all four. Start with excellent optics, check their collimation to insure that they deliver their best, and use high-quality eyepieces or Barlow lenses to match the image to the camera.

**Test your telescope's optics.** The crucial factor is not what *type* of telescope you have, but whether it forms high-quality images. Pop in an eyepiece and spend a few hours on several different nights looking at the Moon at 40x or 50x per inch magnification. You will probably see a lot of turbulence, but in moments when the air settles you should see crisp, diffraction-limited detail. On nights with good seeing, use the evaluation techniques detailed in *Star Testing Astronomical Telescopes* by H.R. Suiter. Deep-sky observers might not care about a quarter-wave of spherical aberration, a few bumpy zones, and a half-wave of coma, but they're negative factors in planetary imaging.

**Check optical alignment.** When good optics get out of alignment, they act

## Section 5.8: Lunar, Planetary, and Solar Imaging Techniques

like bad optics. Collimation is nothing more than making sure that each component in the optical system is where it is supposed to be—not tilted, not off center, not ahead of or behind its proper location. Begin alignment with a set of collimation tools or a laser collimator, and follow the manufacturer’s instructions. After alignment, carefully assess the quality of star images inside and outside focus and at best focus. Here again, *Star Testing Astronomical Telescopes* should be your bible. Residual astigmatism or coma is a red-flag warning that something is wrong with the alignment.

**Strive for thermal equilibrium.** Although observers cannot control atmospheric turbulence (except through their choice of observing location), a key element in exploiting quality optics is making sure that the telescope tube, the observatory (or observing site), and its surrounding area are conducive to good local seeing. In Newtonians, the tube should have low thermal mass (i.e., not retain heat), allow free exchange of outside air, and include a fan to pull air down the tube and over the mirror to aid equilibration of their temperature with that of the surrounding air.

Your observatory or observing area should be open and airy; the building, equipment, and floor must not retain the heat of the day. Concrete block walls may provide necessary security, but they virtually guarantee that plumes of hot air will distort images for many hours after dark. A light wooden building with a roll-off roof makes an inexpensive structure that cools rapidly. The building should be raised a few feet off the ground to allow air to circulate and aid in cooling.

### 5.8.2 Focal-Ratio Matching Optics

It is almost always necessary to enlarge planetary and lunar images from their size at the focus of your telescope to a scale appropriate for digital imaging. At the primary focus of most instruments, the pixels on the sensor are too large to capture diffraction-limited detail in planetary images. The diameter of the bright central portion of the diffraction disk,  $d_{\text{FWHM}}$ , is:

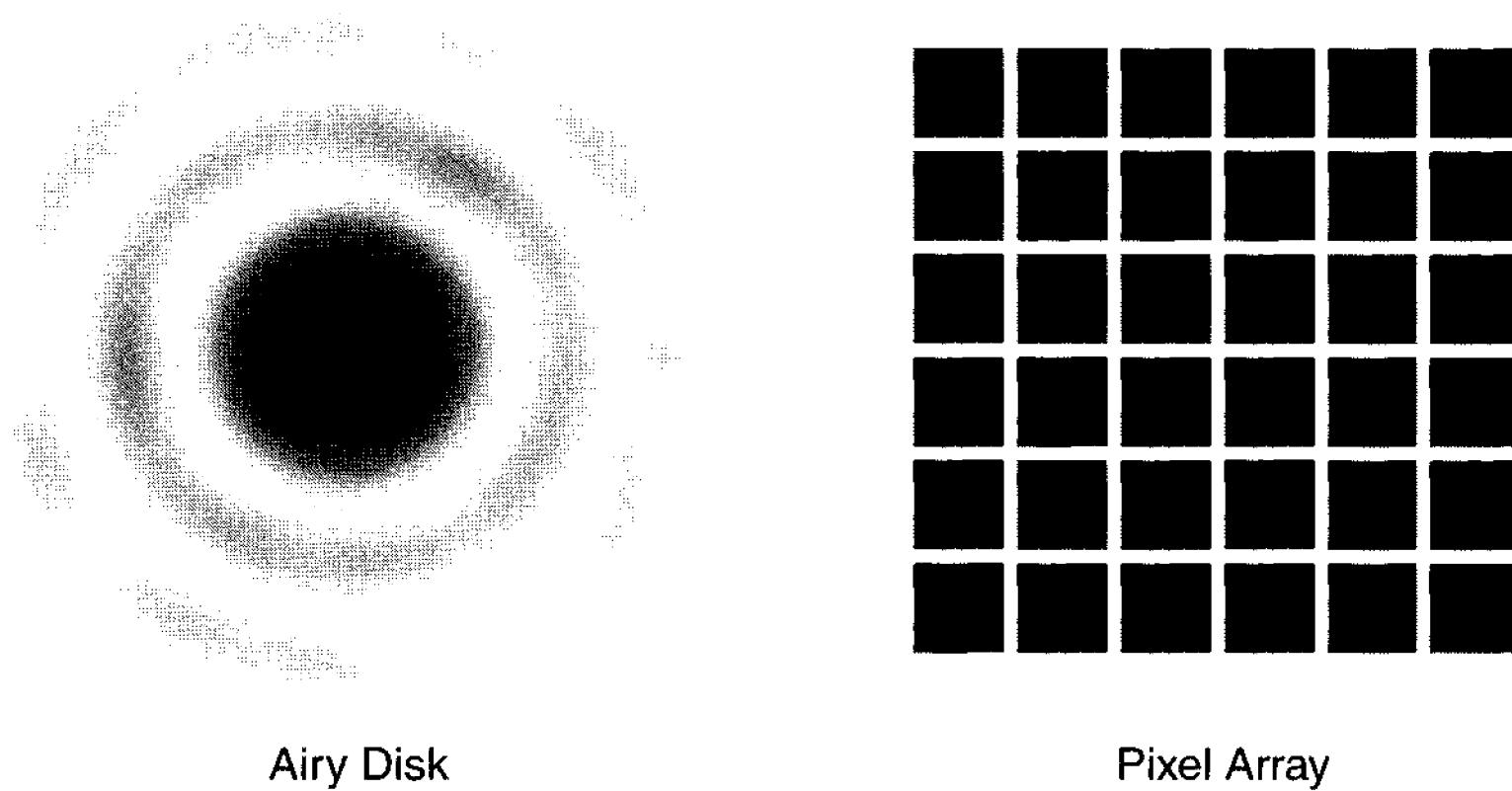
$$d_{\text{FWHM}} = 1.02\lambda N = \frac{1.02\lambda F}{A}, \quad (\text{Equ. 5.2})$$

where  $\lambda$  is the effective wavelength,  $A$  is the aperture of the telescope,  $F$  is the focal length, and  $N$  is the focal ratio,  $F/A$ . At the wavelength of peak sensitivity for many digital cameras and astronomical CCDs (550 nanometers), the diameter of the diffraction disk at the focus of an  $f/6$  telescope is 3.4 microns.

Compare this figure to the 7- to 12-micron dimensions of pixels in a typical astronomical CCD camera: because the diffraction disk is considerably smaller than the pixel size, diffraction-disk-size detail cannot possibly be captured.

For critical sampling of image detail, the bright central region of the diffraction disk should meet the Nyquist criterion—it should be at least twice as large as the pixels that sample the image. For a sensor with 10-micron pixels, the diffraction disk should be enlarged to twice their size, or 20 microns. At an effective wavelength of 550 nanometers, this corresponds to a focal ratio of  $f/36$ .

## Chapter 5: Imaging Techniques



**Figure 5.14** Theory says that to capture all of the information present in a telescope image, the bright inner portion of the Airy disk should span a minimum of two pixels on the detector. As a practical matter, however, the seeing blur rather than the Airy disk often limits the information content of telescopic images.

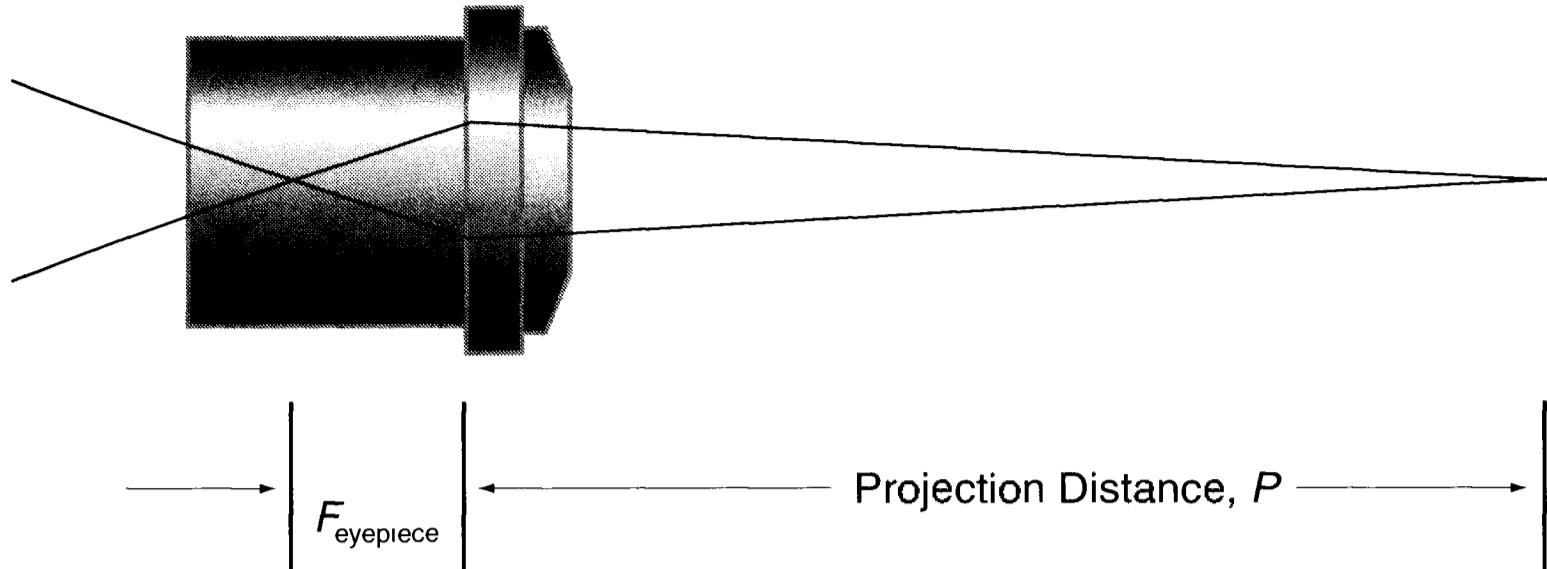
For a variety of reasons, it is usually expedient to use a less extreme focal ratio than the Nyquist focal ratio for the diffraction disk. First, atmospheric turbulence enlarges the star image into a blur that is bigger than the bright core of the ideal diffraction disk. Second, there's a trade-off between exposure time and atmospheric stability. At the Nyquist focal ratio, the image may blur during the integration; but at lower focal ratios, the integration time necessary becomes shorter, and you can "freeze" moments of good seeing.

In the end, use the Nyquist focal ratio as a starting point and let your imaging results be your guide. A practical starting point for planetary and lunar imaging with an astronomical CCD having 10 micron pixels is  $f/25$ . For a digital camera with 7 micron pixels, a reasonable starting point is  $f/18$ .

The two classic techniques for enlarging the image are projection from an eyepiece and projection with a Barlow lens. Astrophotographers have long used eyepiece projection to increase the scale of their lunar and planetary images.

**Eyepiece Projection.** In this method an eyepiece is placed slightly farther from focus than it normally is for viewing, so that instead of emitting parallel rays, it emits a converging beam that forms a real image at some distance (called the projection distance) behind it. Solar observers use eyepiece projection to view an image projected safely onto a sheet of white cardboard. For digital imaging, instead of the cardboard, the image falls on a CCD.

To set up for eyepiece projection, you must first figure out how much image enlargement you need and then calculate the eyepiece-to-CCD distance. For example, to convert an  $f/6$  Newtonian to  $f/36$ , you need to increase the focal length by a factor of  $36/6$ , or 6 times:



**Figure 5.15** The geometry of eyepiece projection is simple: to enlarge the image by a factor of  $E$ , add 1 to  $E$  and multiply by the focal length of the eyepiece. The resulting value is the projection distance,  $P$ . To avoid scattered light with eyepiece projection, place a small stop at the focal plane of the eyepiece.

$$P = (E + 1)F_{\text{eyepiece}} \quad (\text{Equ. 5.3})$$

where  $P$  is the projection distance;  $E$  is the desired enlargement; and  $F_{\text{eyepiece}}$  is the focal length of the projection eyepiece. To boost your  $f/6$  focal ratio to  $f/36$  with a 12-millimeter eyepiece, the projection distance is  $(6+1) \times 12$ , or 84 millimeters. You can set this distance with a ruler when you assemble the eyepiece projection system.

**Barlow Lens Projection.** Barlow lenses are negative (i.e., diverging) lenses placed in front of an eyepiece to increase the focal length of the telescope. Modern high-performance systems such as the TeleVue Powermate series increase the focal length of a telescope by factors from 2x to 5x, converting an  $f/6$  system to focal ratios from  $f/12$  to  $f/30$ . On an  $f/10$  telescope, a 2x Barlow increases the focal ratio to  $f/20$ , which is about right for a digital SLR with 7-micron pixels.

The distance from the focal plane to a standard Barlow lens must be set correctly to produce the design magnification. The Powermate systems were designed to produce the desired image enlargement over a much wider range of lens-to-focal-plane distances than conventional Barlows.

### 5.8.2.1 Field of View

At long focal lengths and long focal ratios, the field of view becomes extremely small. The formula for the field of view in arcseconds is:

$$\vartheta_{\text{CCD}} = 206265 \times \frac{d_{\text{CCD}}}{EF} \text{ [arcseconds]} \quad (\text{Equ. 5.4})$$

where  $d_{\text{CCD}}$  is the size of the CCD,  $E$  is the enlargement provided by the lens, and  $F$  is the focal length of the optical system. For reference, the angular size of Mars at opposition varies between 13 and 25 arcseconds, the ball of Saturn is 18 arcseconds and the rings are about 40 arcseconds from tip to tip, and the giant Jupiter

## Chapter 5: Imaging Techniques

ranges from 40 to 48 arcseconds across the equator. Even at fairly high magnification, these objects appear quite small in a telescopic field of view.

What is the field of view in making CCD images with a 12-inch telescope operating at a focal ratio of  $f/40$ , using a camera with a  $6.9 \times 4.6$  mm sensor? The fields of view across the 6.9 and 4.6 mm dimensions of the CCD are:

$$206265 \times \frac{6.9}{12 \times 25.4 \times 40} \approx 117 \text{ arcseconds} \quad (\text{Equ. 5.5})$$

$$206265 \times \frac{4.6}{12 \times 25.4 \times 40} \approx 78 \text{ arcseconds} . \quad (\text{Equ. 5.6})$$

This field of view—117 by 78 arcseconds—would be a fine size for Jupiter and Saturn: either would float comfortably in the image. However, finding objects—even bright planets—with such a small field of view can be frustrating. In response, you will need to develop techniques for finding your target object efficiently.

One method is to insert a flip-mirror unit between the projection optics and the CCD camera. This will enable you to see a much larger field of view, and simplify locating and focusing planetary images. Another method is to attach a high-power auxiliary telescope in ring mounts so that you can align it accurately. Equip it with a reticle eyepiece that gives a magnification of 25x to 40x. Once you have it aligned, the high-power finder technique can save you lots of frustration.

Of course, alignment itself is frustrating, because *before* you align the finder, you may not be able to find anything except the Moon. Work from low power visual to high-power CCD imaging. Start with an eyepiece at prime focus and align the finder on a bright star, then put the projection system in place and reacquire the same star with an eyepiece. Align the finder. This gets you close. Now put the camera on the projection system, find the star, focus, and center the star. Tweak the alignment of the finder until it is perfect.

### 5.8.3 Recording High-Resolution Images

The camera itself is third in the triad of high-resolution imaging factors. Because moments of good seeing occur unpredictably, an effective strategy is to take a large number of images and save the best. At an outstanding site, one image in every three may be worth saving; at a typical site, only one in ten might be worth keeping. Assuming that the primary optics and projection optics have been properly matched to the chip and pixel sizes of the sensor, pursuing the “many images” strategy means that the camera must not only be capable of making the relatively short integrations required for planetary imaging, but it should also allow the observer to make a large number of images rapidly.

**Astronomical CCD Cameras.** Few of these cameras are designed for imaging bright objects with short exposures. In general, those that can do a good job are equipped with a mechanical shutter that keeps the CCD in darkness except at the moment of exposure. Because speed and agility matter, CCDs with small pixel

## Section 5.8: Lunar, Planetary, and Solar Imaging Techniques

counts tend to be more effective for planetary imaging. It is no coincidence that many of Don Parker's spectacular planetary images were taken with a Spectra-source Lynxx camera (using the TC211 CCD with its  $192 \times 165$ -pixel array) equipped with a mechanical shutter.

**Webcams.** Unlike astronomical CCD cameras, webcams are designed for making lots and lots of images. Most have CCD or CMOS-based sensors with an integral Bayer array for color imaging. The shuttering is electronic, so there are no moving parts, and the sensor electronics are designed to produce up to 30 images per second at a resolution of  $640 \times 480$  pixels (307,200 pixels). Although the Bayer array means that the effective resolution is about twice the size of the physical pixels, or about 12 microns, simultaneously recording three color channels confers a significant tactical advantage on the humble webcam.

**Digital Cameras.** For lunar imaging, digital cameras really shine. The Moon is large, bright, and loaded with fascinating high-contrast detail. Used afocally, it is easy to attain a pixel-matching image scale and still capture a large portion of the Moon's surface. Furthermore, digital cameras are designed for taking lots and lots of images. Used with eyepiece projection or a Barlow, digital SLRs are also great for lunar imaging. For planetary work, however, both types of digital cameras must capture several megapixels of black sky to get roughly 250,000 pixels worth of planet!

### 5.8.4 Making High-Resolution Images

Having determined your strategy with excellent telescope optics, a quality CCD camera, and enlarging optics that give you the desired field of view or optimum focal ratio for pixel-size matching, you must next develop a tactical approach to making good images.

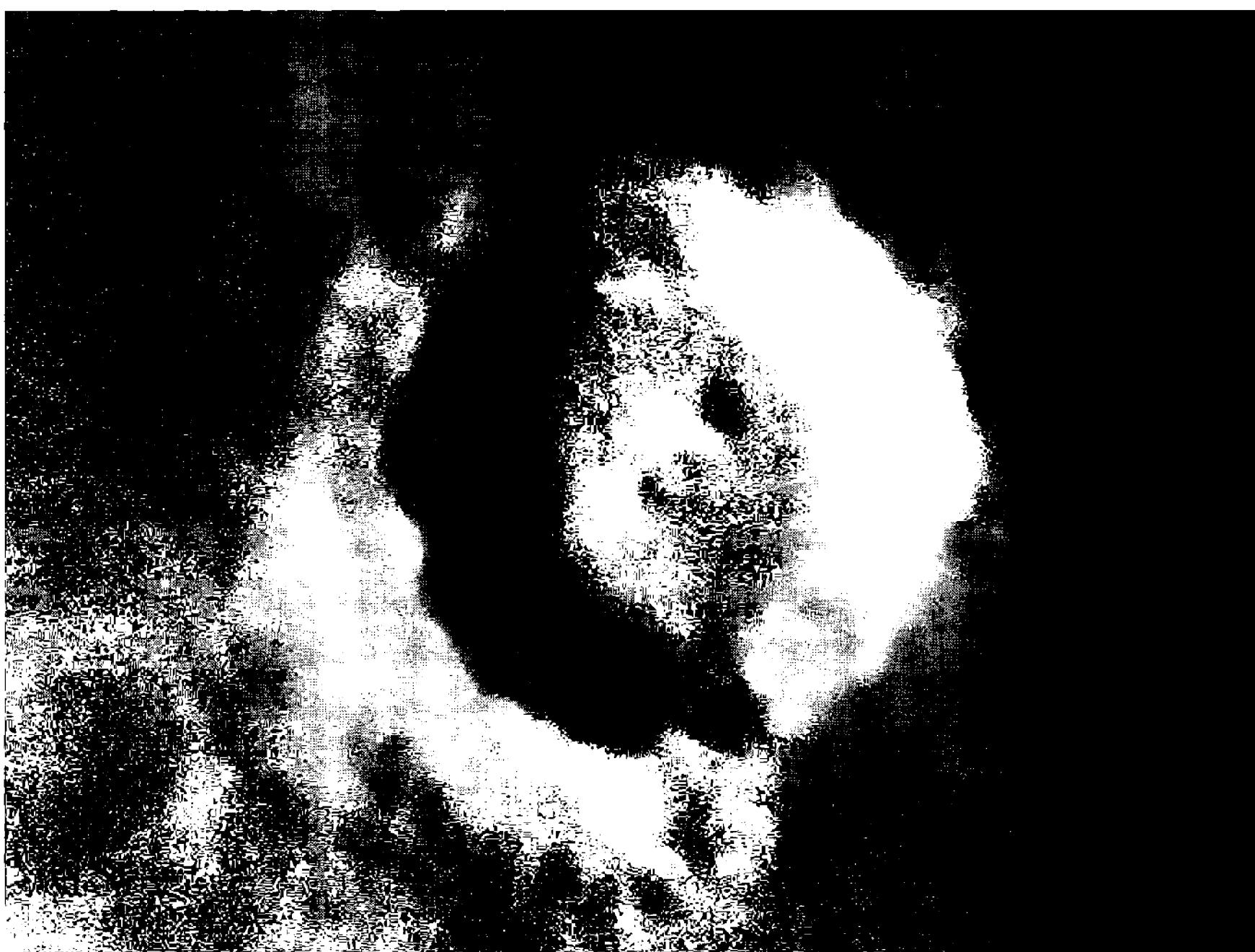
**Get good polar alignment.** Although lunar and planetary integration times are short, you will need good polar alignment, or the drift in declination over the time you gather images will drive you nuts. An error of  $1^\circ$  in polar alignment makes Jupiter move its own diameter north or south every two minutes. Do you want to chase Jupiter, or make images of it?

**Set the clock drive rate.** For planets, set the drive to sidereal rate—but for the Moon, set the drive rate to “lunar” to compensate for the Moon’s average west-to-east motion of 33 arcseconds per minute. Each time a planet drifts out of the field of view, you lose image-making opportunities.

**Clean the projection optics.** A converging cone of light at high focal ratio reveals even the smallest specks of dust as dark shadows. Clean the optics in your projection system, clean the window of your CCD camera, and take flat fields to remove the dust specks that escape your best efforts at cleaning.

**Determine the integration time.** You now need to determine the integration time. The goal is to set it so that the brightest pixels have values of at least half the full-well pixel value but less than two-thirds full-well. (For a 12-bit camera, shoot for top values between 2,000 and 2,800; for a 16-bit camera, between

## Chapter 5: Imaging Techniques



**Figure 5.16** The most important lesson to learn in lunar and planetary imaging is that raw images *always* look fuzzy. In this case study, the raw image of the lunar crater Copernicus appears soft and mushy. Note, however, that the sharp transitions from light to dark hint at detail revealed on the page opposite.

30,000 and 42,000.)

Make a test integration of 100 milliseconds; then examine the histogram. If the highest pixel values are 1,200 (out of 4,096), for example, then double the integration time. If the integration time is considerably too long, the brightest areas of the image will be saturated. Any time the highest pixel values exceed three-fourths of full-well capacity, reduce the integration until the brightest pixels are around 2,500.

Remember that astronomical CCDs are extremely sensitive, so integration times may be surprisingly short. Even with long focal ratios, unfiltered images of Jupiter and Saturn may require integrations of only around 100 milliseconds (short enough to avoid severe atmospheric smearing); and even with green and blue filtration for color imaging, perhaps only 1 second.

**Shoot lots of images. Save the good ones.** Since you can't beat the seeing, join it. Take an image, make a snap judgement, and save it if it's good. Then take another. If you shoot ten to twelve per minute and save one or two, you're doing pretty well. Don't bother to spend a lot of time studying each one: the idea is to make snap judgements and take lots. In an hour of intensive imaging, you may save 60 to 80 images.

**Expect mushy-looking images.** Raw planetary images look *terrible*. There

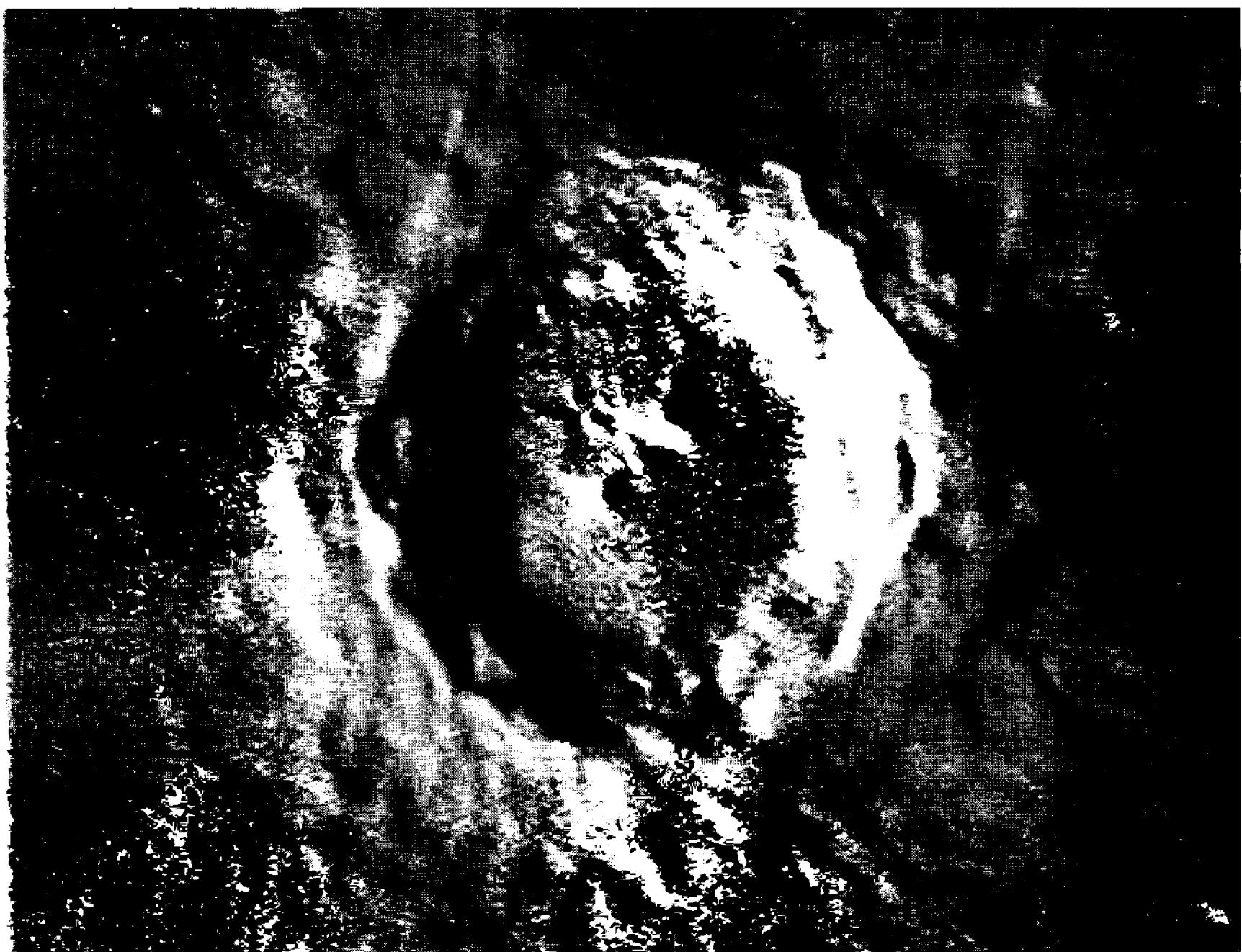


Figure 5.17 After processing, hard-to-see low-contrast image details are revealed clearly. However, compare the two images carefully, and you will see that every feature in the processed image was already present in the raw image. Image processing does not *create* detail, it only *reveals* existing detail.

is no other word for it. Even when everything is working right and one after another is popping onto the screen, raw Moon and planet images look mushy. Nine times out of ten, they come up wavered or smeared, and even the fabled “tenth image” looks pretty bad. Mushy images are completely normal. Save those that look a bit less mushy than the others.

**Touch up focus every few minutes.** Don’t focus once and assume that it’s right. Touch up the focus every few minutes. If you don’t already have an electric focuser, you’ll start to understand just how nice they can be. Electric focusing means no jiggle and shake when you focus, and a more repeatable movement from pushing a button. As soon as an image looks good, get back to taking pictures. The more pictures you take, the more good ones you’ll get. After you have saved five or six images, go back to the camera’s focus mode and touch up the focus.

Fussing with the focus may seem counterproductive, but if you touch it up frequently, you stand a better chance of hitting it right on the nose some of the time. If you actually get it perfect—and you’ll know this because you’ll start seeing a higher fraction of really sharp images—suspend the next touch up until the quality seems to degrade. The point is that with long focal ratios, in most telescopes the point of best focus will drift, and you will need to follow it by checking and correcting focus often.

## Chapter 5: Imaging Techniques

**Keep a written record.** Keep a written record of the imaging session, especially key factors such as the start time and stop time of sequences of images, eyepiece or Barlow in use, the projection distance used, the exposure times, and any other information that you will need when you process and evaluate your image the next day. Note changes in seeing, when and how you shoot your darks and flats, and all crucial settings that you make. If you get great images, you'll need this information to repeat the performance.

**Shoot dark frames.** It is tempting to get lazy and skip dark frames. Don't do it. Since you are going to enhance your planetary images to bring out low-contrast detail, you need to make them linear and get rid of the hot pixels. Shooting dark frames is easy: at some point during the session, cap the telescope and shoot a dozen dark frames. They will be almost perfectly uniform since not much thermal noise builds up in a short exposure, but proper calibration definitely improves the quality of the lunar and planetary images. When you subtract dark frames, you're also removing the bias frame, which is necessary for flat-fielding.

**Shoot flat fields.** During the imaging session, shoot a dozen flats and flat darks. An illuminated flat-field box is the best way to shoot flats, because you can make the flats without pointing the telescope away from the planet. Determine the integration time just as you did for the planetary image: set the peak pixel value between half and two-thirds well. Save eight to ten integrations as well as an equal number of flat dark frames taken with the same integration time.

Flat fields are important because they remove dust and vignetting that otherwise plague planetary images. However, remember that a flat field is good for only one physical setup. If you change the projection magnification, remove and replace, or rotate the camera, or alter anything else that changes the path light takes through the optical system; you will need a new set of flats. The only change you should allow—an unavoidable one—is a small focus change. It pays to get everything set up and running right at the beginning of the night and avoid further alterations. Make the flats about two-thirds of the way through the imaging session.

**The next day, select the best images.** By the end of an intensive imaging session, you will be tired. You may have saved as few as 20 or as many as 400 images, and you have a dozen dark frames and a set of raw flats and flat darks for calibration. Shut down the equipment, pack up, and go to bed. Leave the processing (and the emotional highs and lows that will inevitably accompany the outcome of your efforts) for the next day.

Process the images when you have the time to do it right. Make the master darks and master flat, then process all of them exactly the same way.

- **Tip:** *AIP4Win allows you to process a whole night's take of images using the Multi-Image Auto-Process tool. This takes the drudgery out of calibration. Work out a fairly aggressive standard enhancement, and then run that enhancement on every image. When you line them up and compare them, roughly one in ten will be noticeably sharper and less smeared than the others.*

## Section 5.8: Lunar, Planetary, and Solar Imaging Techniques

Processing the next day is a vital step in the “quality control” for your images. You need a night’s sleep to review in your mind what you did well and what you did poorly during the session. If you process them immediately, because of the significant element of chance in lunar and planetary imaging you can easily (and mistakenly) credit poor technique with good results, or vice versa.

**Archive your images.** Archiving digital information has never been so cheap or easy. Save the best selected raw images, the master dark and master flat, and a processed version of the best images on CD-ROMs. If you begin to produce good-quality planetary images on a regular basis, submit them to an organization such as A.L.P.O. (American Lunar and Planetary Association) or the Planetary Section of the B.A.A. (British Astronomical Association), where they will become part of our permanent record of the planets’ behavior.

### 5.8.5 Solar System Targets

The planets, the Moon, and the Sun differ in angular size, surface brightness, and spectral properties. It helps greatly to adapt your imaging techniques to accommodate the particular features of each object. For more information, and to participate in organized and directed imaging, contact organizations like A.L.P.O. and the Planetary Section of the B.A.A.

**Mercury.** Despite its high surface brightness, this fugitive planet is difficult to image because of its proximity to the Sun—but of course this only adds to the challenge! As an inferior planet, Mercury presents its full disk when it is far from Earth, and a narrow crescent when it is near. The best times for imaging are near quadrature, when the planet presents a half-illuminated disk 6 to 8 arcseconds in diameter.

To image Mercury, you can either catch it when it is high in the daytime sky, or try to image it in twilight through a long and turbulent atmospheric path. Of the two options, daytime imaging is probably best. Because its surface features—Moon-like maria and craters—are neutral in color, you can use a red filter to reduce daytime sky brightness. Its small angular diameter and high surface brightness call for long focal ratio.

**Venus.** Cloud-covered Venus has the highest surface brightness of any planet, making blooming and overexposure a significant problem for frame-transfer CCDs. The only features on its disk are cloud markings that have highest contrast in violet and ultraviolet light. Like Mercury, Venus is most readily imaged when it is near quadrature, and presents a half-illuminated disk 25 to 30 arcseconds in diameter.

To record cloud markings, you should make images through a deep blue or violet filter (such the Wratten #47) coupled with an auxiliary infrared blocking filter. For any planet but Venus working in the deep blue would pose a problem, but Venus is so bright you can expect reasonable exposure times even with a red-sensitive CCD camera.

**Mars.** You can see more features on Mars by webcam imaging than you can

## Chapter 5: Imaging Techniques

visually. This is because Mars has a high surface brightness, so you can use a long focal length with short exposures, and also because the surface features have their highest contrast in red light, where many digital sensors have their greatest sensitivity.

Mars comes to opposition about once every two years, then reaching an angular diameter between 14 and 25 arcseconds; however, significant detail can be captured any time that Mars is larger than 6 arcseconds, so the observing season is 4 to 6 months long. Mars presents its full face at opposition and shows a pronounced gibbous phase at quadrature. Because of the planet's rapid rotation, to make accurately registered tricolor images of Mars, it is necessary to obtain three good images within a 5-minute time span.

Its appearance varies enormously depending on the imaging wavelength. In blue light, the disk often appears featureless except for lighter clouds over the poles; but in red and infrared light, the dark surface features and polar caps stand out clearly. Mars observers are interested primarily in tracking clouds and dust storms and measuring the seasonal variation in the size of the polar caps.

**Jupiter.** With its large angular diameter, good surface brightness, and omnipresent and ever-changing cloud belts, Jupiter is the most rewarding of the planets to image. It ranges from 40 to 48 arcseconds in diameter at opposition, large enough to fill the frame of a TC211-based astronomical CCD camera with a 10-to 12-inch telescope.

Jupiter's clouds are fairly strongly colored; images taken through blue, green, red, and infrared filters bring out the differences clearly. The Great Red Spot is the best known colored feature on the planet, rivaled only by blue features that appear in the tropical and subtropical belts and zones.

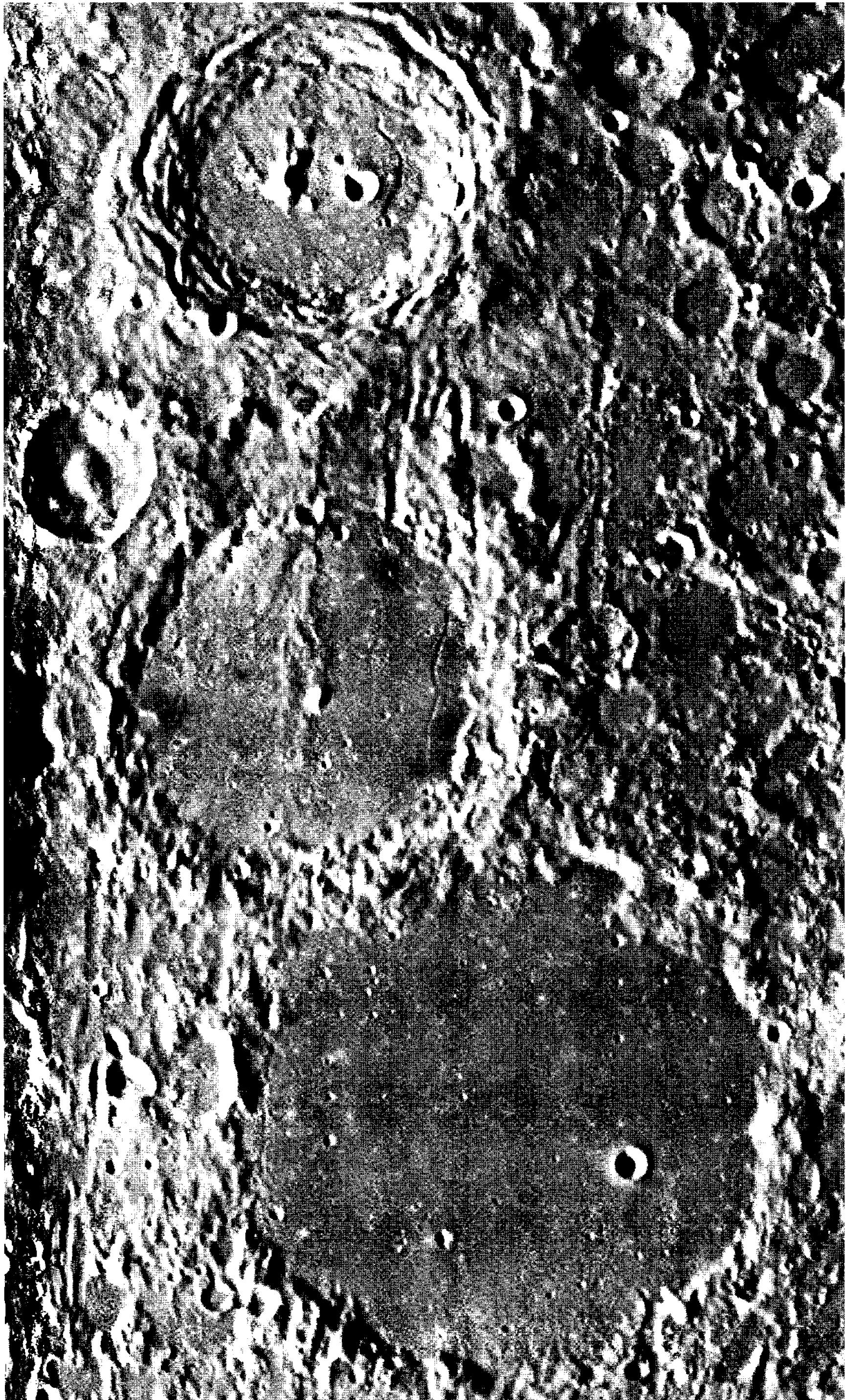
Making tricolor images of Jupiter is quite difficult because of its 9.8-hour rotation period. To prevent misregistration, the three filtered images must be made within a 2-minute time span.

**The Jovian Satellites.** Jupiter's Galilean satellites show disks large enough to record and possibly show light poles and dark equators with telescopes of 12 inches aperture and larger. The largest satellite is Ganymede; this fifth-magnitude object is about 1 arcsecond in diameter at a favorable opposition. Its surface brightness is about 50% lower than Jupiter's.

To image Jovian satellites, use an exceptionally long focal length so that you are sampling the tiny images at twice the Nyquist criterion. Take a large number of images (a few hundred) to build a sufficiently high signal-to-noise ratio for aggressive image enhancement. Select the best 20 or 30 images, register them by centroid, and stack them. Process the resulting composite. For comparison, apply the same techniques to a fifth-magnitude double star such as  $\varepsilon$  Lyrae, and a single star of fifth magnitude.

Figure 5.18   Opposite: Enlarging the image from a 10-inch  $f/9$  SCT to match the 9-micron pixels of the detector allowed critical Nyquist sampling. This image captures diffraction-limited detail in the lunar craters Ptolemaeus, Alphonsus, and Arazachel. Image by Thierry Legault.

## Section 5.8: Lunar, Planetary, and Solar Imaging Techniques



## Chapter 5: Imaging Techniques

**Saturn.** Saturn's magnificent ring system inspires heroic efforts to make outstanding images, but it is a fairly difficult target because its surface brightness is about one-third that of Jupiter, and at 18 arcseconds the disk is considerably smaller. The ring system spans around 40 arcseconds. As exposures move from fractions of a second for Jupiter to full seconds for Saturn, seeing becomes a relatively more important factor in imaging the ringed planet.

Saturn's atmosphere is less colorful and has lower contrasts than Jupiter's. Every few decades, however, bright white clouds well up suddenly and then fade away over several months. In the ring system, the Cassini Division is about 0.4 arcseconds wide, and shows up as a dark band even when it is unresolved. The next time Saturn's rings are well presented, it would be interesting to attempt imaging the ring spokes discovered by the Voyager spacecraft.

**Uranus and Neptune.** These two gas giants are roughly 4 and 2 arcseconds diameter at opposition, and sixth and eighth magnitude, respectively. It is easier to image the disk of Uranus than Ganymede because the planet is much larger; but Neptune is only a little larger and considerably fainter. The multiple-image technique (described for Ganymede) should reveal their disks. Voyager showed Uranus as having a deep, featureless haze layer, but more recent Earth-based images have shown cloud belts. At present we see Uranus at an oblique angle. Neptune has both light and dark cloud features, but it is doubtful they could be imaged on the tiny disk.

**Lunar Imaging.** The lunar surface presents a myriad of wonderful details on a body with high surface brightness; it is an excellent target for high-resolution CCD imaging. Because the lunar surface has very little color, the primary uses for filters in lunar imaging are to remove residual chromatic aberration that might be present in a refracting objective or enlarging optics, or to reduce the light of this sometimes-too-bright subject.

However, different geological units on the lunar surface do have subtly different colors. It would be an interesting project to make matched images of Mare Imbrium, for example, through blue, green, red, and infrared filters, and combine these to make enhanced color images showing lava flows of different ages.

The surface brightness of the Moon varies with phase, and it varies across the illuminated disk. Its surface brightness is roughly 20 times greater at full than as a three-day-old crescent; and in the gibbous phase, the areas in full Sun with no shadows are about 10 times brighter than the shadow-rich regions under slanting illumination on the terminator.

One significant technical problem in lunar imaging is that the Moon has such a large angular diameter that stray Moonlight can scatter in the projection eyepiece and projection tubes, causing field flooding and other stray-light problems. A solution for this difficulty is to let through only a small section of the lunar image on the optical axis by placing a small mask made of brass shim stock at the focus of the projecting eyepiece. The opening in the mask should be large enough to illuminate the CCD fully, but small enough to prevent additional Moonlight from entering the optical system.

**Solar Imaging.** Imaging the Sun is similar to imaging the Moon, except that it is a million times brighter. To reduce the Sun’s brightness, place a full-aperture aluminized glass or Mylar filter over the aperture. Standard filters made for visual solar observing are suitable. These have a neutral density between 4.5 and 5.0 (ND=4.5 to 5.0). Like the Moon, the solar surface is virtually colorless, so color filters are not needed.

Noteworthy solar features are sunspots, granulation, regions of plage (spidery bright regions) near the limb, and rare white-light solar flares. Because these features are dynamic on time-scales of a few hours, it would be worth the effort of making a few hundred images of a small sunspot in the course of a day and replaying the sequence as a movie to view the changes. CCD cameras can also be used with narrow-band H $\alpha$  filters to capture images of prominences, monochromatic solar surface features, and solar flares.

Whenever you point a telescope at or near the Sun, you risk damaging it and/or your eyesight. Pay close attention to precautions such as capping the finder telescope and making sure that solar filters are securely attached.

## 5.9 The Role of “Technique”

Equipment does not make images: people do. What people—what you, the astro-imager—bring to the telescope is the skill and knowledge to use equipment effectively. You don’t need to wait long before you’ll hear someone new to digital imaging whining the complaint: “These images are no good. This telescope/camera/software is a piece of %\*#@#.” What you’re hearing is a frustrated person attempting to place blame on some inanimate objects.

It is admittedly sometimes difficult to analyze an imaging challenge as a problem to solve—but seeking the cause of problems leads to solutions, whereas blaming the equipment does not. (Admittedly, sometimes equipment does leave much to be desired, and a systematic problem-solving approach will reveal this fact.)

In its broad sense, *technique* means “a manner in which technical details are treated.” How should you treat technical details? First, recognize that every such detail matters. Leave out one crucial “detail”—such as precise focus—and nothing else you do will give you good images. Second, look for causes. Good imaging is based on cause and effect. When something goes awry, ask questions. What might have caused this? Whenever you’re tempted to say, “Stuff happens,” you need instead to say, “Stuff happens because....”

Third, work to understand what it is that you’re doing. In systems engineering, there’s a concept called *solution space*. Solutions are high spots, failures are low spots. If you wander around changing what you do bit-by-bit in response to your results, your images will get better because you’ll be moving toward a peak in solution space. Once you’re on a local peak, small changes will make your images get worse. In short, you’ll be trapped in techniques that work—but not necessarily the best techniques.

## Chapter 5: Imaging Techniques

If you understand what you’re doing, you may be able to look around to see whether the local peak you’re standing on is not the highest peak. Consider this example: A few years ago, it looked like the best way to shoot planetary images was to make lots of short exposures with a 16-bit camera and select the very best image from hundreds taken. To make color images, you made three images with color filters. It was a local peak in solution space, and a rather high peak at that.

Then along came webcams with their tacky little 8-bit images. Even the best webcam images look weak and noisy. But registering and stacking a few hundred of those noisy little images built a high signal-to-noise ratio, and selecting the best hundred from a thousand captured the best moments of seeing. The 8-bit webcam technique peak is higher in solution space than the single best 16-bit image CCD technique—and it’s easier to use the webcam, too.

Digital imaging is a dynamic subject. Every time you turn around, there’s a new device entering the market that just might be the next shining beacon on a yet-higher peak. But not always. That webcams are great for imaging bright objects with short exposures hardly guarantees that they’ll be good for faint targets—summing readout noise over hundreds of photon-starved images simply cannot beat longer integrations. The numbers demand the result.

As you develop your imaging skills, you face a multitude of technical challenges—and solving challenges is what makes imaging so much fun. Pay attention to the details, look for causes, and survey the entire solution space.

---

# 6 Image Calibration

---

In the previous chapter, we outlined how to make support frames—darks, flats, and bias frames—for your images. In this chapter, we examine the nitty-gritty details of image calibration and explain what it is and why it is necessary.

Images straight from the CCD carry a number of unwanted “signals.” The goal of calibration is to correct the raw image so that the calibrated (but otherwise unprocessed) image contains pixel values that accurately portray the intensity of light that fell on the CCD during exposure.

The unwanted signals in a raw CCD image include two additive components and one multiplicative component. The additive components are a voltage offset, or bias, from zero volts; and a signal generated by thermal emission of electrons that grows linearly with exposure time. The multiplicative error arises because photosites have differing sensitivities to light. Calibration involves removing the bias, subtracting the dark current, and dividing the image by a map of photosite sensitivity.

Performing these corrections exacts a toll on the CCD user: the bias, dark current, and sensitivity map of the CCD must be determined. Ideally, these should be determined one or more times during each observing session, but in practice the calibration regimen may be simplified.

This chapter examines the behavior of the unwanted signals found in raw CCD images and suggests three observational protocols—basic, standard, and advanced—for calibrating CCD images. The word “protocol” means a plan or prescription for accomplishing a task. In this context, calibration protocols are recommendations as to how an observer using an astronomical CCD camera can make and calibrate satisfactory images.

- The **basic protocol** meets the needs of many observers, removing bias and dark current for a relatively small investment in observing time. During an observing session, from time to time, the observer takes dark frames having the same integration time as the images. To calibrate the latter, the observer combines dark frames to reduce noise and subtracts them from the raw CCD images. Basic calibration is suitable for search

## Chapter 6: Image Calibration

and survey images, and often meets the needs of observers who are just getting into CCD observing.

- The **standard protocol** adds to the basic protocol a correction to compensate for the CCD's photosite-to-photosite sensitivity variation. In addition to taking dark frames, at some point during the night, the observer shoots flat-field frames either from the twilight sky or from a low-intensity illuminated box or panel. Standard calibration produces high-quality monochrome or color images suitable for display or publication, as well as images that can be used for precise astrometry or photometry.
  - The **advanced protocol** is a prescription for producing images of “research” quality, appropriate for use with high-performance scientific-grade CCDs. In addition to dark frames and flat-field frames, the observer obtains bias frames that permit more precise removal of the bias and dark current than does the standard protocol. Advanced calibration methods are also more flexible and forgiving than the basic and standard protocols, and thus often repay the additional time required to obtain the required bias-frame data.
- **Tip:** *AIP4Win supports the basic, standard, and advanced calibration protocols. As an observer, remember that calibration frames must be taken when you make your observations.*

From the foregoing, it should be clear that the basic calibration protocol requires the least work. All that's necessary is that during imaging, you shoot an adequate number of dark frames. The biggest constraint is that the integration time you use for your dark frames must be the same as the integration time you are using for your images. Since most observers settle on a convenient integration time, such as 60 seconds, there is no difficulty in making dark frames with the same integration time.

Standard calibration is a step more sophisticated because it corrects for variations in CCD sensitivity and subtracts the dark current. In addition to shooting an adequate number of dark frames, the observer must also shoot a set of flat-field frames and matching flat-field dark frames during the night. Flat-fielding corrects for the dark corners caused by vignetting and for the little dark donuts caused by dust as well as for variations in CCD sensitivity. Most observers will probably find that the standard calibration protocol is entirely adequate.

A step more complex than the standard protocol is the advanced calibration protocol. Within an already crowded shooting schedule, observers must find time to shoot an adequate number of bias frames. However, by doing so, they gain a significant advantage: when a bias frame is subtracted from a dark frame, the resulting image (called a *thermal frame*) contains only dark current. Because dark current increases linearly with time, a thermal frame can be scaled to match integrations of any length. Thus the observer can shoot a set of dark frames with 300-second integrations. During calibration, a master bias frame is subtracted from the

dark frame to produce scalable thermal data. This can be scaled and then subtracted from integrations of 5 seconds, 30 seconds, 150 seconds—any integration shorter than the integration used for the dark frame.

- **Tip:** *For optimum results, decide in advance which calibration protocol you will use; then be sure to obtain the necessary calibration frames during the observing session. Most observers choose either the basic or standard calibration protocol.*

## 6.1 What's in a CCD Image?

To better understand how calibration works, we begin by reviewing how data are created by a CCD camera. The different calibration frames allow us to remove and correct peculiarities in the CCD camera, so that our images are accurate recordings of the light that fell on the CCD chip.

However, in a working CCD, the intensity of the light that falls on the sensor is wrapped in a succession of other effects, like the layers of an onion. At the very center is the information that we want, and surrounding it are layers of information that we do not want. In this section, we'll build the onion starting at the center and adding layers, and then in the sections that follow, we'll show that calibration is a process of peeling away layers to reach the core.

### 6.1.1 Photon Flux

Recall that a CCD chip consists of a rectangular array of light-sensitive regions called photosites. When an astronomer places a CCD at the focus of a telescope, the goal is to measure the flux of photons incident on each photosite,  $I_{x,y}$ . *The flux is the rate at which photons arrive; this flux is the quantity that we want to measure.* We cannot measure flux directly. Instead, we must count how many photons accumulate during an interval  $t$ , called the integration time or exposure time.

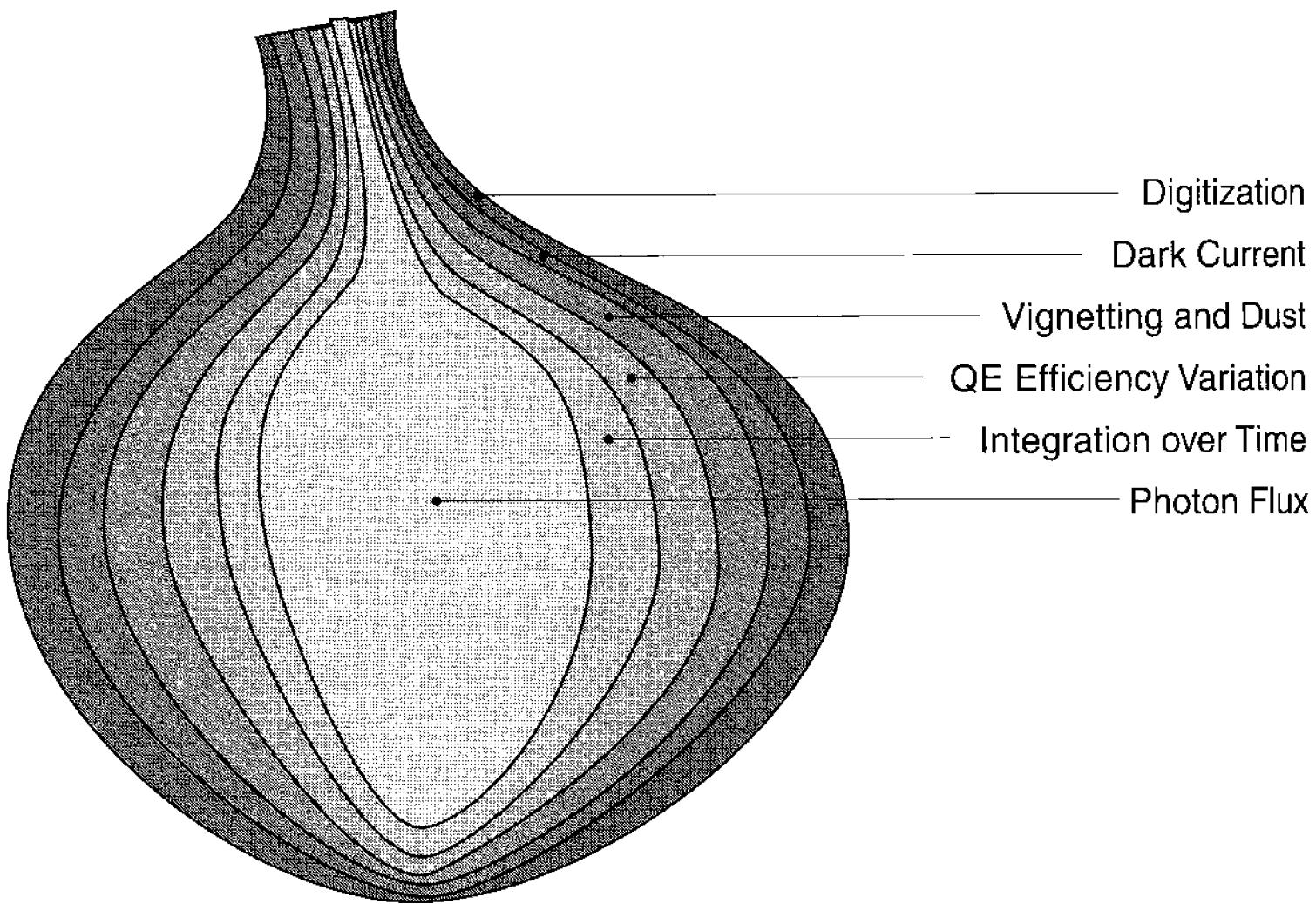
Upon exposure to incoming photons, some number of free electrons proportional to the incoming number of photons accumulates at each photosite. The electrons are held in place by electric charges applied to the gate structure by the CCD control electronics, until the exposure is complete. Only then do we read out the image. The image is thus always an accumulation over time, not an instantaneous measure of the flux.

The photon flux lies at the center of our metaphoric onion. A CCD image is always a flux of photons integrated over the integration time  $t$ . The total number of photons that impinge on the CCD during the integration time is:

$$P_{x,y} = \int_0^t I_{x,y} dt , \quad (\text{Equ. 6.1})$$

where  $P_{x,y}$  is the number of photons received. The  $x, y$  subscripts mean that  $I_{x,y}$  and  $P_{x,y}$  refer to the  $x^{\text{th}}$  photosite in the  $y^{\text{th}}$  row on the CCD. The subscripts allow

## Chapter 6: Image Calibration



**Figure 6.1** CCD information is layered like an onion, with the data that you really want—the flux of photons—hidden away in the innermost layer. To retrieve the photon flux, you must peel away unwanted signals and factors that influence the signal from the CCD. This process is called image calibration.

us to use a single equation to describe the behavior of every photosite on the sensor.

The equation above says that the total number of photons captured during an integration (i.e., exposure) is the total number of photons that arrive during the integration time. The longer the integration time, the greater the number of photons that accumulate. To make quantitative measurements of brightness, we need to know the length of the integration so that we can determine an average flux (i.e., the average rate at which photons arrive).

To find the flux from the accumulated photon count  $P_{x,y}$  and the integration time  $t$ , we divide the photon count by the integration time:

$$\bar{I}_{x,y} = \frac{P_{x,y}}{t}. \quad (\text{Equ. 6.2})$$

The quantity  $P_{x,y}$  describes the image captured by the CCD. A different total photon count accumulates in each photosite; that is,  $P_{x,y}$  is different at each location  $(x, y)$  on the CCD. Furthermore, because the instantaneous flux  $I_{x,y}$  may have varied during the integration, we can only determine the *average* flux,  $\bar{I}_{x,y}$ , during that interval. (The bar over the  $I$  means it is an average value.) Any variation in the flux that occurs on a timescale less than the integration time is averaged out.

In Chapter 2 we described how and why *Poisson noise* is an intrinsic part of the flux of photons. Because photons arrive randomly, during any particular sub-interval, the actual number of photons may be greater or lower than the mean rate.

## Section 6.1: What's in a CCD Image?

The rule describing the expected departure from the average is extremely simple: if the average rate is  $\bar{P}_{x,y}$ , then the standard deviation,  $\sigma$ , from that rate is:

$$\sigma_P = \sqrt{\bar{P}_{x,y}}.$$

The quantity  $\sqrt{\bar{P}_{x,y}}$  describes the width of the distribution curve, and is called the standard deviation. We usually designate it as  $\sigma_P$  (the Greek letter sigma stands for “standard,” with a subscript P to remind us that we’re talking about photons).

In everyday situations, we have so many photons we don’t worry about photon noise. Outdoors on a sunny day, our eyes receive billions of photons every second—but astronomy deals with faint objects. To collect enough photons to make a good image, the integration time may need to be minutes or even hours long.

The very fact that we must integrate for some length of time is the first layer of the onion: integration can obscure real variations in the flux of photons from a source. When we examine an image after an integration with a CCD camera, we only know the integrated total number of photons collected by each photosite while the CCD was exposed to starlight. The photon sample is prone to random variation, but if the flux really did vary during a long integration, the image retains no sign of it—all we have is the total.

### 6.1.2 Nonuniformity

Although modern CCDs are extremely uniform, there are nonetheless differences on the order of 1% from one photosite to the next between number of incoming photons and the number of electrons liberated. Thus, even as the photon signal is converted into electrons, an effect occurs that we will later need to correct. When a uniform flood of photons falls on a CCD, different photosites produce different numbers of electrons. Nonuniformity is thus the next layer in the “image onion.”

The efficiency with which a photosite  $(x, y)$  converts photons into free electrons is called its *quantum efficiency*,  $Q_{x,y}$ , and the total number of electrons created at the photosite,  $E_{x,y}$ , during an integration is:

$$E_{x,y} = t Q_{x,y} I_{x,y}. \quad (\text{Equ. 6.3})$$

Notice that the quantum efficiency is multiplicative—that is, quantum efficiency is not added to the signal; rather, the number of photons is multiplied by the quantum efficiency. When the time comes to “peel away” variation in the quantum efficiency in calibration, we will need to divide rather than subtract.

Quantum efficiency is itself a statistical process, an average over a large number of photons. Within a sample, the conversion of photons into free electrons itself varies randomly. If an average flux of 100 electrons per second falls on a photosite, then a single 1-second sample would contain  $100 \pm 10$  photons 68% of the time. If the average quantum efficiency is  $\bar{Q}_{x,y}$ , then the number of electrons produced is

$$100 \bar{Q}_{x,y} \pm \sqrt{100 \bar{Q}_{x,y}}.$$

## Chapter 6: Image Calibration

If, for example, the average quantum efficiency were 40%, then we would measure  $40 \pm 6.3$  electrons per second. When we measure the quantum efficiency, we want to do so with as many photons as possible to obtain an accurate measurement of the quantum efficiency of each photosite on the CCD.

Vignetting in the optical system is familiar to most observers. Vignetting occurs because the optical system does not illuminate the outer edges of the CCD as fully as it does the center. A similar problem, dust shadowing, is caused by bits of dust on filters and windows that cast shadows on the sensor. Vignetting and shadowing vary from pixel to pixel, and are multiplicative factors that influence the total number of photoelectrons produced at each photosite. We designate dust shadowing and vignetting by the term  $V_{x,y}$ :

$$E_{x,y} = tV_{x,y}Q_{x,y}I_{x,y}. \quad (\text{Eqn. 6.4})$$

Nonuniformities thus add two layers to the image onion: variation in quantum efficiency and variation in illumination. For the record, it is worth noting that quantum efficiency also varies with the wavelength of the light; and in many optical systems, the vignetting depends on which filters are in front of the CCD. Both of these nonuniformities are multiplicative, and with the proper calibration techniques, both layers can be peeled away in one operation called flat-fielding.

### 6.1.3 Dark Current

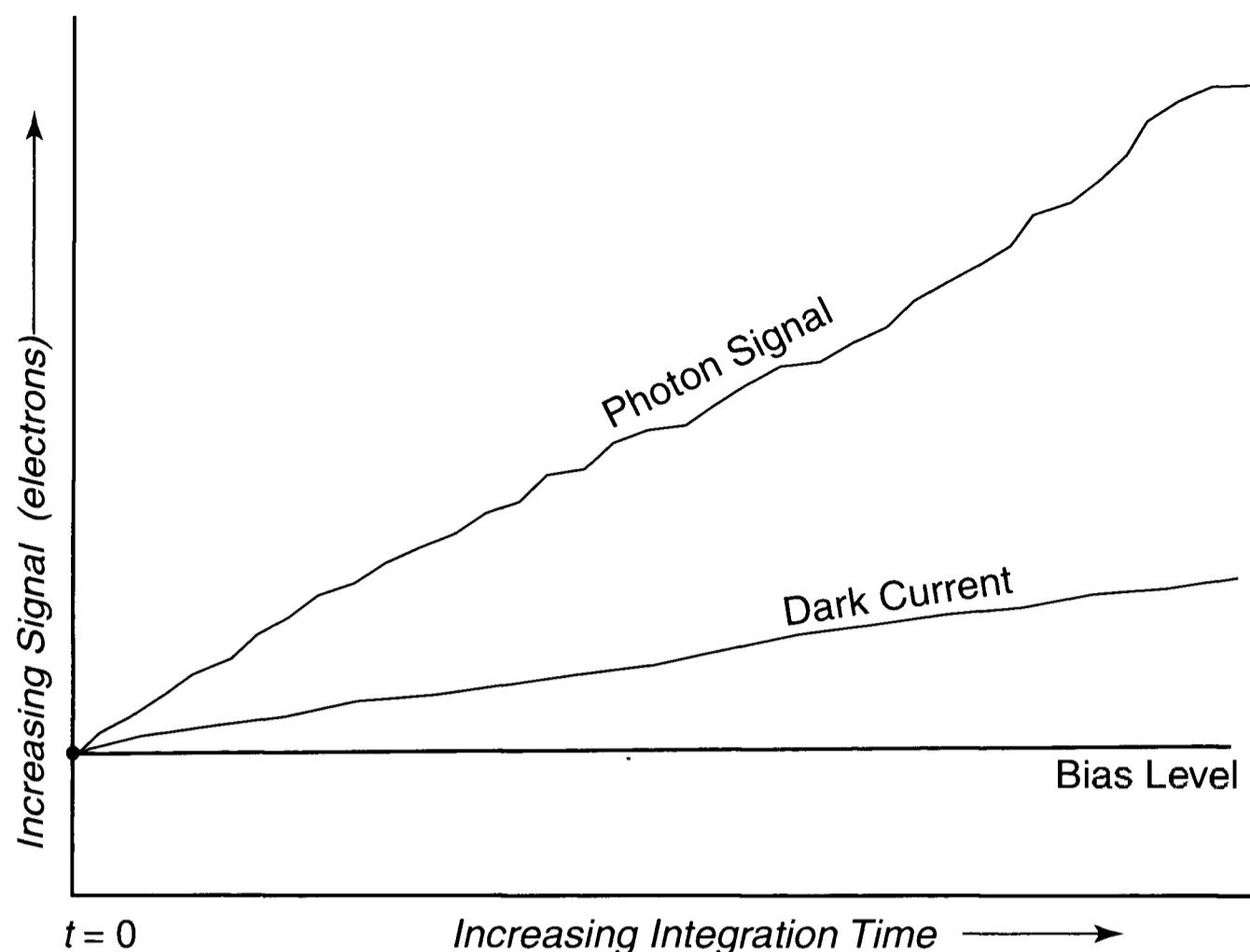
Thermal agitation of atoms in the silicon substrate of the CCD frees electrons. This process goes on even when it is in total darkness; hence, the steady creation of free electrons is called dark current. However, dark current also occurs when light falls on the CCD. Regardless of the illumination, the dark current is  $d_{x,y}$  for photosite  $(x, y)$ . The total number of thermal electrons that accumulate during an integration,  $T_{x,y}$ , equals the rate at which they accumulate (the dark current) times the integration time  $t$ :

$$T_{x,y} = td_{x,y}. \quad (\text{Eqn. 6.5})$$

Thermal electrons add another layer to our metaphorical onion, an accumulation of signal that comes not from light but from processes occurring in the bulk silicon and the interface between the silicon and gate structures on the CCD.

At any given temperature, the rate at which electrons are freed is constant; but for each 6 degrees Celsius that the temperature rises, the dark current approximately doubles. This is because the thermal agitation of the atoms increases with temperature. Because they operate at low temperature, the number of thermal electrons generated by a typical amateur astronomer's CCD lies in the range of one to a few hundred electrons per second—low enough to allow integrations of many minutes.

Depending on the design, construction, and quality of a CCD camera, the dark current may be nearly the same from one photosite to the next; or it may vary greatly. In commercial-grade models the dark current is large but fairly uniform. However, when commercial-grade CCDs are operated in low-dark-current or



**Figure 6.2** During an integration with a CCD, the bias remains constant, the dark current increases, and photoelectrons accumulate. At the end of the integration, you only know the total signal. To find the average photon flux, you must subtract both the bias level and the dark current from the total signal.

charge-inversion mode, they display totally different behavior: the dark current at most photosites is small, but a small fraction of a percent of the photosites (called “hot pixels”) produce from 100 to 1000 times greater dark current than does the average photosite. This “hot-pixel” behavior is typical of CCD cameras made for amateur astronomy. In scientific-grade models the dark current is both small and uniform; and because they are used at much lower temperatures, there are very few or no hot pixels.

#### 6.1.4 Zero-Point Bias

Most CCD cameras are adjusted to give an output somewhat above zero when the signal from the detector is zero. When the CCD is read out, the electrons that have accumulated in each photosite are moved to its detection node, where a charge-sensitive amplifier that is part of the CCD produces a voltage proportional to the number of electrons.

The voltage output might be  $-3.5$  volts when there are no electrons, and  $-4.1$  volts when the charge detection node is saturated with electrons. This output voltage goes to an inverting amplifier which amplifies and changes the sign of the voltage, so that  $-3.5$  volts (no signal) produces an output of  $0.2$  volts, and  $-4.1$  volts input (full saturation) produces an output of  $9.8$  volts. Many analog-to-digi-

## Chapter 6: Image Calibration

tal converters produce a binary output of zero for zero volts on the converter. Depending on the resolution of the converter, a 10-volt input will produce a maximum binary output of 4095 (12 bits), 16387 (14 bits), or 65535 (16 bits). A zero-charge signal from the CCD, therefore, produces a small positive binary output on the converter. A small positive bias is useful because it prevents underflow when the signal is digitized. Thus, each pixel in the image contains a measurement of the system electrical offset, which is called the bias signal,  $B_{x,y}$ .

### 6.1.5 Quantization

The outermost layer of the onion arises from digitizing the analog signal from the inverting amplifier. As we saw in the preceding section, this step converts the units of measurement from an output voltage into integer *analog to digital units*, or ADUs. The pixel values in an image are measured in ADUs.

Signals from the CCD are converted from their original units (electrons or volts) into ADUs by way of the conversion factor  $g$ , which has units of electrons per ADU. The conversion factor varies from around one electron per ADU for a scientific-grade CCD camera with a 16-bit analog-to-digital converter to several hundred electrons per ADU for inexpensive 8-bit CCD camera systems.

Quantization imposes a “granulation” on how accurately we know the signal coming out of the CCD. The smallest difference that an observer can detect in the quantized signal is one ADU. If the conversion factor is 10 electrons per ADU, then the smallest difference between two signals that an observer can resolve is one ADU, or 10 electrons. CCD camera designers usually set the gain so that observers can detect about half the readout noise, to minimize the loss of information. This outermost layer of the onion imposes a restriction on the accuracy with which we can measure a signal from the CCD.

In addition to limiting the resolution of the signal, because a range of input signals can produce the same output, quantization introduces an uncertainty equivalent to random noise. The quantization noise  $\sigma_{QN}$  is:

$$\sigma_{QN} = \frac{g}{\sqrt{12}} \text{ [electrons]}, \quad (\text{Equ. 6.6})$$

where  $g$  is the conversion factor (the number of electrons equivalent to one ADU). When the conversion factor is 10 electrons per ADU, quantization adds another random noise of  $\pm 2.9$  electrons.

### 6.1.6 Calibration is Peeling the Image Onion

In review, every CCD image consists of many effects piled atop of one another like the layers of an onion. At the core is the flux of photons falling on each photosite. Over the core lie these layers:

- Integration of flux, hiding short term changes in brightness.
- Nonuniform quantum efficiency of photosites.
- Vignetting, so that photosites receive unequal illumination.

- Dark current, which adds a false signal.
- Bias voltage, displacing the zero point of the output.
- Quantization, converting the CCD’s analog output into ADUs.

Calibration is the process of peeling away the layers that surround the core of information that we seek. In the sections that follow, you will see how the onion must be peeled from the outside in—working in reverse order—to reach the core. An image is calibrated by removing the bias first, then subtracting the dark current, and last by dividing out the effects of vignetting and nonuniformity. The following section describes three specialized types of images, called calibration frames or support frames, that enable us to peel away unwanted effects to reach the accumulated flux of photons.

## 6.2 Calibration Frames

As astronomers, we want to know the photon flux—that is, the rate at which photons fall on our CCD. Yet a raw CCD image contains the signal we want mingled with a bias voltage, a dark current, and nonuniform photosite sensitivity. To recover the photon flux, we take additional images that allow us to subtract the additive values and divide out the multiplicative factors. These additional images—support frames—include dark frames, flat-field frames, and bias frames.

Support frames provide a record of the CCD’s peculiarities, allowing us to peel off layers of the onion one by one. For best results they should be taken at the telescope immediately before, during, or immediately after a set of astronomical images. This insures that the image frames and support frames match.

Although it is possible to use “ordinary” bias, dark, or flat-field frames in calibration, it is better to use “master” frames made by combining multiple bias, dark, or flat-field frames into a support frame that has less random noise than a single one. Although making master frames takes more observing time, the resulting calibrated images are significantly better.

In this section, we explore each type of support frame, the specific types of information that they capture, and how we can use that information to recover a celestial image that is corrected for bias, dark current, and nonuniformities.

### 6.2.1 Bias Frames

Astronomers make bias frames to capture the bias level. Because the CCD is in total darkness and the integration time is zero, the bias level,  $B_{x,y}$ , should be exactly the same for every pixel on the CCD since (in theory) neither photoelectrons nor thermal electrons were generated.

However, the bias level may fluctuate because of things that happen every time the CCD is read out or because extraneous signals are added to the bias level. Repetitive events during readout form a fixed pattern in the bias,  $(B_{FP})_{x,y}$ , while extraneous events such as interference from a nearby computer monitor are unpat-

## Chapter 6: Image Calibration

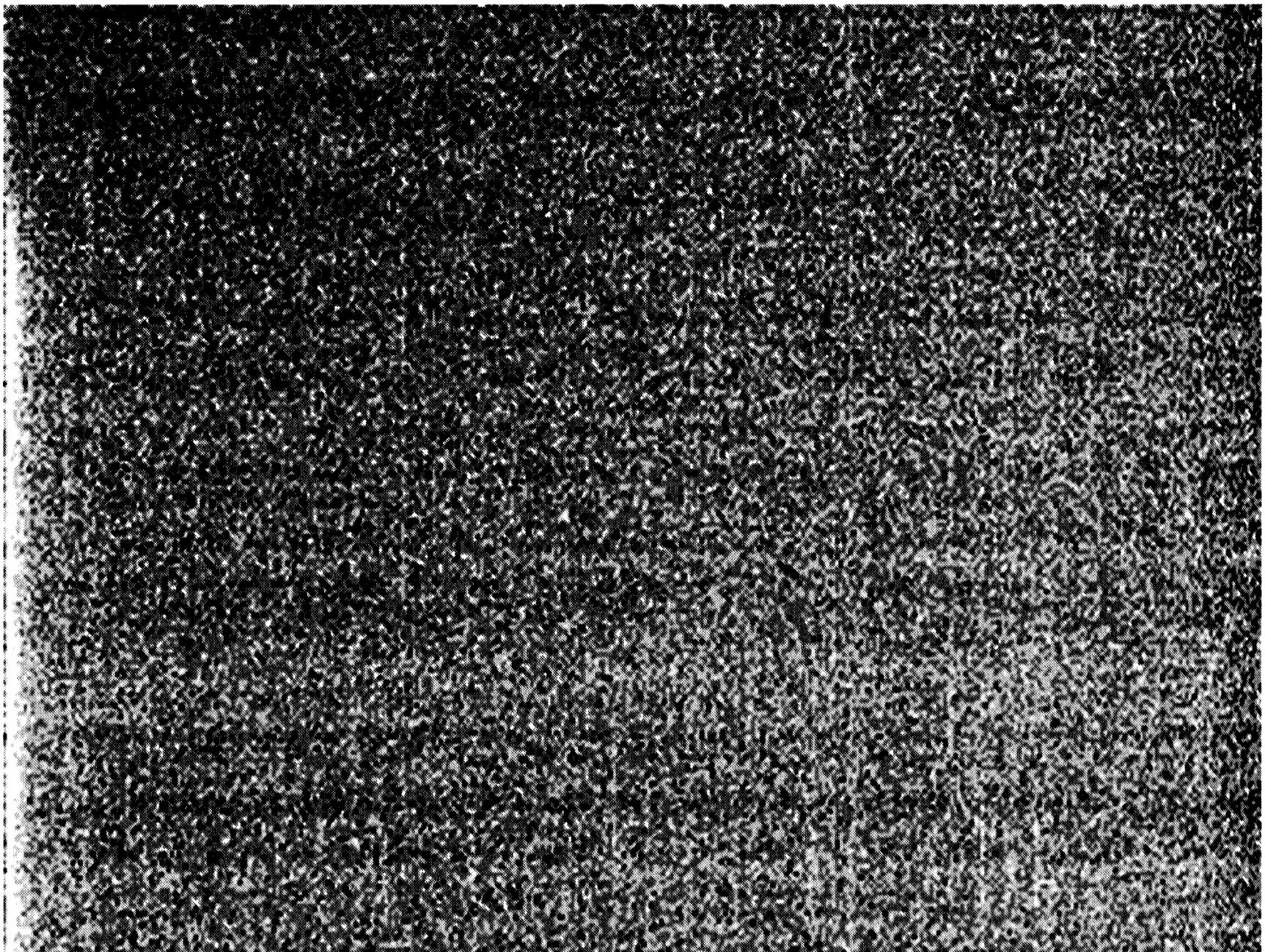


Figure 6.3 In this bias frame, a pixel value of 99 appears black and a pixel value of 102 appears white. The average pixel value is 100.00 ADU, and the standard deviation is 0.56 ADU. Fixed-pattern bias is visible at the lower left, and low-level horizontal and vertical pattern noise are also apparent.

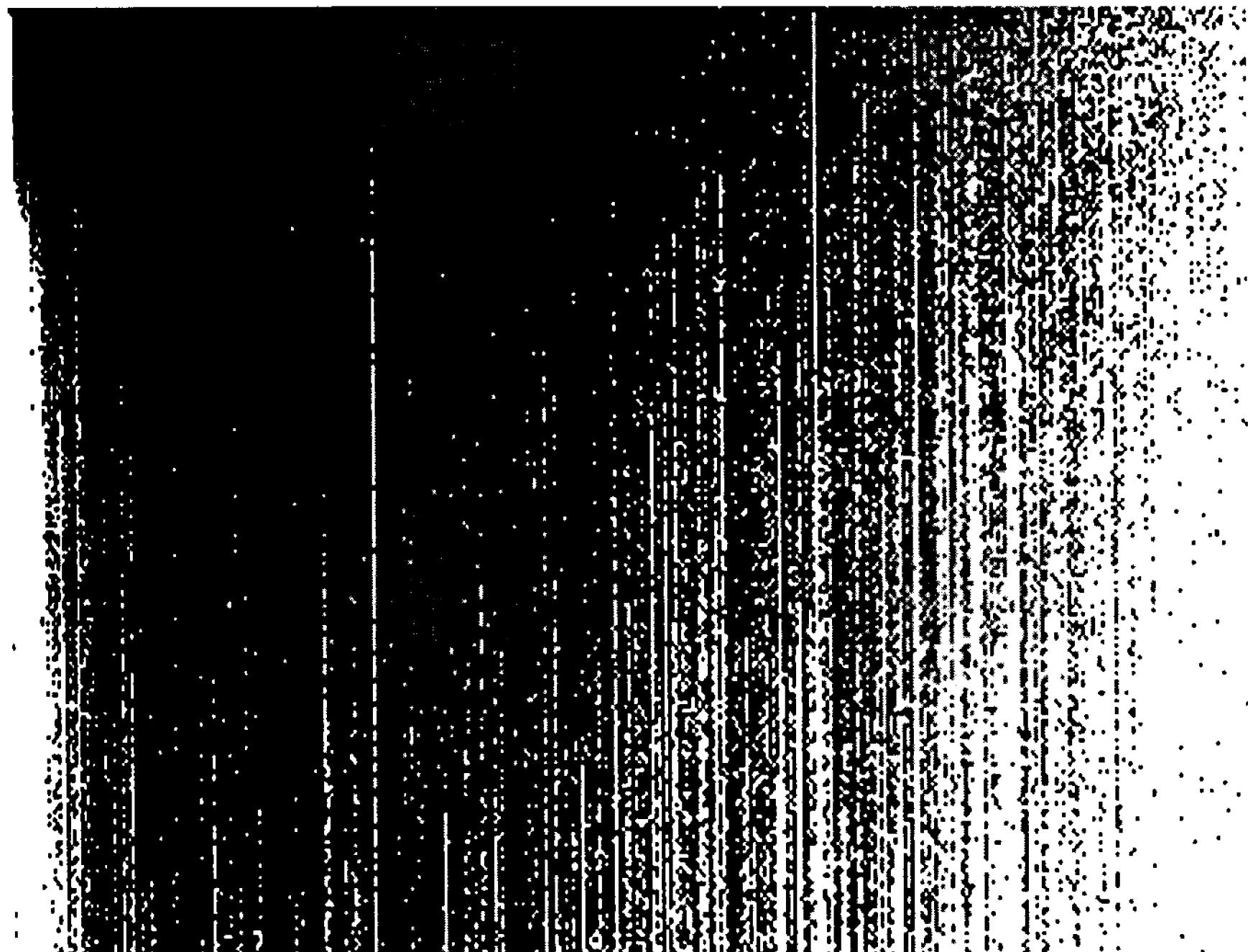
terned,  $(\sigma_{UP})_{x,y}$ , and are considered noise because they appear randomly at a given photosite.

Fixed-pattern bias may result from light-emitting circuit elements on the CCD or the clocking circuits in its control electronics. (In this context, the word “fixed” means unchanging.) A fixed pattern in the bias frame is not a problem because it is the same in every bias frame, and can therefore be subtracted out.

Unpatterned events originate in power supplies, nearby electronics, motors, and radio-frequency interference from computers and monitors. These unwanted contributions are usually very small. Nonetheless, you should take and examine bias frames at every observing session simply to check that the bias frames are indeed free of spurious signals. If such signals are present, the condition causing them should be diagnosed and corrected.

In addition, the charge detection node on the CCD that converts electrons into an output voltage adds a random component to the bias called readout noise,  $\sigma_{RO}$ . Because readout noise has a Gaussian distribution, the pixel values in the bias frame will display a Gaussian distribution around the bias value.

Finally, even when the integration time is zero, it takes several seconds to read out the CCD, and a small amount of dark current accumulates in that time. The readout time,  $t_{RO}$ , short near the CCD’s serial register, is relatively long on



**Figure 6.4** This image is the median of 64 bias frames, displayed with black = 99.8 and white = 101.4. With the exception of the artifact at lower left, every pixel value is either 100 or 101. “Hot” photosites generate a few thermal electrons during the time it takes to read out the CCD, producing the vertical streaks.

the side of the CCD farthest from the serial register; so there is a gradient in the number of thermal electrons,  $d_{x,y}$ , from the bottom to the top of the image, plus a random variation,  $\sigma_{TE}$ , in the electron count.

Thus, the bias frame, the simplest type of image, with no light on the CCD and zero integration time, contains the following elements:

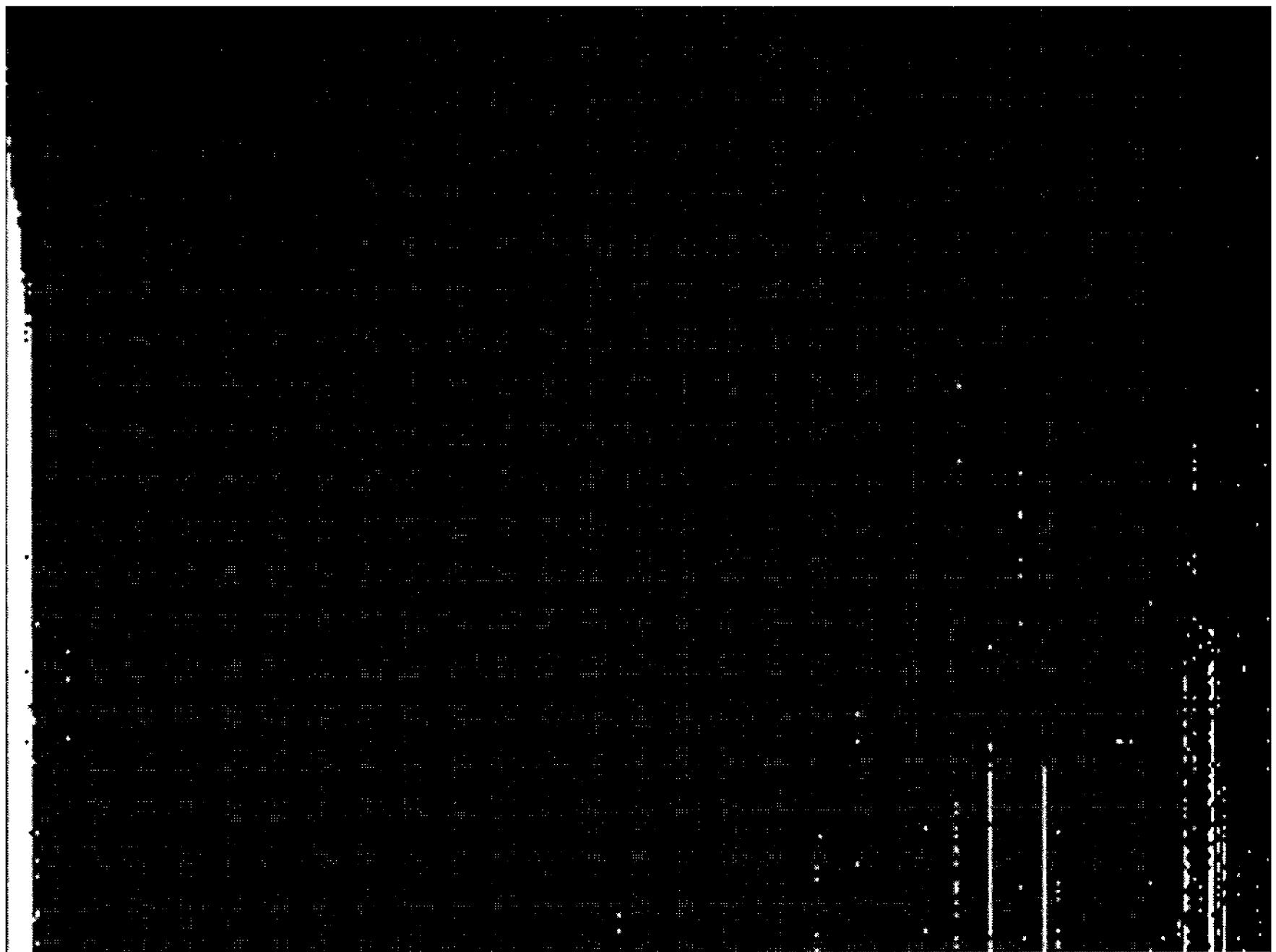
$$\langle \text{BIAS} \rangle_{x,y} = \frac{1}{g} \{ B_{x,y} + (B_{FP})_{x,y} + (\sigma_{UP})_{x,y} + \sigma_{RO} + t_{RO} d_{x,y} + \sigma_{TE} \}. \quad (\text{Equ. 6.7})$$

Recall that the conversion factor  $g$  has units of electrons per ADU; so when the bias, fixed-pattern bias, unpatterned interference, readout noise, and dark current are measured in electrons, the bias frame is expressed in ADUs.

### 6.2.1.1 Using a Single Bias Value

In a healthy CCD camera, the bias value  $B_{x,y}$  is by far the most important element of the bias frame. The bias is important because it displaces the zero-point of the pixel-value scale away from the zero signal of the CCD. Because of this offset, pixel values in the image are not proportional to the signal. When the bias is removed, the zero points of the CCD output and pixel-value scales coincide, and pixel values are proportional to the signal from the CCD.

## Chapter 6: Image Calibration



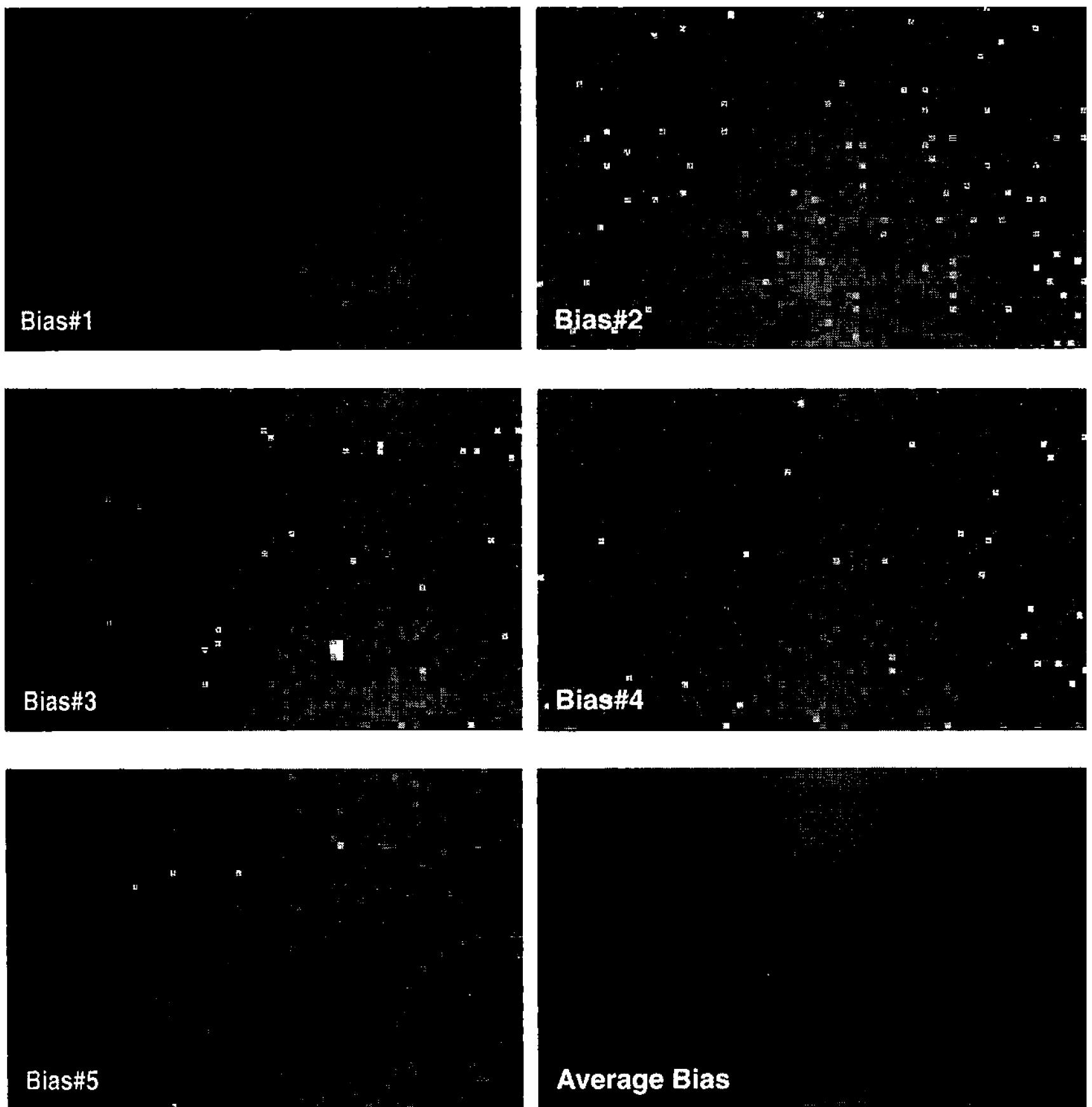
**Figure 6.5** This is the average of the same 64 bias frames shown in Figure 6.4, shown using the same black and white display values. Note that the overwhelming majority of pixels have a value of 100 (between 99.50 and 100.49 ADU), showing that for this CCD camera, a single bias value is justified.

The amplitudes of the fixed-pattern bias, interference, and readout noise are usually quite small, sometimes less than one ADU in a master bias frame. When this is the case, it is best to treat the bias frame as if it had only one value,  $B_{t,j}$ , obtained by taking the average value of the bias frame.

For example, if your CCD camera has a measured readout noise of 20 electrons r.m.s., and the conversion factor is 30 electrons per ADU, averaging bias frames will produce a master bias frame with most pixels having the average bias value, plus a scattering of pixels with values one ADU lower or higher than the rest. The low and high pixels are probably no different from the average pixels; they are simply statistical accidents. In this case, it is best to use a single bias value.

### 6.2.1.2 Bias with Drift-Subtraction

Some CCD cameras, the Cookbook camera among them, feature a *drift subtract mode* designed to eliminate changes in the bias level caused by temperature-sensitive components in the camera. When this mode is active, the acquisition software continues to clock the CCD after the image storage area has been emptied, and several hundred readings of  $B_{x,y}$  are averaged to obtain the system bias. This value is subtracted from every pixel in the image, and a fixed bias (usually 100) is added to prevent negative values from appearing in the image data. This technique



**Figure 6.6** These enlarged sections of five bias frames were taken a few seconds apart, yet no two are the same. When many bias frames are averaged—the case for the frame in the lower right—the root-mean-square noise decreases with the square root of the number of bias frames.

removes the effects of a drifting analog bias value, and locks the value of  $\langle \text{BIAS} \rangle_{x,y}$  at the fixed bias value. If you use drift subtract mode, use it for *all* of your images and for *all* of your calibration frames.

- **Tip:** *AIP4Win allows you to use a single bias value in the advanced calibration protocol. The Cookbook camera and several commercial CCD models make drift subtraction an option in their image acquisition software.*

### 6.2.1.3 When to Make a Master Bias Frame

To calibrate a scientific-grade CCD camera with a 16-bit analog-to-digital converter, it is desirable to create a master bias frame by calculating the average of many bias frames (or, alternatively, by finding the median value of many bias frames). Assuming normal statistics, the readout noise and the unpatterned bias

## Chapter 6: Image Calibration

decrease with the square root of the number of frames averaged.

Consider a high-quality CCD with measured readout noise of five electrons r.m.s. and a conversion factor of one electron per ADU. The pixels in a single bias frame display a random variation of five ADUs about some central value. Averaging 50 bias frames will reduce statistical uncertainty in the bias to less than one ADU, thereby improving the quality of the calibration. In so sensitive a camera, a high-quality master bias frame will help define low-level bias patterns that may exist.

There are two basic methods of combining multiple bias frames: averaging and taking the median. If you operate your CCD in an electrically noisy environment, individual bias frames sometimes show large noise spikes. Averaging bias frames makes these abnormal events part of the master bias frame, whereas determining the median of the bias frames excludes the abnormal values, but does not reduce random noise as effectively. The bottom line is: use the median if you operate in an electrically noisy environment; use the average if you do not.

- **Tip:** *To take bias frames, your CCD camera should be fully cooled and operating normally. Cap the telescope or close the camera's shutter. Set the integration time to the minimum allowed by the operating software, and then make at least as many bias frames as you make dark frames.*

### 6.2.2 Dark Frames

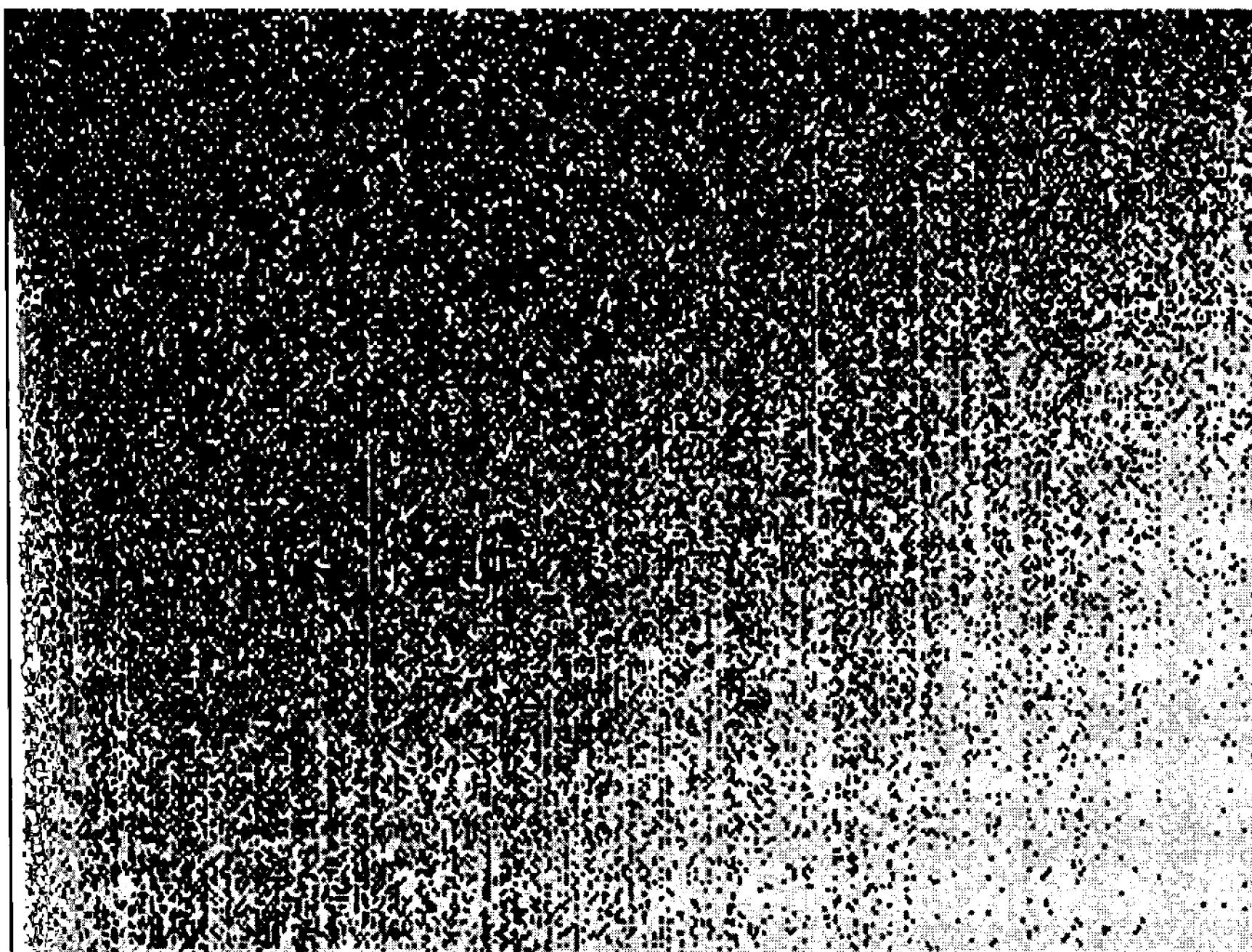
The dark frame captures a sample of the dark current to be used in peeling away the dark-current layer from an image. During the integration time of a dark frame, no light is allowed to strike the CCD. Depending on the calibration protocol, the integration time is chosen to be equal to or greater than the integration time for the images to be calibrated. For amateur CCD cameras, taking good dark frames is the single most important step in the calibration process.

The dark frame contains the thermal electrons that accumulate during integration, of course, but it also contains thermal noise, a random variation in the number of thermal electrons that accumulate, *plus* all of the elements that make up a bias frame:

$$\langle \text{DARK} \rangle_{x,y} = \langle \text{BIAS} \rangle_{x,y} + \frac{1}{g} \{ t d_{x,y} + \sigma_{\text{TE}} \}. \quad (\text{Equ. 6.8})$$

Recall that the factor  $1/g$  converts the number of electrons to ADUs. The accumulation of thermal electrons is  $t d_{x,y}$ , the product of the dark current and the integration time. Thermal noise,  $\sigma_{\text{TE}}$ , is the random variation in the number of thermal electrons. It obeys a simple law that governs many random processes involving unlikely events over long intervals: the standard deviation in the number of thermal electrons is the square root of the number of electrons:

$$\sigma_{\text{TE}} = \sqrt{t d_{x,y}}. \quad (\text{Equ. 6.9})$$



**Figure 6.7** Above is an average of 16 dark frames with a 60-second integration time, displayed with black = 100 and white = 102. At this stretch, the dark frame clearly shows the underlying fixed-pattern bias. Pixels with high dark current, including a few scattered pixel values as high as 627, display as white.

Other phenomena, such as the number of raindrops that fall on a patch of ground in a given interval, or the number of clicks from a Geiger counter in a given time, obey the same statistical law.

Consider an example: if the average rate at which thermal electrons accumulate is 100 electrons per second, then in one average second, 100 electrons will accumulate. However, in any particular 1-second interval, the number of electrons will be  $100 \pm 10$  electrons. The “plus or minus” symbol means that 68% of the time, some number between 90 and 110 electrons will accumulate. This statistical property has important consequences for CCD imaging: it means that a dark frame is merely a *sample* of the dark current, not a precise measure.

### 6.2.2.1 “Image-Times-Five” Rule for Dark Frames

The purpose of taking dark frames is to determine the dark current accurately, so that you can peel away another layer of the “onion.” To accomplish this, it is necessary to acquire a large sample of thermal electrons. To see why this is so, look at the numbers: for 100 electrons, the uncertainty is 10%. Take a bigger sample of 1,000 electrons and the uncertainty is  $1000 \pm 31.6$ , or 3%. Take an even bigger sample of 10,000 and the uncertainty drops to 1%. The goal in calibration is to create a dark frame that is sufficiently accurate that subtracting it from a raw image will not significantly increase the noise in the calibrated image.

## Chapter 6: Image Calibration

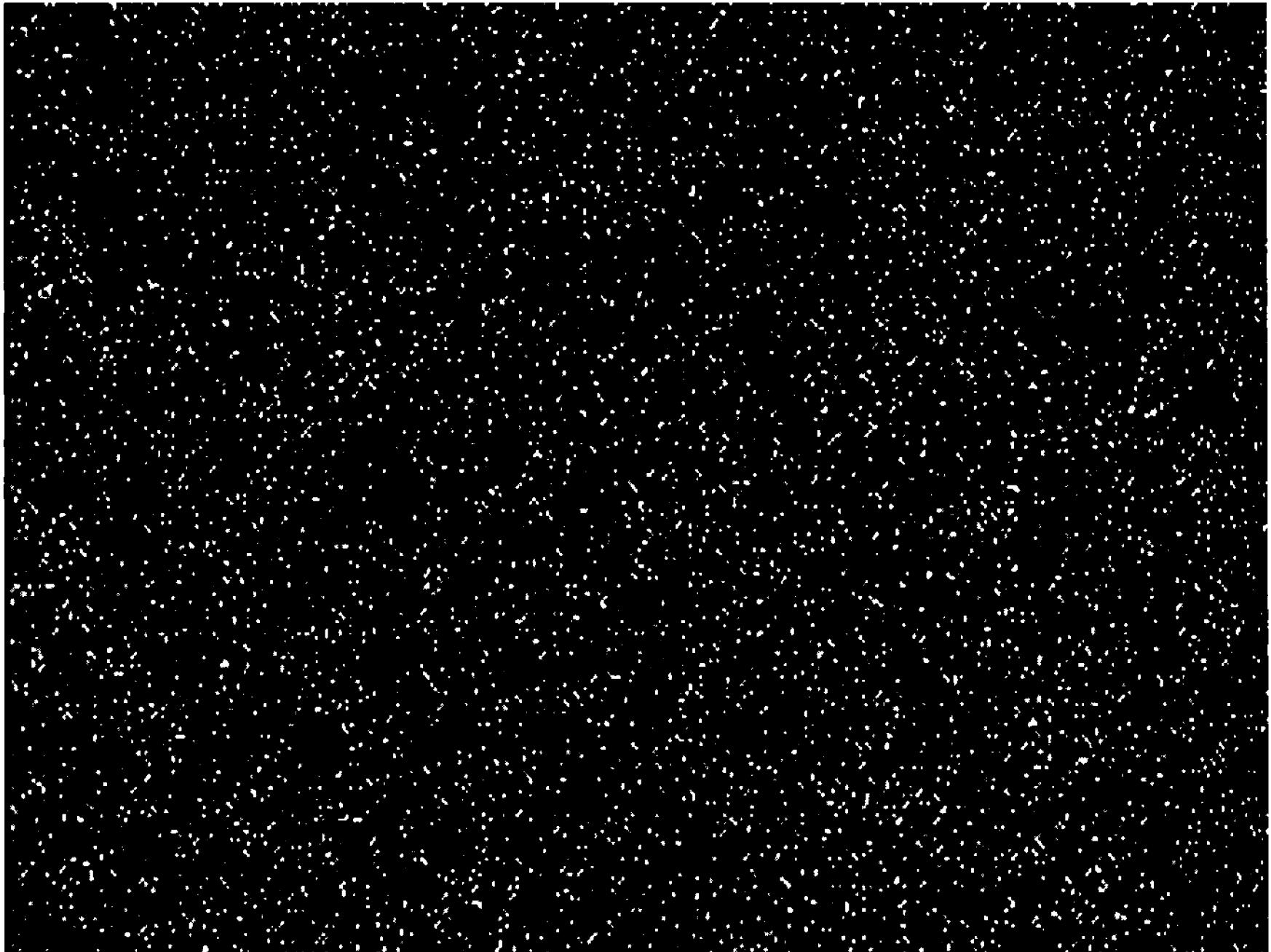


Figure 6.8 Above you see the same dark frame shown in Figure 6.7, but displayed with black = 100 and white = 120. The dark frame is the sum of the bias plus the dark current, but the principal effect of the bias is to add an average of 100 ADUs to the dark current that accumulates during the 60-second integration.

This goal is met when the total integration time of the dark frames (whether the information is collected in a long integration or is the average of many short ones) is *at least* five times longer than the image integrations. This number is somewhat arbitrarily chosen because it reduces the noise contribution caused by dark subtraction to an addition of 10%, which is generally acceptable.

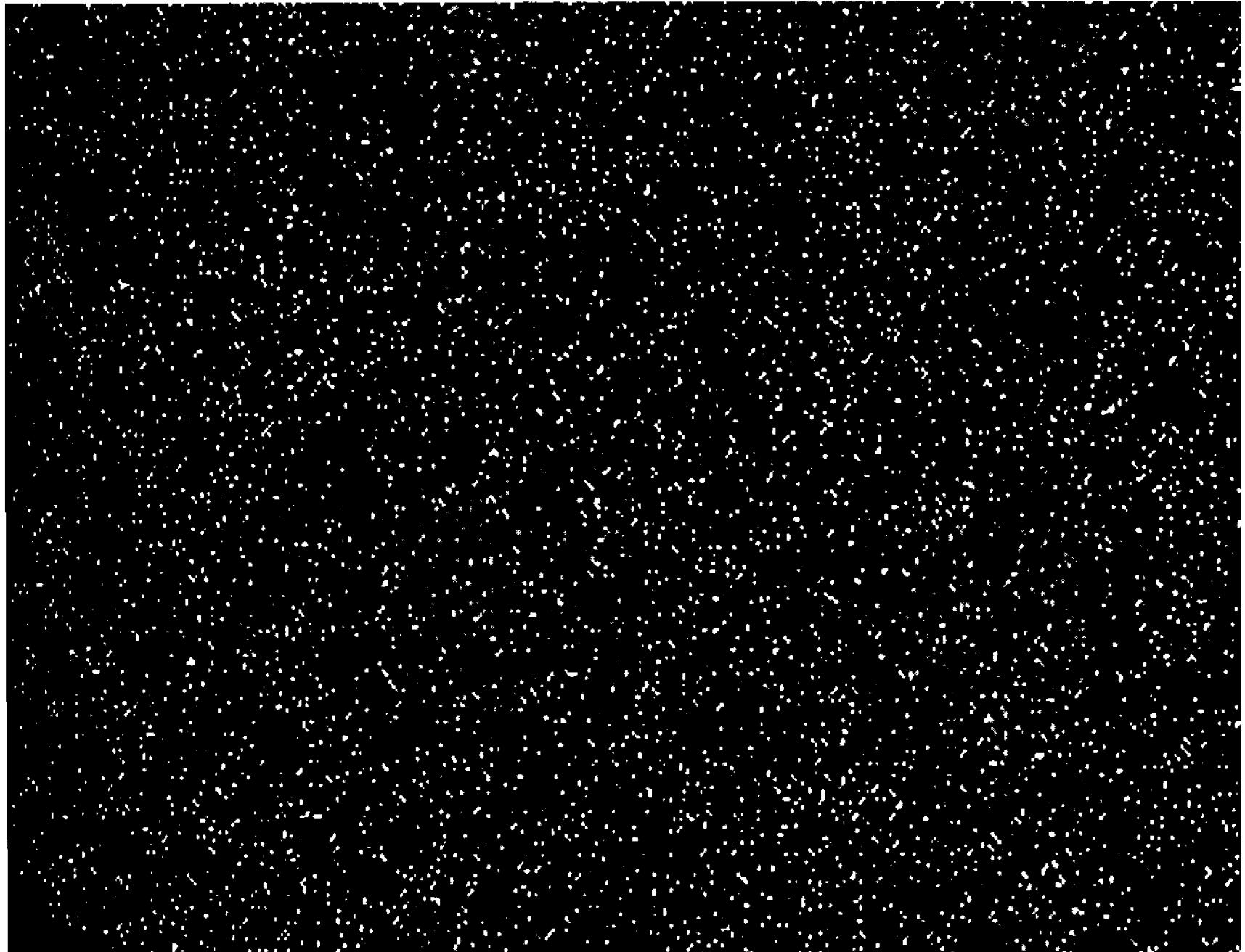
Here is the detailed thinking behind the “image-times-five” rule: suppose that a raw image accumulates some number of thermal electrons,  $td_{x,y}$ . The thermal noise is

$$\sqrt{td_{x,y}},$$

or  $\sigma_{TE}$ . Single dark frames with  $T$  thermal electrons will also have noise  $\sigma_{TE}$ . When the images are subtracted, the total number of thermal electrons will be zero, but because noise adds in quadrature, the noise will be

$$\sqrt{\sigma_{TE}^2 + \sigma_{TE}^2} = 1.414\sigma_{TE},$$

a 41% increase in thermal noise in the dark-subtracted image over that in a raw image. However, if the dark-frame integration is increased from  $t$  to  $5t$ , so that  $5td_{x,y}$  thermal electrons accumulate, the thermal noise increases to



**Figure 6.9** This is a scalable dark frame (thermal frame) displayed with black = 0 and white = 100. It is the average of ten dark frames of 300 seconds' integration each *minus* the average of 64 bias frames. Subtracting bias from a dark frame creates an image with pixel values proportional to the dark current.

$$\sqrt{5}\sigma_{\text{TE}}.$$

If we now divide the pixel values in the dark frame by 5, the amplitude of the thermal noise becomes

$$\frac{1}{5}\sqrt{5}\sigma_{\text{TE}} = 0.447\sigma_{\text{TE}}.$$

When the thermal electrons in the dark frame are subtracted from those in the raw image, the total noise becomes

$$\sqrt{\sigma_{\text{TE}}^2 + (0.447\sigma_{\text{TE}})^2} = 1.095\sigma_{\text{TE}},$$

an increase in thermal noise just under 10%, which is usually acceptable. Obviously, the longer the dark integration time, the less the noise added during calibration.

### 6.2.2.2 Thermal Frames

Dark frames cannot be scaled because they contain a bias offset in addition to the accumulation of thermal electrons. Imagine how nice it would be if you had a “thermal frame” consisting only of thermal electrons:

## Chapter 6: Image Calibration

$$\langle \text{IDEAL-THERMAL} \rangle_{x,y} = \frac{1}{g} \{ t d_{x,y} + \sigma_{\text{TE}} \}. \quad (\text{Equ. 6.10})$$

In reality, dark frames contain the unwanted baggage of a bias frame. To get at the thermal electrons, you must make a bias frame and subtract it from the dark frame:

$$\langle \text{THERMAL} \rangle_{x,y} = \langle \text{DARK} \rangle_{x,y} - \langle \text{BIAS} \rangle_{x,y}. \quad (\text{Equ. 6.11})$$

When you perform the subtraction, the bias level,  $B_{x,y}$ , the fixed-pattern bias,  $(B_{\text{FP}})_{x,y}$ , and the thermal electrons that accumulate during readout,  $t_{\text{RO}} d_{x,y}$ , are indeed removed. However, a real-world thermal frame contains not only thermal electrons and thermal noise but also readout noise from two readouts *plus* noise from two doses of unpatterned interference. The noise adds in quadrature; so, for example, two readouts produce  $\sqrt{2}$  times the noise of a single readout:

$$\langle \text{THERMAL} \rangle_{x,y} = \frac{1}{g} \{ t d_{x,y} + \sigma_{\text{TE}} + \sqrt{2} \sigma_{\text{RO}} + \sqrt{2} (\sigma_{\text{UP}})_{x,y} \}. \quad (\text{Equ. 6.12})$$

When the integration time is long enough for several thousand thermal electrons to accumulate,  $\sigma_{\text{TE}}$  dominates the other noise sources, and it is possible to obtain high-quality measurements of the dark current.

However, a thermal frame is scalable, meaning that you can multiply its pixel values to recreate a dark frame with an arbitrary integration time. A normal dark frame is *not* scalable because of the bias value that it contains. If you can create a scalable dark frame, you can match its dark current to the dark current in image frames taken with different integration times.

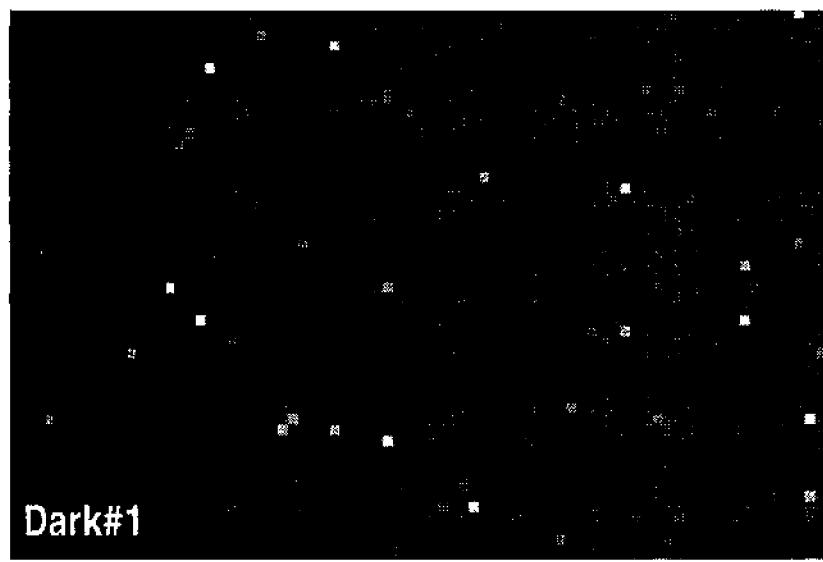
### 6.2.2.3 Standard and Scalable Dark Frames

There exist two widely employed strategies for creating master dark frames. The first strategy is used to create master dark frames for the standard calibration protocol, and the second is used to create scalable master dark frames for the advanced calibration protocol.

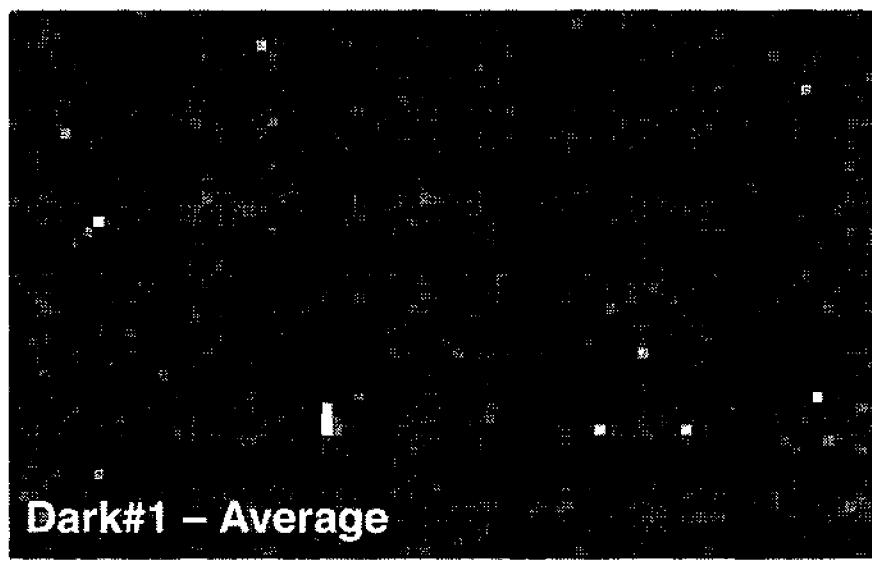
- The standard protocol requires dark frames that are made from multiple dark frames having the *same integration time* as the raw images taken by the observer; because in the standard calibration protocol, the dark frame is subtracted directly from the raw images, without the use of a bias frame.
- The advanced protocol can use dark frames with any integration time—that is, scalable dark frames. Ideally, these are com-

Figure 6.10 Opposite: “If you’ve seen one dark frame, you’ve seen them all.” Not true! Even though dark frames are similar, they are by no means identical. Every dark frame is unique. The left column contains greatly enlarged sections of five 60-second dark frames; if you inspect them closely, you will see the differences. At right, the differences are highlighted: from each dark frame, the average of all of the dark frames has been subtracted, leaving the residual dark noise.

## Section 6.2: Calibration Frames



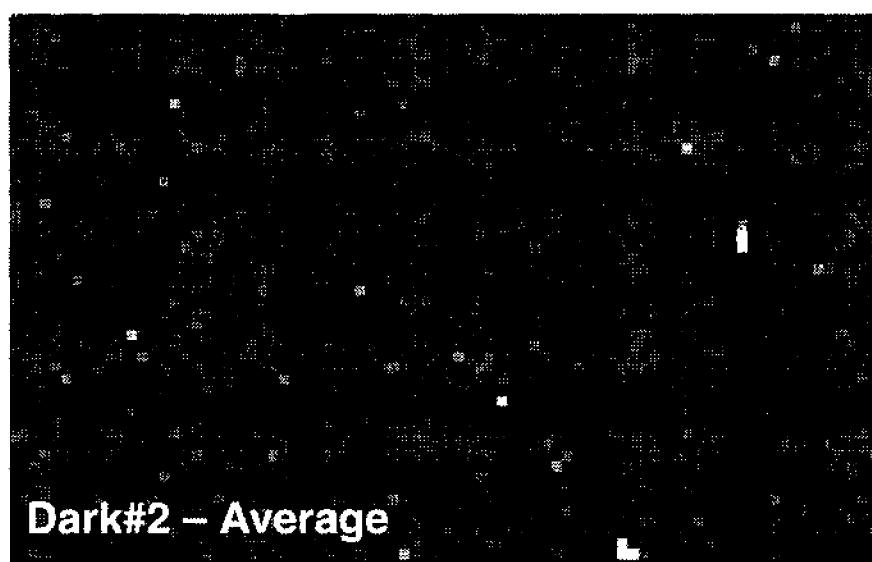
Dark#1



Dark#1 – Average



Dark#2



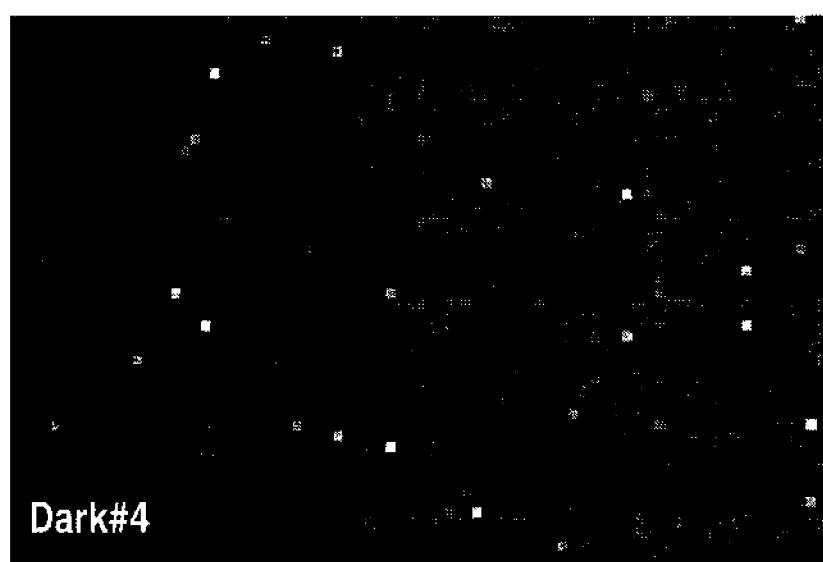
Dark#2 – Average



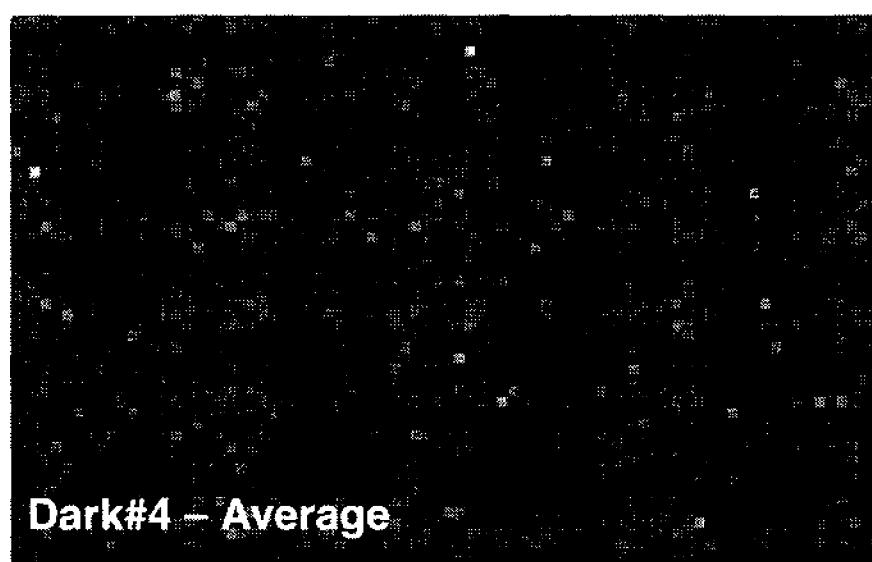
Dark#3



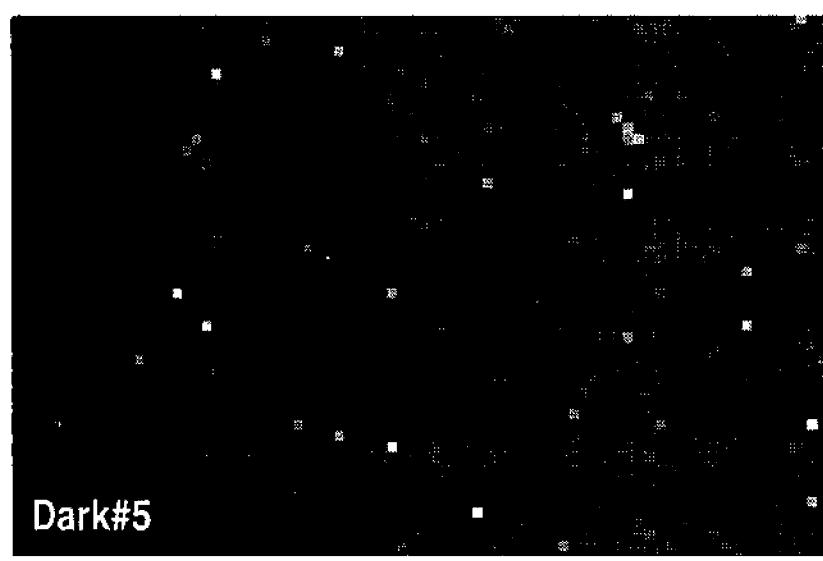
Dark#3 – Average



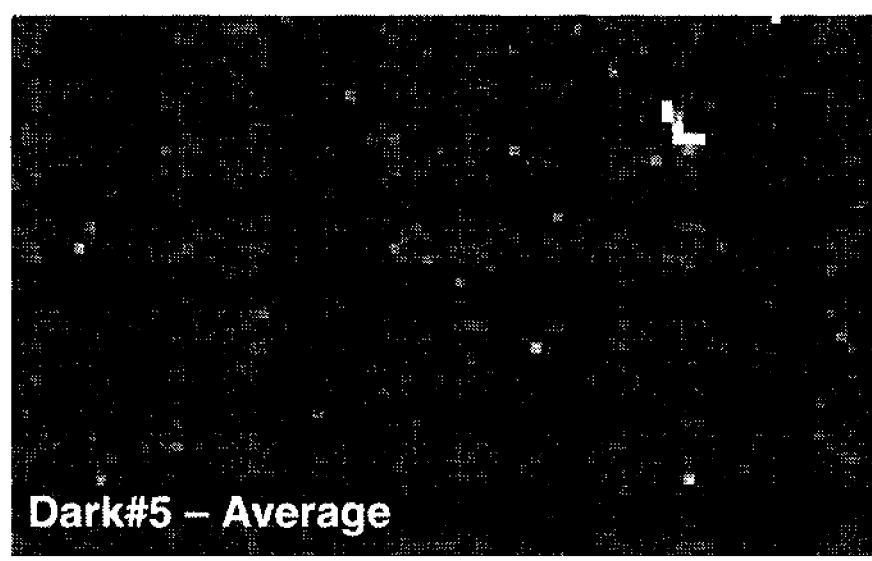
Dark#4



Dark#4 – Average



Dark#5



## Chapter 6: Image Calibration

binations of dark frames having integration times that are longer than the longest integration time used for any raw image. In calibration, a master bias frame is subtracted from the master dark, and the resulting thermal frame is scaled.

Many observers prefer the standard protocol because they do not wish to expend time and effort taking bias frames, and the standard calibration procedure is simpler. On the other hand, it forces the observer to use the same integration time for all images, or to shoot different sets of dark frames having different integration times.

However, the time required to make, for example, 16 bias frames and 10 dark frames with 300-second integration times for the advanced calibration protocol is repaid by precise subtraction of dark current with somewhat lower thermal noise. Moreover, for CCD cameras that produce electronically “clean” images, the master bias frame can be replaced with a single bias value, making the advanced calibration protocol just as easy to use as the standard procedure, but having the advantage of greater flexibility in choosing image integration times.

### 6.2.2.4 Dark Frame Matching

Because the accumulation of thermal electrons during integration is proportional to integration time, the number of thermal electrons in a thermal frame, i.e., a bias-subtracted dark frame, can be matched to an image with a different integration time simply by multiplying by the ratio of exposure times:

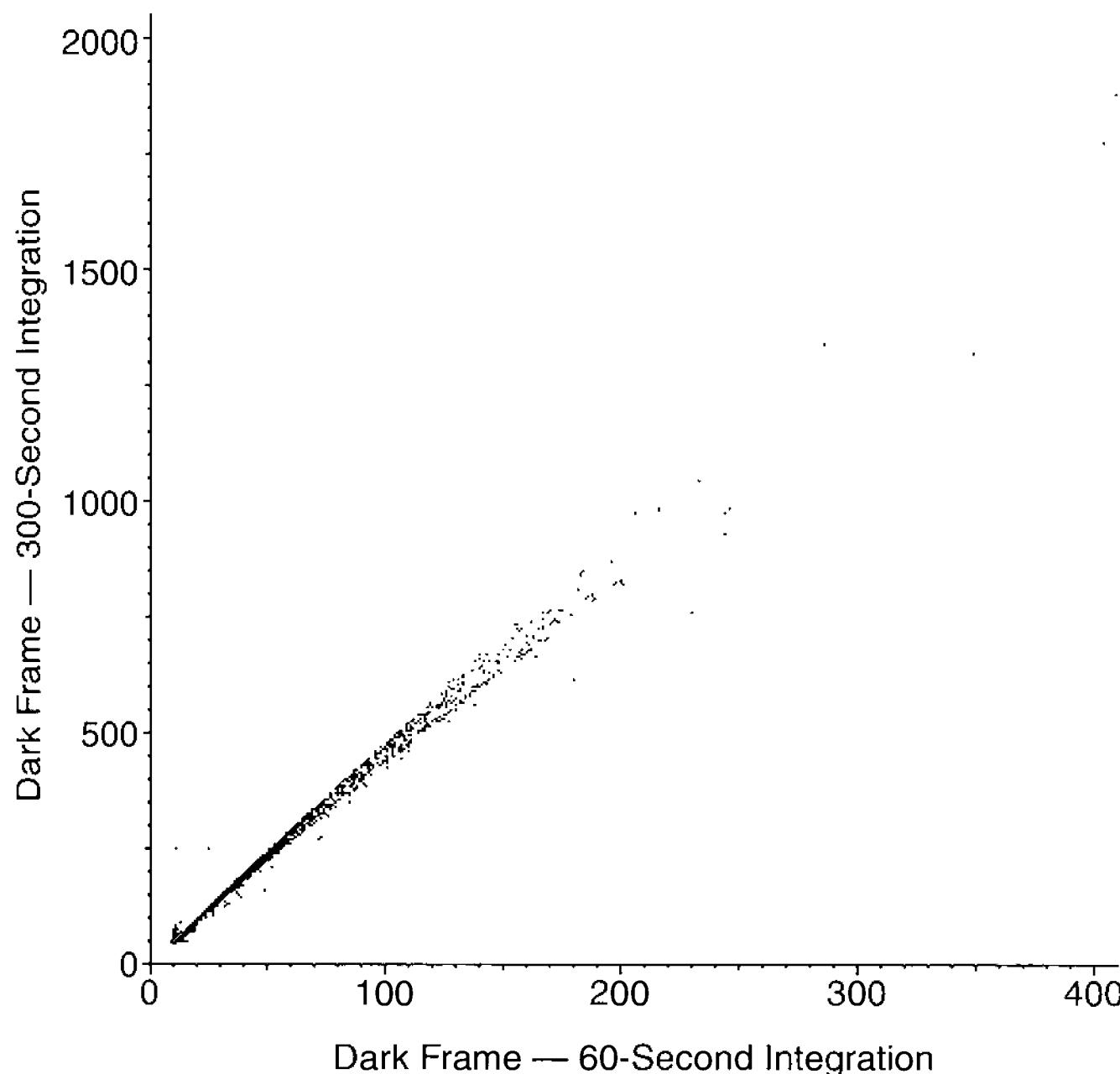
$$t_1 d_{x,y} = \frac{t_1}{t_2} (t_2 d_{x,y}). \quad (\text{Equ. 6.13})$$

Knowing the integration times, a master dark frame made with 300-second integrations can be scaled to match to images with 60-second integrations. This is done by subtracting the bias value, multiplying the pixel values in the master dark by a factor of

$$\frac{60}{300} = 0.2,$$

and then subtracting the scaled master dark frame.

Because of a small change in the temperature of the CCD, you may not know how many thermal electrons have accumulated, but it is possible to determine the relationship using the small population of pixels with higher dark current than most photosites—the so-called “hot” pixels. The dark-matching algorithm searches the dark frame for a representative sample of “hot” pixels and measures their excess compared to adjacent photosites, and then determines the excess for corresponding pixels in a raw image. For each hot pixel, the algorithm determines the ratio of the excesses, discards spurious values, and averages the representative ratios to obtain a scaling factor. The dark frame is then bias subtracted, scaled using that ratio, and subtracted from the image.



**Figure 6.11** This cross-histogram plots pixel values in a 60-second dark frame against the pixel value for the same pixel in a 300-second integration. Although not made at the same temperature, the points lie close to a straight line; therefore scaling even compensates for small changes in CCD temperature.

- **Tip:** *AIP4Win features automatic dark-frame matching. Dark-frame matching enables observers to make one set of dark frames with long integration times and then use a master dark frame made from them to dark-subtract images with differing integration times made during the night.*

### 6.2.2.5 Changing CCD Temperature

Dark current depends quite sensitively on the temperature of the silicon substrate of the CCD. If its temperature varies by a fraction of a degree Celsius during the interval between taking an image and taking a dark frame, the number of thermal electrons in the dark current will not match the number in the image; and you will see “hot” pixels or “dark” pixels in the calibrated image.

If you observe with a CCD camera that does not actively control the CCD temperature, use the advanced calibration protocol with automatic dark-frame-matching. Even if the CCD temperature changes by a few degrees, a dark-frame-matching algorithm such as that described above can find the best fit between the master dark frame and the dark current generated during the image integration.

- **Tip:** *In its Advanced Calibration tool, AIP4Win allows you to automatically scale (or “match”) a dark frame to the image. In addition to allowing you to use dark frames with integration times that*

## Chapter 6: Image Calibration

*differ from those used for images, automatic dark-frame matching allows you to calibrate images using dark frames taken at somewhat different temperatures.*

### 6.2.2.6 When to Use a Single Dark Value

Throughout the preceding discussion, we have assumed there will be a significant accumulation of thermal electrons during image integrations, and also that different photosites have significantly differing dark current. However, if you observe with a CCD that has extremely low, uniform levels of dark current, you can subtract the average value of the dark frame from all pixels in the image. When thinned and back-illuminated scientific-grade CCDs are cooled below  $-80^{\circ}$  Celsius, the dark current becomes vanishingly small (i.e., in the range of 1 to 3 electrons per photosite per hour). If the variation in the accumulated thermal electrons from one photosite to the next is random, and if the accumulation of thermal electrons drops below about half the readout noise of the CCD, then it is best to determine the average value of dark current for the whole CCD and subtract this single dark value from the entire raw image.

### 6.2.2.7 Cosmic Ray Events

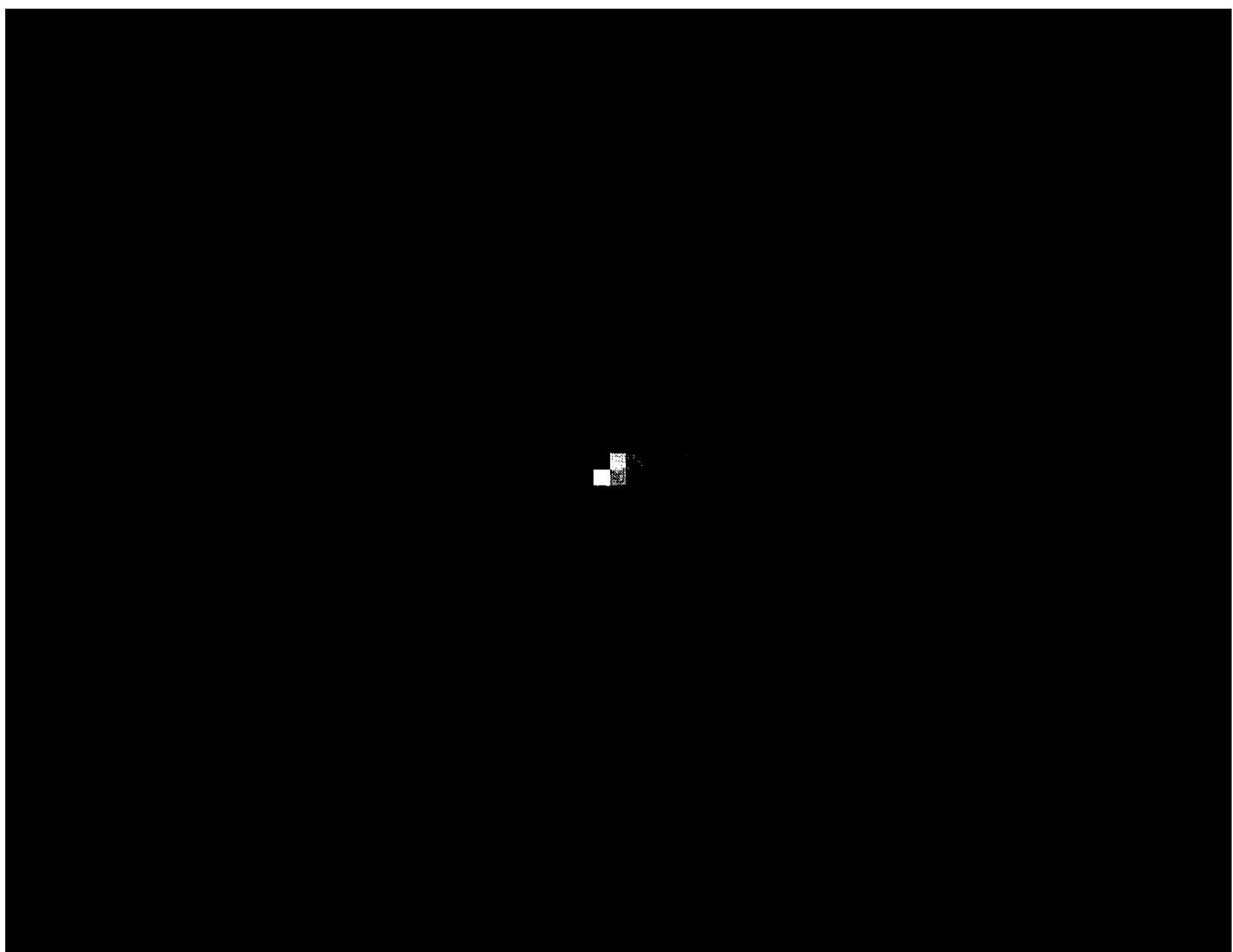
When cosmic rays strike the Earth's upper atmosphere, they trigger a shower of high-energy particles, some of which reach the ground. When a high-energy subatomic particle crashes through the silicon matrix of your CCD, it leaves a track of several thousand free electrons. Cosmic ray events may appear as an intensely bright point of light or as a track several pixels in length. When a dark frame with a cosmic ray is used for calibration, the resulting images have a dark spot where the cosmic ray signal was subtracted.

Even when you average multiple dark frames to create a master dark frame, a single cosmic ray hit can mar all images calibrated with it. Instead of averaging, it is often good practice to take the median of multiple dark frames. Although the median combination does not reduce the statistical variation in the number of thermal electrons as effectively as averaging, it rejects extreme pixel values such as those arising from a cosmic ray hit.

- **Tip:** *AIP4Win supports both dark-frame averaging and determining the dark-frame median. If you average dark frames, it is a good idea to inspect the individual frames for cosmic ray hits before averaging them. Do not include frames with obvious cosmic ray hits in an averaged master dark frame.*

### 6.2.2.8 Electroluminescence

Electronic circuit elements on the CCD may act as light-emitting diodes, giving off light whenever voltage is applied to them. This effect is called electroluminescence. In the earliest amateur CCD cameras, based on the Texas Instruments TC211 chip, the bright spot in the corner was the most prominent feature in the



**Figure 6.12** Here is a typical cosmic ray hit. Whenever you shoot images or dark frames, your images accumulate cosmic rays. Although the rate of accumulation depends on your altitude and the size of the CCD, you should expect around ten cosmic rays to hit each square centimeter of your CCD each hour.

dark frame as seen in Figure 6.13. Fortunately, electroluminescence acts just like dark current and can be subtracted exactly as if it were.

### 6.2.2.9 How to Make Master Dark Frames

How you take the dark frames needed to create a master dark frame depends on how you plan to calibrate your images. If you plan to use the standard calibration protocol, the integration time of individual dark frames should be the same as that used in your images. For the advanced calibration protocol, ideally you should use integration times that are at least five times longer than the integrations used for your images.

To collect data for making a master dark frame, close any shutter in the CCD camera and cap the telescope. It is essential that *no light* reach the CCD when you are making dark frames. The camera should be cooled to normal operating temperature. Often the best strategy is to make dark frames midway through an imaging session, but that depends on your observing program. If you plan to use the advanced calibration protocol, shoot your bias frames at the same time you shoot your dark frames. This insures that you will have scalable thermal frames.

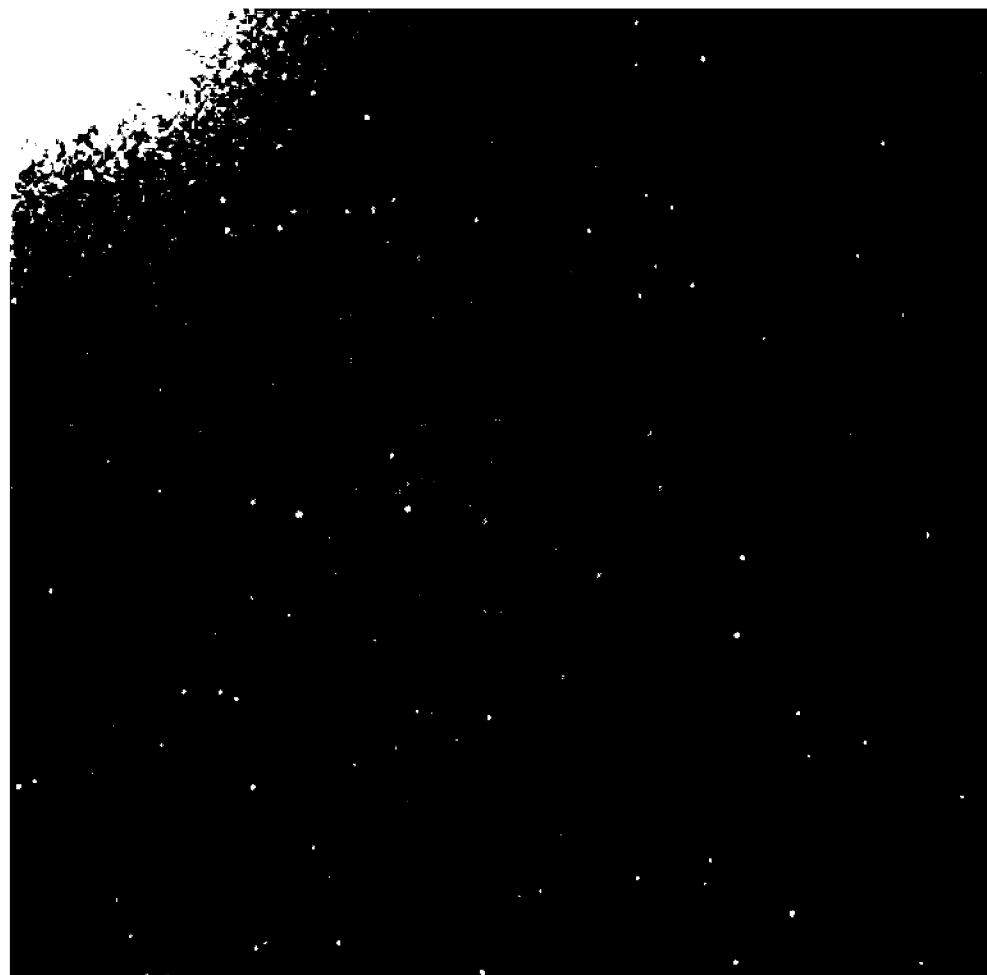


Figure 6.13 Electroluminescence brightens the upper left corner of this dark frame, a 60-second integration with a TC211-based camera. The glow occurs because the on-chip amplifier remained energized during integration, and glowed like an LED. Turning off the amplifier during integration eliminates the glow.

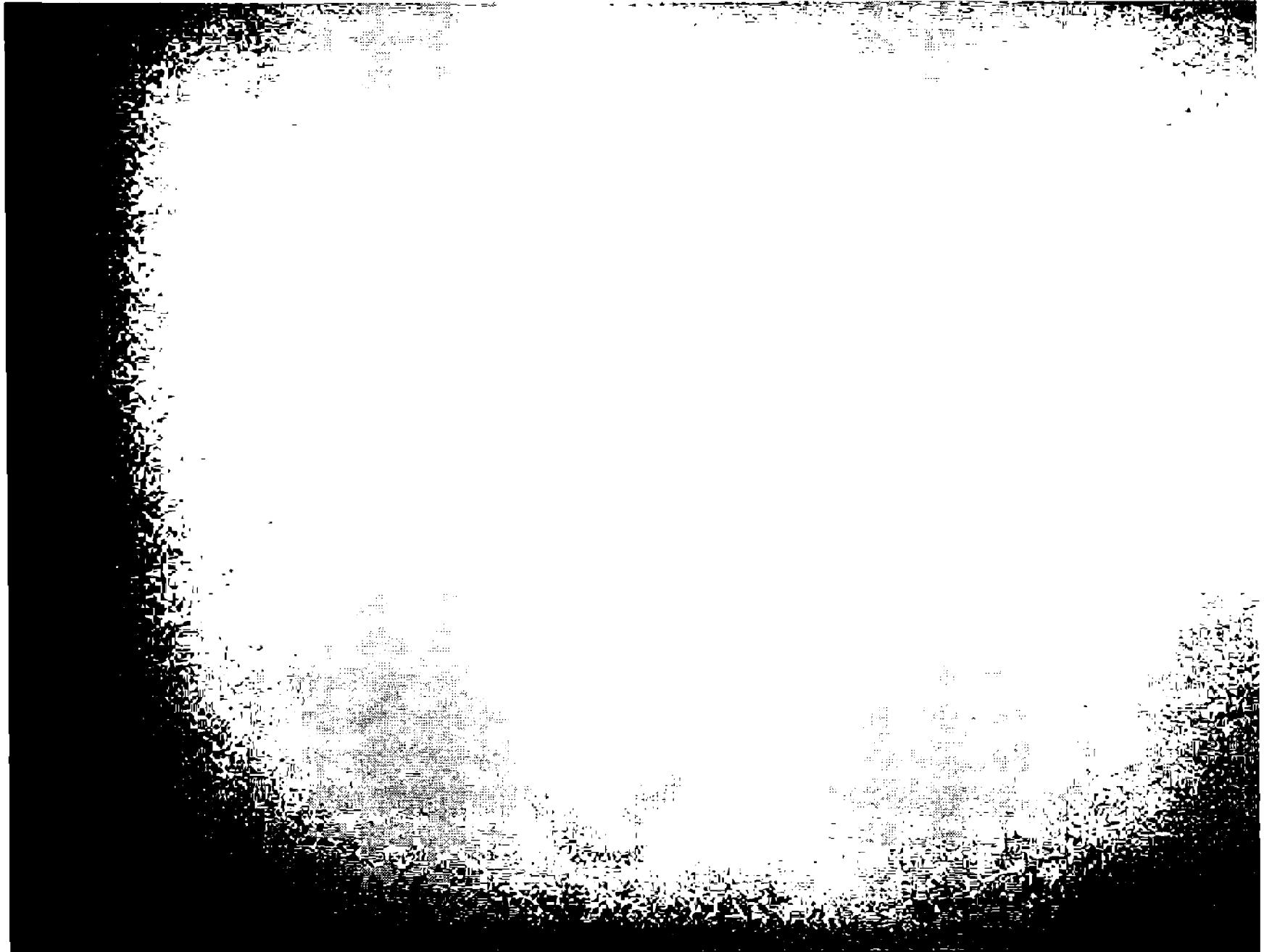
- **Tip:** *For the basic and standard calibration protocols, your CCD camera must be fully cooled before you make dark frames. For best results, make dark frames well into an observing session. Cap the telescope and close the camera's shutter. Use the same integration time that you use for images. Make at least 10 dark frames.*
- **Tip:** *For the advanced calibration protocol, make at least 10 dark frame integrations using an integration time five times the normal integration time that you use for images. For the best results, make bias frames at the same time you make dark frames.*

### 6.2.3 Flat Frames

CCD sensitivity variations and vignetting are deeply buried layers of the raw-image onion, so deeply buried, in fact, that they cannot be removed until the overlying dark current and bias layers have been peeled away. Although CCD sensitivity variations and vignetting can be corrected in two separate steps, they are almost always treated as one “layer” and removed by dividing the image by a master flat-field frame.

The flat-field frame records the response of the entire optical system—the telescope, filters, window, cover glass, and CCD itself—to a uniform, or “flat,” field of light. The resulting flat-field images cannot distinguish whether optical vignetting, quantum efficiency, or some combination of the two produced a particular pixel value in the frame, but it does not matter. So long as the optical system and CCD do not change, a good master flat-field allows the observer to correct both effects as if they were one.

## Section 6.2: Calibration Frames



**Figure 6.14** The image above is a master flat-field frame made by taking the average of 16 flat-field frames and subtracting the average of 16 flat-dark frames. In the version above, 1900 displays as black and 2300 as white; so the rather bland-looking flat is actually seen at five times normal contrast.

Recall from Equ. 6.4 that the number of electrons generated at photosite  $(x, y)$  is:

$$E_{x,y} = tV_{x,y}Q_{x,y}I_{x,y}. \quad (\text{Equ. 6.14})$$

When we illuminate the CCD with a “flat” (uniform) field of light, the resulting data can serve as a map of the CCD’s efficiency in converting photons into electrons. Since both  $t$  and  $I_{x,y}$  are constant, the pixel values in the image are proportional to  $V_{x,y}Q_{x,y}$ . Given this information, we can allow for and correct both effects in raw images taken with the CCD camera on this particular telescope. This is the role of the master flat-field frame.

Like all images, a flat-field frame contains not only the signal that we want, but also bias and thermal electrons:

$$\langle \text{FLATRAW} \rangle_{x,y} = \frac{1}{g}(tV_{x,y}Q_{x,y}I_{x,y}) + \langle \text{BIAS} \rangle_{x,y} + \langle \text{DARK} \rangle_{x,y}. \quad (\text{Equ. 6.15})$$

To remove the bias and dark contributions from the raw flats, we must therefore shoot flat-field dark frames:

$$\langle \text{FLATDARK} \rangle_{x,y} = \langle \text{BIAS} \rangle_{x,y} + \langle \text{DARK} \rangle_{x,y}. \quad (\text{Equ. 6.16})$$

## Chapter 6: Image Calibration

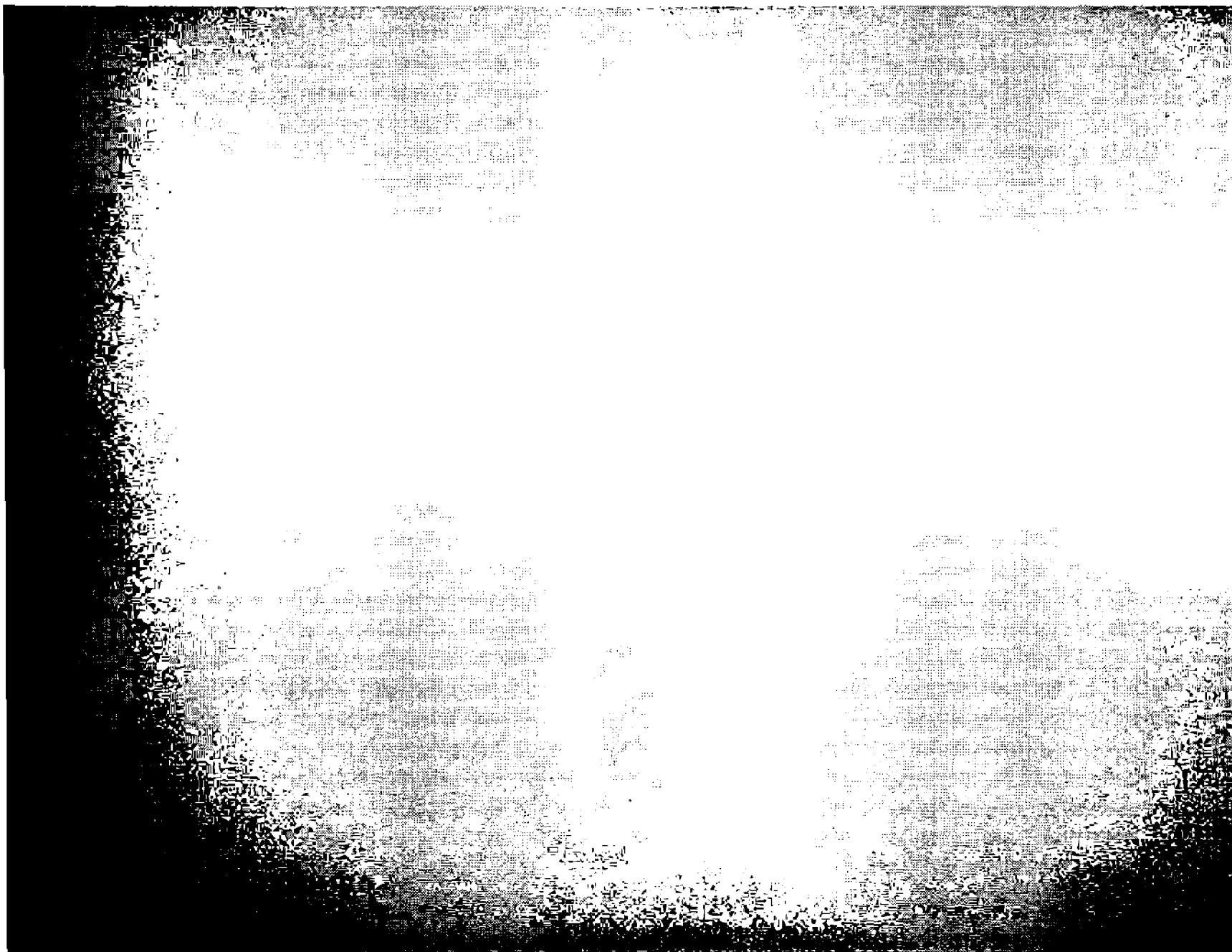
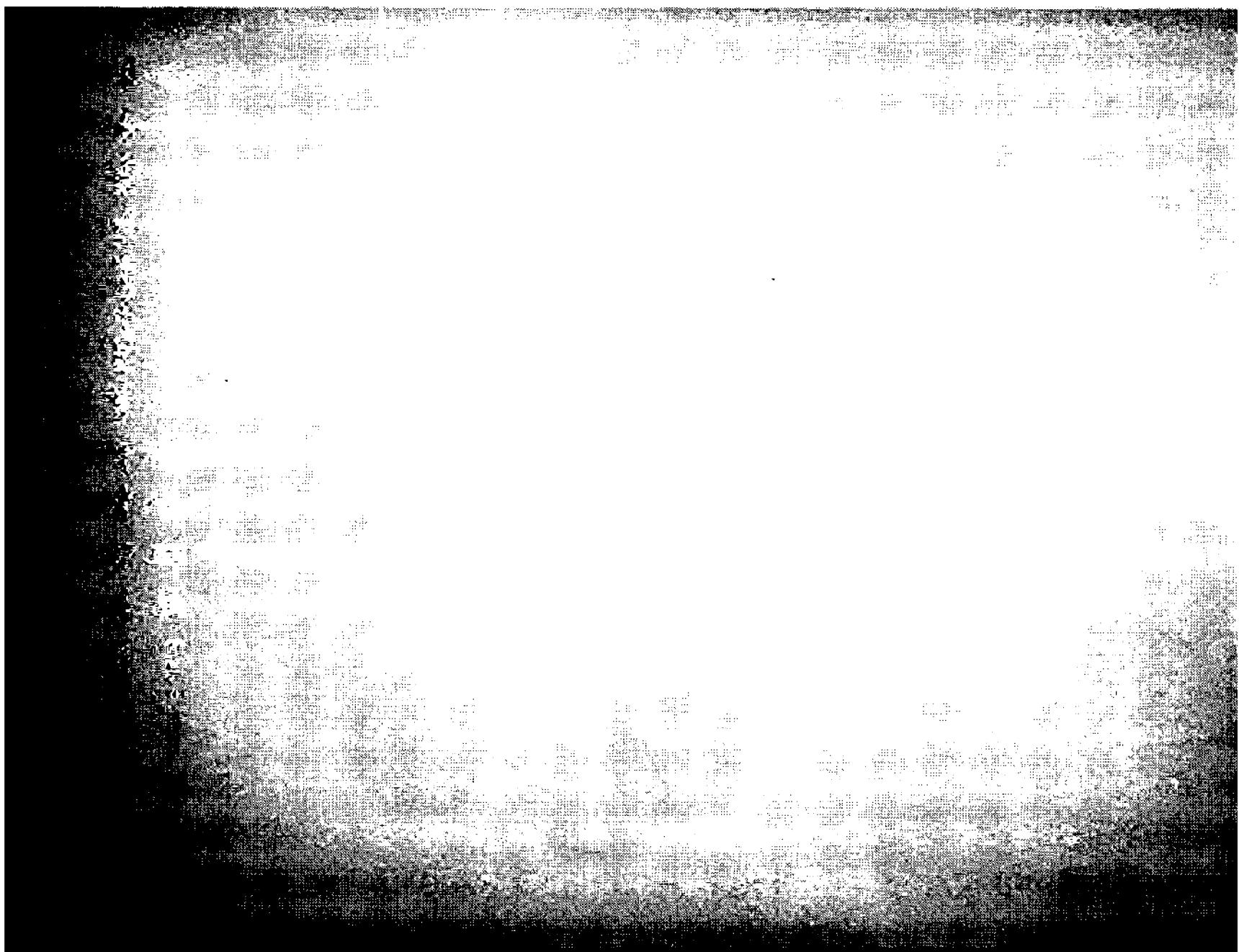


Figure 6.15 This is just one of the images used to make the flat shown in Figure 6.16, shown with a 10x stretch in contrast (black = 2000, white = 2200). Even though the image has been optimally exposed to half the full-well capacity of the CCD, it nonetheless appears slightly noisy.

Although it does not appear explicitly, it is important to remember that the integration time for a flat-field dark frame (a.k.a. “flat dark”) must be the same as the integration time for the raw flats. It is also desirable to shoot a special set of flat darks to insure that the flat dark is an independent sample of the dark current.

Ideally, the raw flat-field exposure should fill the charge wells in the CCD photosites to roughly one-half their full depth, enough to produce a strong signal but not so close to saturation that the CCD’s response to light becomes nonlinear. If you are using an artificial light source to make flat-field frames, set its brightness so the integration times are between 2 and 10 seconds. In the CCDs used by amateur astronomers, the signal will be at least 40,000 photoelectrons, which implies a random variation of roughly 200 electrons. The flat-field frame is, therefore, photon-noise limited rather than limited by any of the internal noise sources of the CCD.

To make a master flat-field frame of extremely high quality, shoot at least 16 raw flat-field frames and 16 flat darks. To make the master, average the flats and average the flat darks, then subtract the averaged flat dark from the average flat field. With a signal of 640,000 total electrons, even allowing for the readout noise from 32 image readouts, the signal-to-noise ratio of the master flat-field frame should be around 600, sufficiently good that no loss of image quality will result



**Figure 6.16** By averaging flat-fields, you can beat down random noise. Compare this flat, made by averaging 16 raw flats, with Figure 6.15, made from just one frame. Both are shown with black = 2000 and white = 2200. The averaged image has a signal-to-noise ratio of ~400:1, giving excellent flat-field correction.

from flat-fielding during calibration.

Several factors bedevil the making of flat-field frames. First is that the optical configuration must be the same as that used for images. Removing or rotating the CCD camera, or even changing the focus, can invalidate a carefully prepared master flat-field frame. In some telescopes, movement of optical components in their cells when the telescope pointing changes can alter the optical configuration. Dust can fall on windows and filters, filter slides may position the filter differently with every insertion, and internal reflections can mimic the appearance—but not the behavior—of vignetting. Scattered light is another factor that can lead to poor flat-fielding. If flat-fielding light reaches the CCD by paths that sky light does not normally take, the resulting master flat will give poor results. Finally, because the quantum efficiency of CCDs is wavelength dependent, the spectral energy distribution of the light used for making flat-field frames should match that of the night sky, and should ideally *be* the light of the night sky. However, at good observing locations, the night sky is not bright enough to allow a deep exposure in a reasonable length of time.

Because of these factors, producing good flat fields is something of an art, and in some people's jaundiced view, akin to black magic. Techniques that work well for one observer may fail in the hands of another, for reasons that remain ob-

## Chapter 6: Image Calibration

scure. In preparing a telescope for CCD imaging, all internal surfaces should be meticulously blackened, all optics mounted securely, and the motions of all mechanical components (such as the focuser, filter slides, and auxiliary lens mounts) made tight and reproducible. Eliminating these common problems will greatly increase your chances of getting excellent flat-field frames.

### 6.2.3.1 Four Types of Flat-Field Frames

There are four generally accepted methods of making flat-field frames: light-box flats, dome flats, twilight flats, and sky flats. Light-box flats are made by placing a back-lit diffusing screen immediately in front of the telescope; dome flats by turning on lights in the dome and shooting a white screen mounted on the dome. Twilight flats are made by shooting the twilight sky, either dusk or dawn; and sky flats are made by taking the median of large numbers of images of the night sky. Each technique has its merits and demerits, but all of them are capable of producing good flat fields.

Light-box flats and dome flats offer the advantage of being under the control of the observer. With small instruments, the light box can be placed directly over the front of the telescope, and the box can be designed to produce any desired level of brightness. Dome flats make sense for large telescopes, where illuminating a screen attached to the inside of the dome or roll-off roof is more practical than a gigantic light box. However, it is very difficult to construct a light-box or dome illumination system whose spectrum matches that of the night sky.

Twilight flats demand quick work on the part of the observer, not only to determine when the sky has reached the right brightness, but also to take enough flat-fields to make a master flat with a good signal-to-noise ratio. Operationally, evening twilight flats are tricky because they may be taken before the telescope has been focused for the night, and dawn twilight flats require observers to stay awake until dawn. Twilight flats also record star images, so that in calibration, you must take a median to eliminate them.

Sky flats require huge amounts of observing time, or an observing program in which the images serve as their own flat-field frames. Sky flats do, of course, match the spectrum of the night sky, so in that respect they are ideal. However, because the sky is dark and full of stars, it is necessary to take the median of hundreds of sky flats not only to eliminate star images, but also to obtain an acceptable signal-to-noise ratio. Another consideration is that the imaging targets must be randomly positioned in the images, lest you end up with a hot spot in the center of the master flat. For an observing program that generates hundreds of images of small, faint, randomly placed objects, sky flats are ideal.

### 6.2.3.2 How to Shoot Light-Box and Dome Flats

Light-box flats and dome flats are easy to shoot. At a convenient time in the observing session, the light box is attached to the telescope or the dome rotated so that the telescope is pointed at the screen. If you are using the camera in a mode

## Section 6.2: Calibration Frames

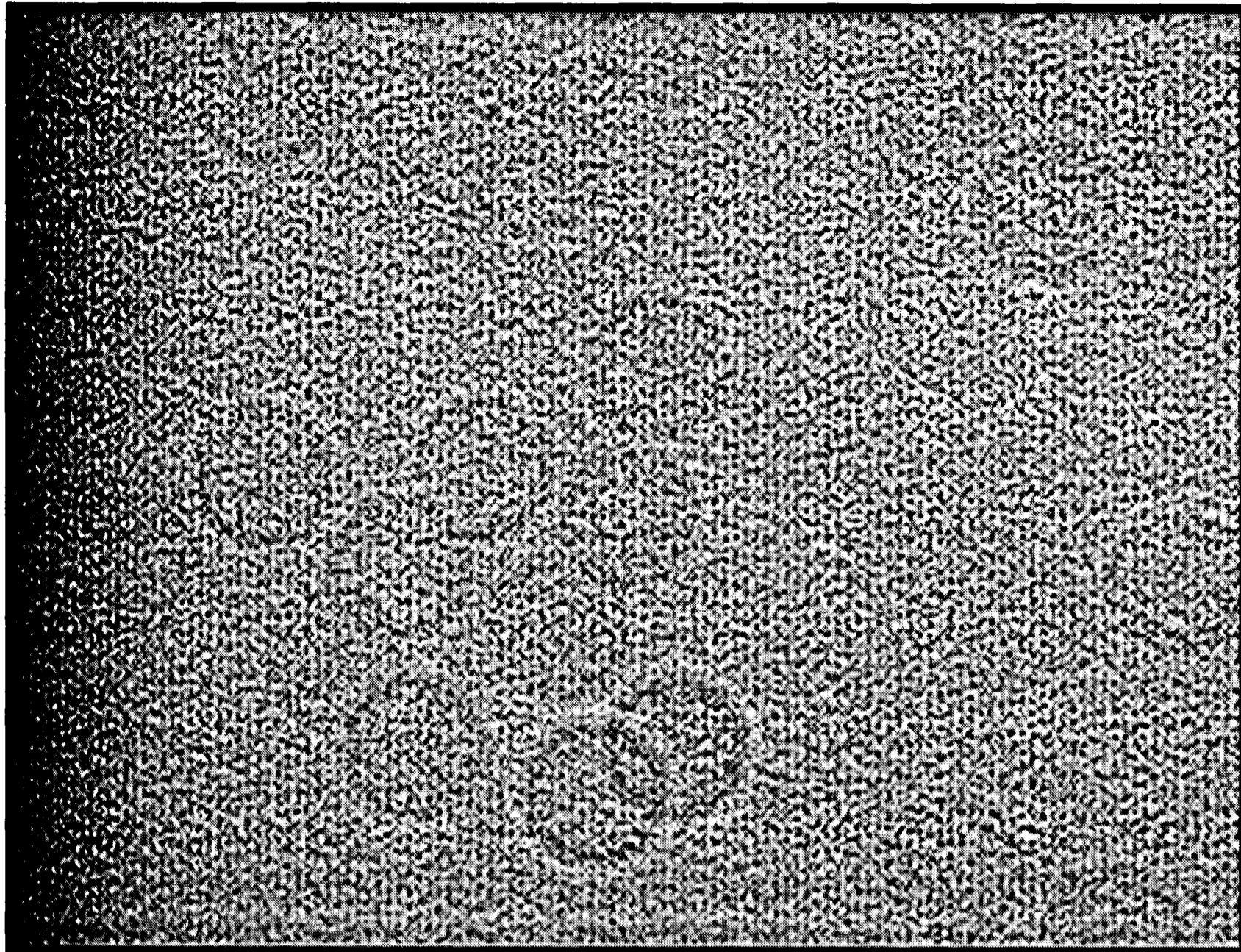


Figure 6.17 Above you can see low-contrast structure in the master flat-field revealed by local adaptive sharpening, a detail-extraction algorithm. The CCD shows small-scale and large-scale variations of about 0.2% in the sensitivity of columns. Flat-fielding removes artifacts like these from your images.

that reduces the well capacity, disable it and set the camera to make integrations of between two and 10 seconds. No other settings are changed. Make the flat-field frames as follows:

1. Switch on the lamps illuminating the dome screen or the light-box.
2. Adjust the lamp brightness and integration time to fill the charge wells in the photosites to half of their full capacity.
3. Shoot enough flat-fields to guarantee a high signal-to-noise ratio. With the CCD cameras used by amateur astronomers, 16 flats are ample.
4. Without changing anything else, turn off the lamps.
5. Shoot an equal number of flat darks.

The entire sequence can be completed in about 10 minutes.

Despite the disadvantage of spectral mismatch between the night sky and the lamps in your light-box or dome screen, with amateur telescopes and CCD cameras, light-box flats or dome flats produce excellent and consistent master flat-field frames.

## Chapter 6: Image Calibration

### 6.2.3.3 How to Shoot Twilight Flats

Twilight flats require careful planning. At the end of the previous observing session, leave the CCD camera on the telescope, and do not change the focus setting. Be sure the camera is fully cooled and operating normally. Then take test integrations of five seconds' duration. During evening twilight, the sky brightness halves every minute. As soon as the sky is dark enough to produce unsaturated images, begin taking and saving five-second integrations as fast as you can. At the end of three minutes, the sky will be too dark to continue. Cap the telescope and shoot as many flat darks as you shot twilight flats. If you plan to use the advanced calibration protocol, shoot your bias frames.

One little wrinkle has foiled many observers: when you're shooting twilight flats, shut off the telescope's drive motor so that stars form trails. If you have the drive running, the star images will become part of your master flat! If you let them trail, you can eliminate them by using a median combination.

To shoot dawn flats, reverse the process. Begin to integrate and save flats when the sky background reaches 5% of full-well capacity, and continue shooting and saving flats until the camera saturates. Cap the telescope and shoot as many flat darks as you shot twilight flats. If you observe all night, be sure to compare the dusk and dawn flats to see if the telescope and camera remained the same all night long.

Because the brightness of the sky changes, you cannot simply average sky flats. Instead, you must subtract an average dark frame from each image, measure the average brightness of a small region near the center of each frame, and then multiply the images so that all have the same average pixel value near the center. This process is called "normalizing" the images. Once the flats have been normalized, you can take the median pixel value (to eliminate the star images) from the set of dark frames as you create the master flat.

### 6.2.3.4 How to Shoot Sky Flats

Making sky flats is tricky because the telescope and CCD camera must remain the same, except for minor changes in focus, long enough to build a "flat library" of at least one hundred images. To be used as flats, images must not have bright objects near the center; and in all other respects they should be as similar as possible, using the same camera settings and integration times. Avoid using images shot near the horizon or on nights with strong Moonlight. Otherwise, a sky-flat library is just a large collection of normal images of small, faint objects that are not centered in the frame. In the course of making the images, shoot lots of dark frames and bias frames.

An observer who expects to use sky flats should be aware that the final calibration of an image may not be possible for days or weeks after it is made, because the raw images can be flat-fielded only when you have accumulated enough images to form a normalized median master flat with an acceptable signal-to-noise ratio. The sky background in sky flats is usually not bright, so the individual

frames are fairly noisy—and not all the same brightness. Later, when we compute the signal-to-noise ratio of a single short-exposure image of the night sky, you will see it's around 15:1. To reach the 600:1 signal-to-noise ratio you would expect in a master flat-frame made with a light box, you will need to take roughly 1,600 sky flats—too many to be practical for most amateur astronomers.

## 6.3 Methods of Calibration

The purpose of calibration is to remove, subtract, and correct the “instrument signature” added to a raw image by the CCD; that is, to peel away multiple onion-like layers of unwanted signal wrapped around the signal that you want. How you carry out calibration depends on what information you want from the final image. As an observer, you must decide what end product you want before the observing session begins, and then take the necessary support frames. Here is a summary of the output from each calibration type and the support frames required:

- **Basic Calibration:** Pixel value proportional to the *number of photoelectrons created in the CCD*. Requires a master dark frame made with the same integration time as the raw image.
- **Standard Calibration:** Pixel value proportional to the *number of photons coming from the astronomical target*. Requires a master dark frame made with the same integration time as the raw image, and a master flat frame.
- **Advanced Calibration:** Pixel value proportional to the *number of photons coming from the astronomical target*. Requires a master flat frame, a master dark made with the same or longer integration time as the raw image, and a master bias frame.

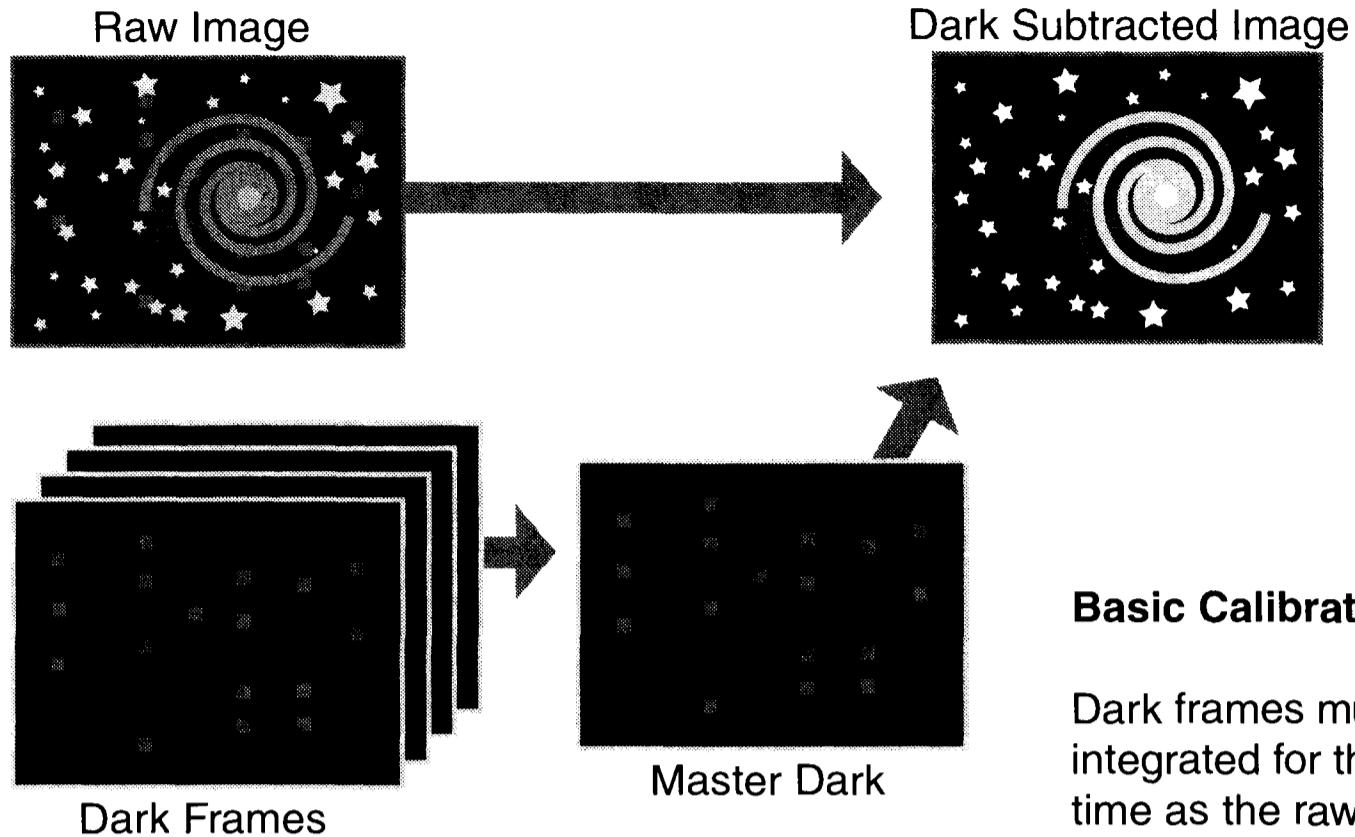
It is tempting for an observer to say, “I want nothing but the best. I will only calibrate images using the advanced protocol,” without considering how the images will be used. The goal of imaging is, after all, to discover something about the universe—perhaps scientific or perhaps aesthetic. If the goal of your observing program is producing supernova search images, basic calibration does a fine job. Standard calibration corrects for vignetting and CCD nonuniformity, but constrains you to the same integration time for your images and darks. Advanced calibration gives you the freedom to use different integration times for your images, and provides superior dark subtraction via dark-frame matching. When you plan your observations, select a calibration protocol appropriate for your imaging goal.

### 6.3.1 Basic Image Calibration

The basic calibration protocol approaches a raw image as the sum of some number of photoelectrons plus a dark frame:

$$\langle \text{RAW} \rangle_{x,y} = \frac{1}{g} E_{x,y} + \langle \text{DARK} \rangle_{x,y}. \quad (\text{Equ. 6.17})$$

## Chapter 6: Image Calibration



### Basic Calibration

Dark frames must be integrated for the same time as the raw images.

Figure 6.18 Basic calibration requires at least one, and preferably many, dark frames combined into a master dark frame. This protocol removes the bias level and dark current, but does not correct vignetting or CCD sensitivity variations. Basic calibration is adequate for simple observing tasks.

The goal of basic calibration is to extract, from the raw image, pixel values proportional to the number of photoelectrons generated on the CCD,

$$\frac{1}{g} E_{x,y} .$$

It does not correct the nonuniformity of the CCD or remove vignetting or shadowing.

It consists of subtracting a dark frame identical to the dark current information contained in the raw image; that is, a dark frame with the same integration time as the raw image. Although it is possible to perform basic calibration using a single dark frame, it is far better to generate a master dark frame by combining as many dark frames as you can conveniently shoot.

- **Tip:** In **AIP4Win** you can make a master dark frame by averaging or taking the median of many individual dark frames. Averaging multiple dark frames reduces dark noise by the square root of the number of frames combined. Taking the median eliminates cosmic ray events in the master dark, but results in a slightly noisier master dark frame.

If your raw images were made with a CCD camera without active temperature control, the dark current and bias level embedded in the dark frame may change over time. If this is the case, it is best to combine dark frames taken before and after the raw images when you use the basic calibration protocol.

If your raw images were made with a temperature-controlled CCD camera with bias drift subtraction, you can probably use dark frames shot several hours before or after your raw images were taken. Temperature control keeps the dark

current constant, and bias drift subtraction determines the true bias during each readout and keeps the bias in the images you shoot at a constant value.

### 6.3.2 Standard Image Calibration

The standard calibration protocol treats the raw image as a photon flux,  $I_{x,y}$ , modified by quantum efficiency variations,  $Q_{x,y}$ , and vignetting,  $V_{x,y}$ , plus a dark frame:

$$\langle \text{RAW} \rangle_{x,y} = \frac{1}{g}(tV_{x,y}Q_{x,y}I_{x,y}) + \langle \text{DARK} \rangle_{x,y}. \quad (\text{Equ. 6.18})$$

The goal of standard calibration is to extract the term:

$$\frac{1}{g}(t\bar{Q}I_{x,y})$$

from the raw image, where  $\bar{Q}$  (read as “Q-bar”) is the average quantum efficiency of the CCD array. Preparation for calibration consists of the following steps:

1. Averaging or taking the median of the dark frames to create a master dark frame. Note that the integration time used for the dark frames must be the same as the integration used for the raw images.
2. Averaging or taking the median of the raw flat-field frames to create a combined raw flat-field frame.
3. Averaging or taking the median of the flat-field darks to create a combined flat-dark frame.
4. Subtracting the combined flat dark from the combined raw flat to produce a master flat-field frame.

Once the master dark and master flat have been made, they can be used for all of the raw images made using the same integration time and optical setup. Calibration itself consists of the following steps:

1. Subtracting the master dark from the raw image.
2. Computing the average pixel value of the master flat-field frame.
3. Dividing, on a pixel-by-pixel basis, the dark-subtracted image by the ratio of the master flat-field pixel value over the average pixel value of the central region of the master flat-field frame.
4. Saving the resulting calibrated image.

Although flat-fielding is sometimes called “division,” the actual operation is more complex. We assume that a region near the center of the master flat-field is free of vignetting and has a quantum efficiency typical of the entire CCD, and find the average pixel value of this region. The size of the averaged region can be as small as 100 pixels on a side, or the central 50% of the frame; as long as the flat-field is reasonably uniform, it doesn’t matter. This average value,

$$\langle \overline{\text{FLAT}} \rangle,$$

## Chapter 6: Image Calibration

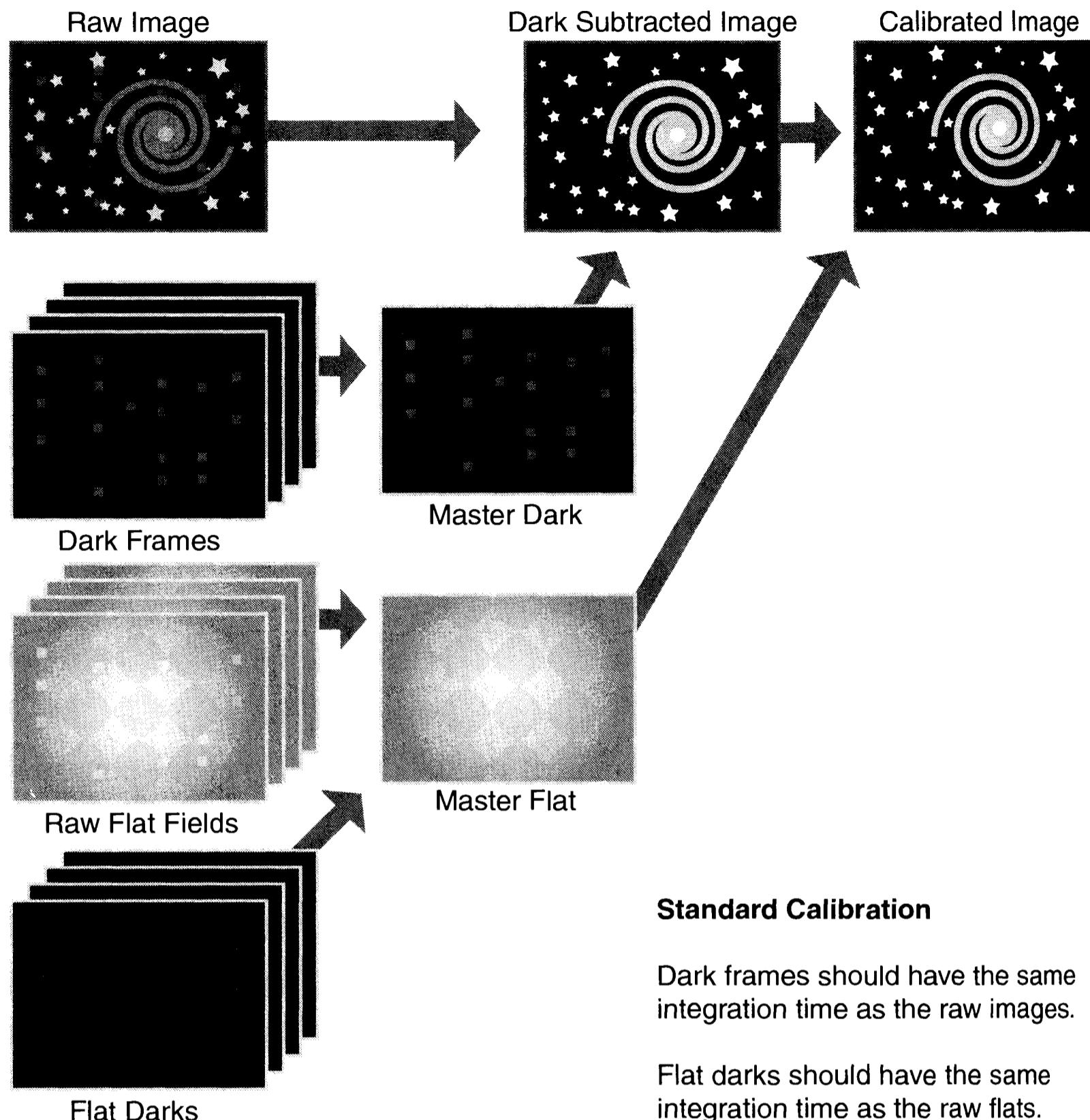


Figure 6.19 Although standard calibration requires multiple darks, flats, and flat darks, it produces an image corrected for bias, dark current, and vignetting. Dark frames must have the same integration time as that of your images. Standard calibration does an excellent job for most types of CCD imaging.

forms a standard for the rest of the master flat:

$$\langle \text{CALIB} \rangle_{x,y} = \left( \frac{\langle \overline{\text{FLAT}} \rangle}{\langle \text{FLAT} \rangle_{x,y}} \right) (\langle \text{RAW} \rangle_{x,y} - \langle \text{DARK} \rangle_{x,y}). \quad (\text{Equ. 6.19})$$

If the flat were perfectly uniform, the ratio of the average value to the value of the individual pixel would always be one. If a particular photosite has low sensitivity or is shaded by a speck of dust, then the pixel value in the master flat will be low. When the ratio is computed, its value will be greater than one and the value of the corresponding pixel in the raw image will be raised, correcting its value to what it should have been.

- **Tip:** *AIP4Win takes care of the details of flat-fielding for you. All you have to do is select the raw flats and flat darks to make the master flat-field frame.*

### 6.3.3 Advanced Image Calibration

The aim of the advanced calibration protocol is the same as standard calibration: to extract the quantity

$$\frac{1}{g}(t\bar{Q}I_{x,y})$$

from the raw image. However, this procedure seeks to give the observer greater flexibility by removing the constraint of equal integration time for images and darks. To an observer contemplating using the advanced calibration protocol, the raw image looks like this:

$$\langle \text{RAW} \rangle_{x,y} = \frac{1}{g}\{(tV_{x,y}Q_{x,y}I_{x,y}) + (td_{x,y} + \sigma_{\text{TE}}) + \langle \text{BIAS} \rangle_{x,y}\} \quad (\text{Equ. 6.20})$$

where  $g$  is the conversion factor,  $t$  is the integration time,  $I_{x,y}$  is the photon flux,  $Q_{x,y}$  is the quantum efficiency,  $V_{x,y}$  is the vignetting factor,  $d_{x,y}$  is the dark current, and  $\sigma_{\text{TE}}$  is the thermal noise. The crucial change in thinking is that the bias is now seen as separate from the accumulated dark current instead of being lumped together as part of a dark frame.

Now consider a dark frame in which the integration is  $t_{\text{DK}}$  instead of  $t$ :

$$\langle \text{DARK} \rangle_{x,y} = \frac{1}{g}\{(t_{\text{DK}}d_{x,y} + \sigma_{\text{TE}}) + \langle \text{BIAS} \rangle_{x,y}\}. \quad (\text{Equ. 6.21})$$

By subtracting a bias frame, the accumulated dark current during the time  $t_{\text{DK}}$  can be converted to the dark current that would have accumulated during the raw image integration time,  $t$ . Follow it step-by-step. First, subtract the bias frame:

$$\langle \text{DARK} \rangle_{x,y} - \langle \text{BIAS} \rangle_{x,y} = \frac{1}{g}\{(t_{\text{DK}}d_{x,y} + \sigma_{\text{TE}})\}. \quad (\text{Equ. 6.22})$$

Note that if the bias frame is “clean,” and the bias value can be determined accurately, you can perform these steps using the single bias value instead of using a master bias frame. Multiply by the ratio of the image integration time divided by the dark frame integration time to obtain a scaled thermal frame:

$$\langle \text{SCALED THERMAL} \rangle_{x,y} = \left\{ \frac{t}{t_{\text{DK}}} \right\} \left\{ \frac{1}{g}\{(t_{\text{DK}}d_{x,y} + \sigma_{\text{TE}})\} \right\}. \quad (\text{Equ. 6.23})$$

Providing the integration time for the dark is longer than the integration time for the image, the thermal noise is reduced by scaling:

$$\langle \text{SCALED THERMAL} \rangle_{x,y} = \frac{1}{g} \left( t d_{x,y} + \frac{t}{t_{\text{DK}}} \sigma_{\text{TE}} \right). \quad (\text{Equ. 6.24})$$

The scaled thermal frame contains the same accumulation of dark current that it would have had if the integration time were the same as the raw image.

## Chapter 6: Image Calibration

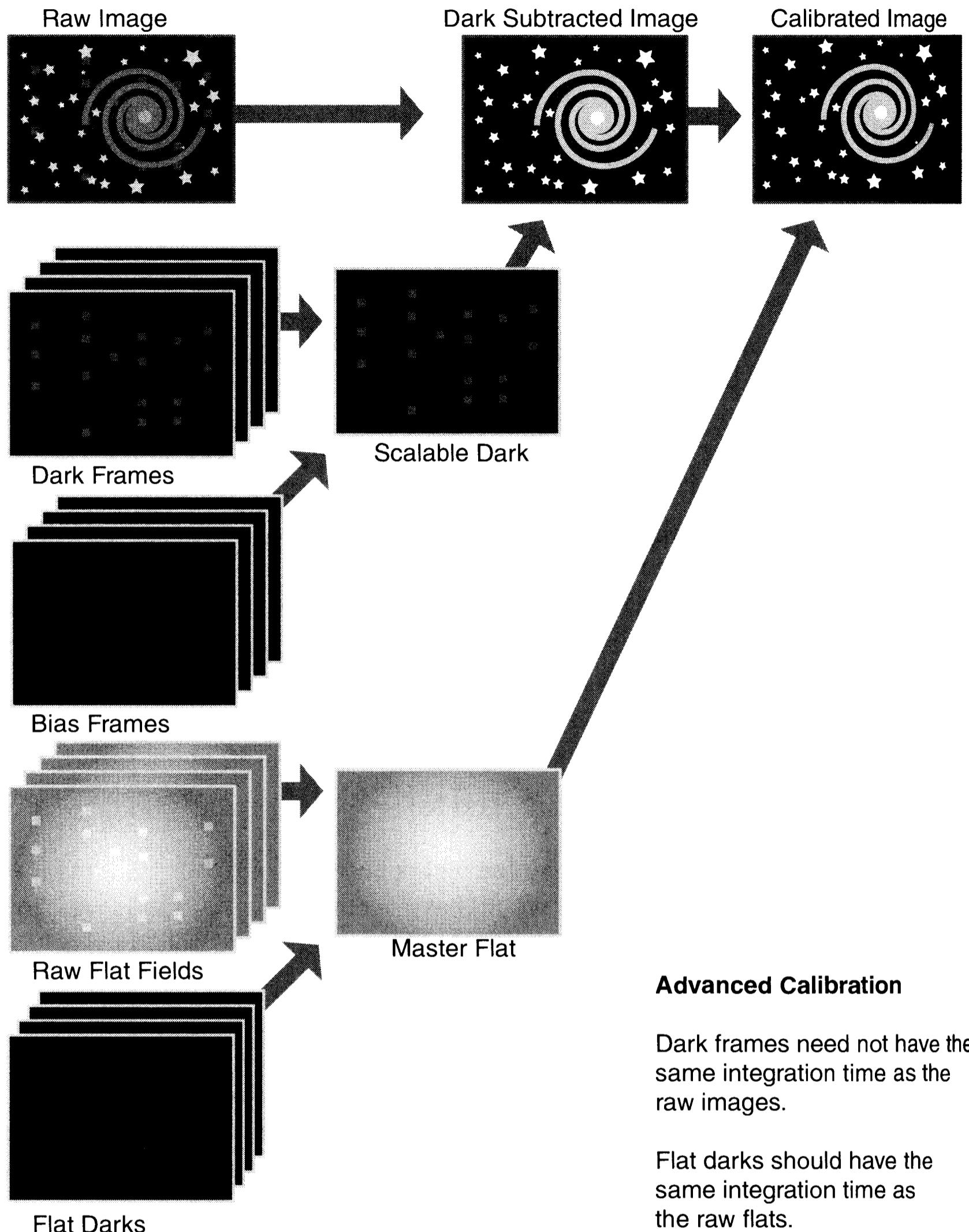


Figure 6.20 The advanced calibration protocol allows you to use one set of dark frames for calibrating all of your images, regardless of their integration times. Considering how easy it is, take the necessary bias frames; the added flexibility in integration times is a boon to the busy observer.

The next step is to subtract the scaled thermal frame and the bias frame from the raw image. After the double subtraction, the image consists of:

- $\frac{1}{g}(tV_{x,y}Q_{x,y}I_{x,y})$  plus the quadratic sum of the following;
- $\sigma_{\text{TH}}$  thermal noise from the image;

- $\frac{t}{t_{\text{TH}}} \sqrt{\frac{t_{\text{TH}}}{t}} \sigma_{\text{TH}}$  thermal noise from the scaled dark frame;
- $\sigma_{\text{RO}}$  readout noise from the image; and
- $\frac{t}{t_{\text{TH}}} \sigma_{\text{RO}}$  readout noise from the scaled dark frame.

The integration time used for the dark frame should be at least five times longer than the integration time used for the raw images. This strategy minimizes the value of the thermal noise and the number of readouts necessary.

- **Tip:** *In AIP4Win, you can enter the ratio of the integration times or allow the program to match the dark frames automatically.*

After dark subtraction, the image is flat-field corrected in exactly the same way as it is flat-fielded in the standard calibration protocol.

- **Tip:** *AIP4Win automates advanced image calibration. After the observer sets up the master dark, master flat, and master bias frames, and selects a method of scaling the thermal data, the current image can be calibrated with one click. Using the Auto-Process Tool, you can automatically calibrate all of your raw images from a long observing session.*

## 6.4 The Calibrated Image

The calibrated image should be an accurate, albeit sampled and quantized, copy of a section of night sky. In the ideal image, pixel values are directly proportional to the average flux of photons falling on the CCD during the integration. The most amazing thing—since we don’t live in an ideal world—is that real-world calibrated CCD images really do approach the ideal. Observers using rather ordinary small telescopes routinely reach 20th magnitude, and routinely pull down images good enough to yield sub-arcsecond astrometry and photometry with a precision of 0.02 magnitude over a range of five magnitudes. Truly these are inspiring times for amateur astronomy.

Despite the accomplishments of CCD imaging, it is important to look critically at the images we obtain, not only to see what’s wrong, but also to learn ways to improve them. As we shall see, after dark current is gone and less-than-perfect uniformity corrected, the noise contributions remain. In this section, we take a realistic look at the properties of calibrated images, and explore their limitations and their strengths.

### 6.4.1 Photons, Dark Current, and Readout Noise

When you take an image, each photosite on the CCD captures some number of photons; but because they arrive at random, if you take a whole series of images one right after the other, the number of photons captured by the very same photo-site will vary. In some images more photons than average will arrive, and in oth-

## Chapter 6: Image Calibration

ers, fewer than average will arrive. When  $\bar{P}$  is reasonably large, the variation in the number of photons obeys a simple law: it forms a roughly normal distribution about the mean value,  $\bar{P}_{x,y}$ . When we count up the samples, 68% lie within  $\sqrt{\bar{P}_{x,y}}$  of the mean value, 95% lie within  $2\sqrt{\bar{P}_{x,y}}$ , and 99% lie within  $3\sqrt{\bar{P}_{x,y}}$ .

While photon statistics might at first seem removed from astronomy, it most definitely is not: short-exposure CCD images look grainy *because* of photon statistics. Suppose that you have shot a beautiful 60-second integration of the Horsehead Nebula with your CCD camera, and, after careful calibration with a high-quality master dark and master flat, you realize that the sky background looks grainier than you would like. You scan a pixel-value tool over the image and read off a few dozen pixel values: you see some 66s, more 67s, lots of 68s, fewer 69s, some 70s, and the occasional 65s and 71s. Why is this happening? Is it right?

Let's look first at the photon statistics in this image by developing a specific example. The average pixel value in the sky is 68 ADUs. Suppose also that you have characterized your camera (see Chapter 8) and found that the conversion factor,  $g$ , is 35 electrons per ADU. A typical photosite exposed to the sky in this example would have generated  $68 \text{ ADU} \times 35 \text{ electrons per ADU} = 2380 \text{ electrons}$  during the integration. You look up the manufacturer's quantum efficiency curve and find that the average quantum efficiency of your CCD is 40%; so you extrapolate that 5950 photons fell on each photosite during the 60-second integration, at an average rate of 99 photons per second. During any one second, an average of  $99 \pm 10$  photons fell on each photosite; during the entire integration,  $5950 \pm 77$  photons fell on each sky pixel.

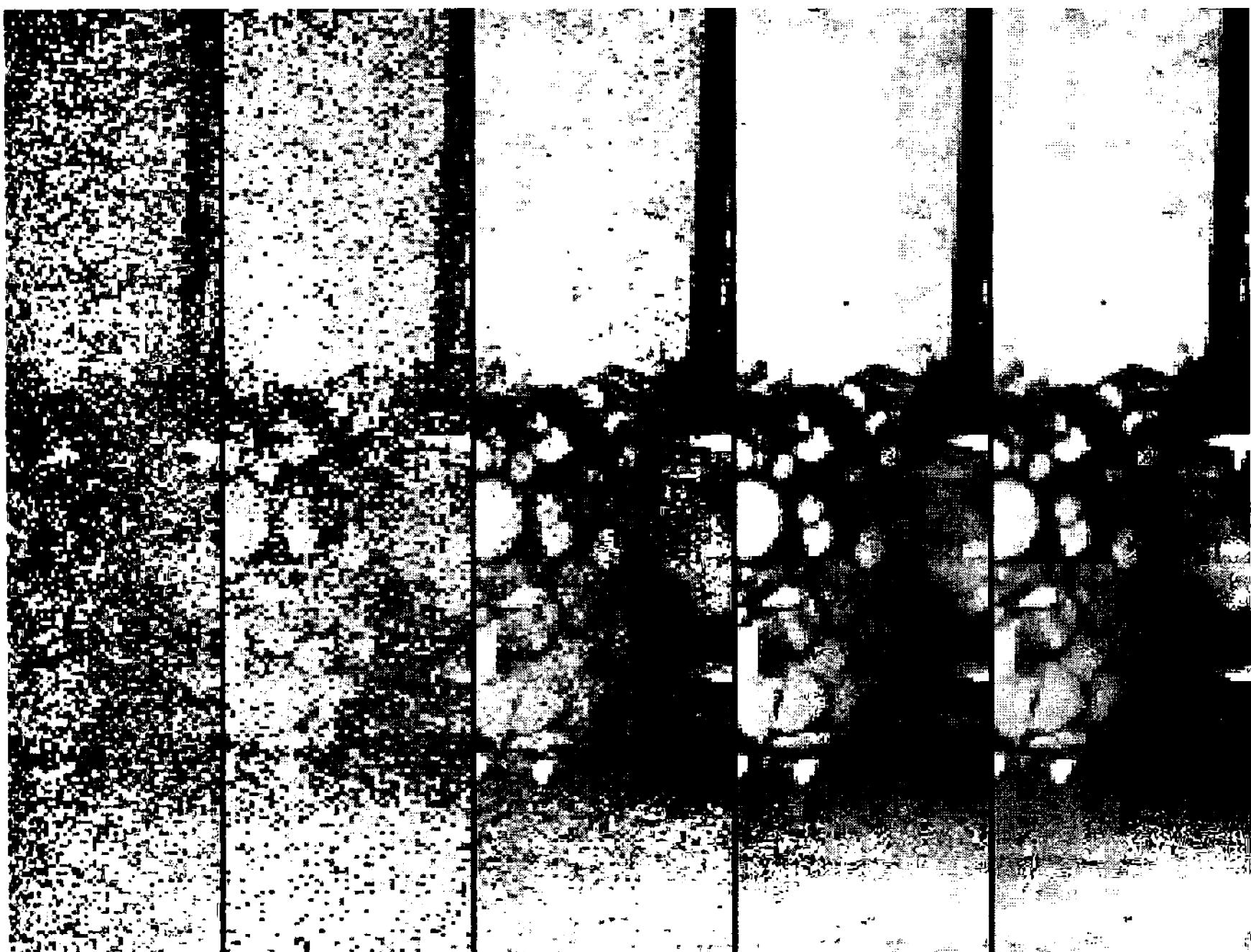
However, you don't know anything about photons—what you actually have is a sample of 2380 photoelectrons created when the photons struck the CCD. The statistical uncertainty in a sample of 2380 electrons is 49 electrons; that is, your measured signal is  $2380 \pm 49$  electrons. When you convert electrons back into ADUs using the conversion factor of 35 electrons per ADU, you find that the pixel value of the sky, *from photoelectron statistics alone*, should be  $68 \pm 1.4$  ADUs. In other words, part of the variation that you see in the sky background is due to the Poisson statistics having to do with random arrival of photons.

Dark current has also made a contribution to your Horsehead image, although its contribution at most photosites turns out to be quite small. With a measured rate of 1.5 electrons per second per photosite, you expect an average of 90 electrons to have accumulated during the 60-second integration of the image, for a thermal noise contribution,  $\sigma_{TE}$ , of  $\sqrt{90} = 9.5$  electrons.

The CCD also has a small population of “hot” pixels. The “hottest” of these generate roughly 250 electrons per second, for an accumulation averaging some 15,000 electrons during integration. With an expected standard deviation of 122 electrons, those oddball photosites are likely to result in a few light or dark pixels even after calibration.

Finally, when the accumulated electrons were clocked off the CCD and converted to a voltage, the charge detection node of the CCD added readout noise,  $\sigma_{RO}$ , a small, random variation generated by it. For this example, let's assume that

## Section 6.4: The Calibrated Image



**Figure 6.21** For good image quality, you need to accumulate enough photons to give a statistically “clean” image. From left to right, you see exposures of 1 millisecond, 10 ms, 100 ms, 1 second, and 10 seconds. When there are enough photons, you can see a wood pile beside a wood-burning stove.

you have measured the readout noise in your camera (using the methods described in Chapter 8) and found that it has an r.m.s. readout noise of 31 electrons.

Capturing the raw image included these noise sources:

- from photon statistics:  $\sigma_p = 49$  electrons;
- from readout noise:  $\sigma_{RO} = 31$  electrons;
- from thermal noise:  $\sigma_{TE} = 9.5$  electrons;
- in the “hot” pixels:  $\sigma_{HP} = 122$  electrons.

To find the total noise of a series of independent noise sources, square them, sum them, and take the square root (this is what “summing in quadrature” means):

$$\sigma_{\text{total}} = \sqrt{\sigma_p^2 + \sigma_{RO}^2 + \sigma_{TE}^2}. \quad (\text{Equ. 6.25})$$

Noise adds in this manner because the individual noise contributions add together or cancel randomly. The total noise is:

$$\sqrt{49^2 + 31^2 + 9.5^2} = 58.8 \text{ [electrons]}. \quad (\text{Equ. 6.26})$$

Converting 58.8 electrons to ADUs, you see that the sky background contains a variation of  $\pm 1.7$  ADUs of noise due to capturing the image. The signal was



**Figure 6.22** Case study: One raw image of the Horsehead Nebula. In addition to the nebula, the raw image contains the bias offset, dark current, and several dark dust shadows. In addition to 44 raw images, the observer shot 20 bias frames, 60 dark frames, as well as flats and flat-darks.

large enough that in normal pixels, photon and photoelectron statistics made a greater contribution to the “graininess” in the sky background than any other cause. For the 100 or so hot pixels, the standard deviation works out to 132 electrons, or about 4 ADUs.

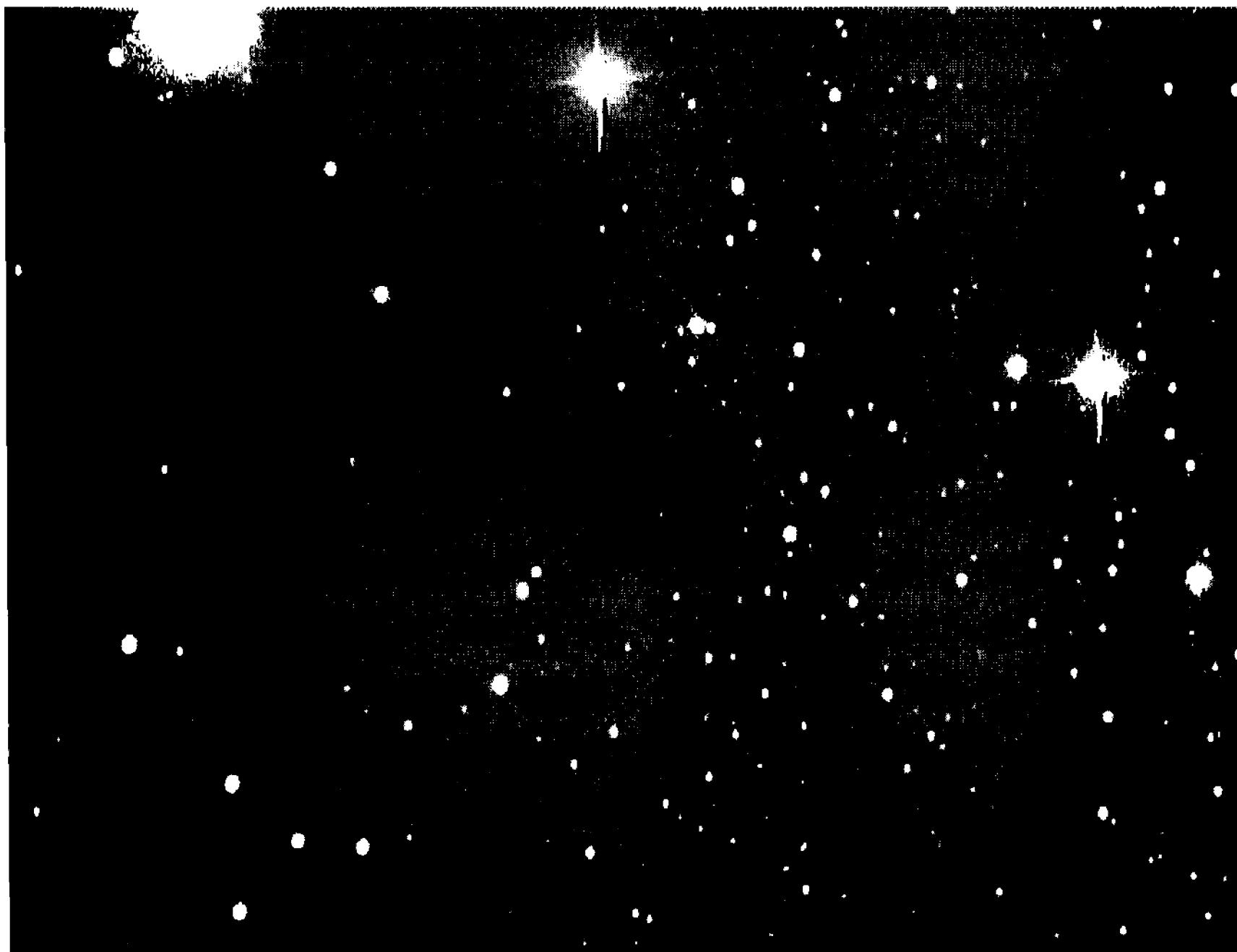
We next assess the noise contribution from calibrating the image.

### 6.4.2 Noise from Calibration

Dark-frame subtraction and flat-fielding add noise because dark frames and flat frames are themselves noisy—although it is fair to say that dark frames are the greater culprit, since a well-done flat-field is uncertain by only one part in 600.

Let’s see how bad it is. When you took the Horsehead image, you made an effort to produce an excellent master dark frame by shooting ten dark frames with 300 seconds integration each. The noise in a single dark frame consists of the following:

- The normal photosites have a dark current of 1.5 electrons per second, in 300 seconds an average of 450 electrons accumulated, so  $\sigma_{TE} = 21$  electrons.



**Figure 6.23** Here is the same image, now with the master dark frame subtracted using a dark-matching algorithm. Because the temperature of the camera changed slightly, the formula used was  $\langle \text{RAW} \rangle - 1.0623 * \langle \text{THERMAL} \rangle$ . Although the hot pixels have been subtracted, the dust shadows remain.

- Each dark frame contains a sample of readout noise. You already know that  $\sigma_{\text{RO}} = 31$  electrons.
- The small number of “hot” photosites have a dark current of 250 electrons per second, so in 300 seconds these accumulate 75,000 electrons with a noise level of  $\sigma_{\text{HP}} = 274$  electrons.

The noise expected from a single normal photosite is therefore 37 electrons, and in a hot photosite, 275 electrons. Averaging the ten dark frames that you took consists of two operations: summing the noise from the ten dark frames, as in Equation 6.25, and then dividing the noise by ten:

$$\sigma_{\text{average}} = \frac{\sqrt{10 \times \sigma_{\text{single}}^2}}{10} = \frac{\sigma_{\text{single}}}{\sqrt{10}}. \quad (\text{Equ. 6.27})$$

When you run the numbers,  $\sigma_{\text{average}} = 12$  electrons. In calibrating the raw image, the noise from dark subtraction is added in quadrature, increasing the noise from  $\pm 58.6$  electrons in the raw image to  $\pm 60$  electrons in the calibrated image. In the hot pixels, the noise from the dark frame works out to 87 electrons; which, when added in quadrature to the 148 electrons of hot pixel noise in the raw image, is 172 electrons r.m.s., or about 5 ADUs.

## Chapter 6: Image Calibration



**Figure 6.24** After flat-field correction, the dust shadows are gone! Although a single calibrated image often looks rather grainy, the image still shows a lot more than most observers could see visually with a much larger telescope. The next step is to build a stronger signal by combining many such images.

The bottom line is that carefully done dark-frame subtraction does not add significantly to the noise in the image; and, of course, it *does* remove the dark current. In the raw image, “hot” pixels have values around 500 ADU, and are, in fact, the most obvious feature therein. After calibration, the dark-subtracted hot pixels have the same *average* value as the sky—68 ADUs; but their standard deviation is  $\pm 5$  ADUs, just about twice that of a normal pixel. Although the hot pixels have been removed on the average, they are in fact a population of unusually noisy pixels, considerably more likely to have abnormally high or low values compared to the normal pixels around them.

In flat-fielding images of deep-sky objects with light-box and dome flats, noise is simply not a problem. Because of the full exposure given to the raw flats and the averaging of many frames, the statistical uncertainty of a pixel in a well-made master flat frame is one part in 600 or better. When the image is flat-fielded, the application of the flat frame to a sky-background pixel generates an uncertainty of about 0.14 ADUs. Since that sky pixel already has an uncertainty of  $\pm 1.7$  ADUs, the noise from flat-fielding is inconsequential. Flats are difficult to make, but the difficulties stem from reasons other than noise.

### 6.4.3 Image Stacking

Whenever photon statistics dominate the noise in a single image, you can achieve a better signal-to-noise ratio by summing multiple images. Providing the images are independent samples of the photon flux, the signal in the resulting image increases with the number of frames added, while the noise increases only as the square root of the number of images. Thus, the signal-to-noise ratio improves with the square root of the number of images summed. In the photon-noise dominated case, shooting many short integrations will produce the same quality image as making a single long integration.

When another source, such as readout noise, is dominant, the improvement in the signal-to-noise ratio is limited not by photon statistics, but by the other noise sources. You cannot stack hundreds of short-exposure images that are dominated by readout noise and expect to create an image that even approaches the quality of a single long integration. However, the signal-to-noise ratio will grow as you add more and more images—but with significantly more noise than you would accumulate with a single long integration.

The key to effective “stacking” is that each image must be statistically independent. Often, however, you must use the same master dark frame to calibrate each one, so that some portion of the noise in each is correlated with noise in the other images, and the gain in image quality is lower—but that difficulty is partially offset when the individual images contain tracking errors and must be shifted into registration. If the noise in the dark frame is a significant fraction of the noise in each image, however, it will show up as a pattern in the sky background. For good results, therefore, you must create a master dark frame with at least twice the total integration time as the total integration used in the stacked images, so that dark noise makes a negligible contribution to noise in the image.

When used properly, summing images (“stacking”), or registering and then summing (i.e., “track-and-stack”), is a powerful method for shooting very deep images. For the best results, the individual exposures should be as long as practical, and the total integration time for the master dark frame should exceed the total image integration time by a factor of two or more.

### 6.4.4 How to Spot Calibration Errors

A variety of errors may occur in the process of taking support frames and calibrating images. Underlying these are three problems that account for most image calibration faults: bias drift, changing CCD temperature, and changing optical configuration. This section gives a brief overview of common problems and how to spot them.

#### 6.4.4.1 Bias Drift

If the electronics in your CCD camera have temperature-sensitive components, the bias value can change when the air temperature changes. As long as the raw

## Chapter 6: Image Calibration

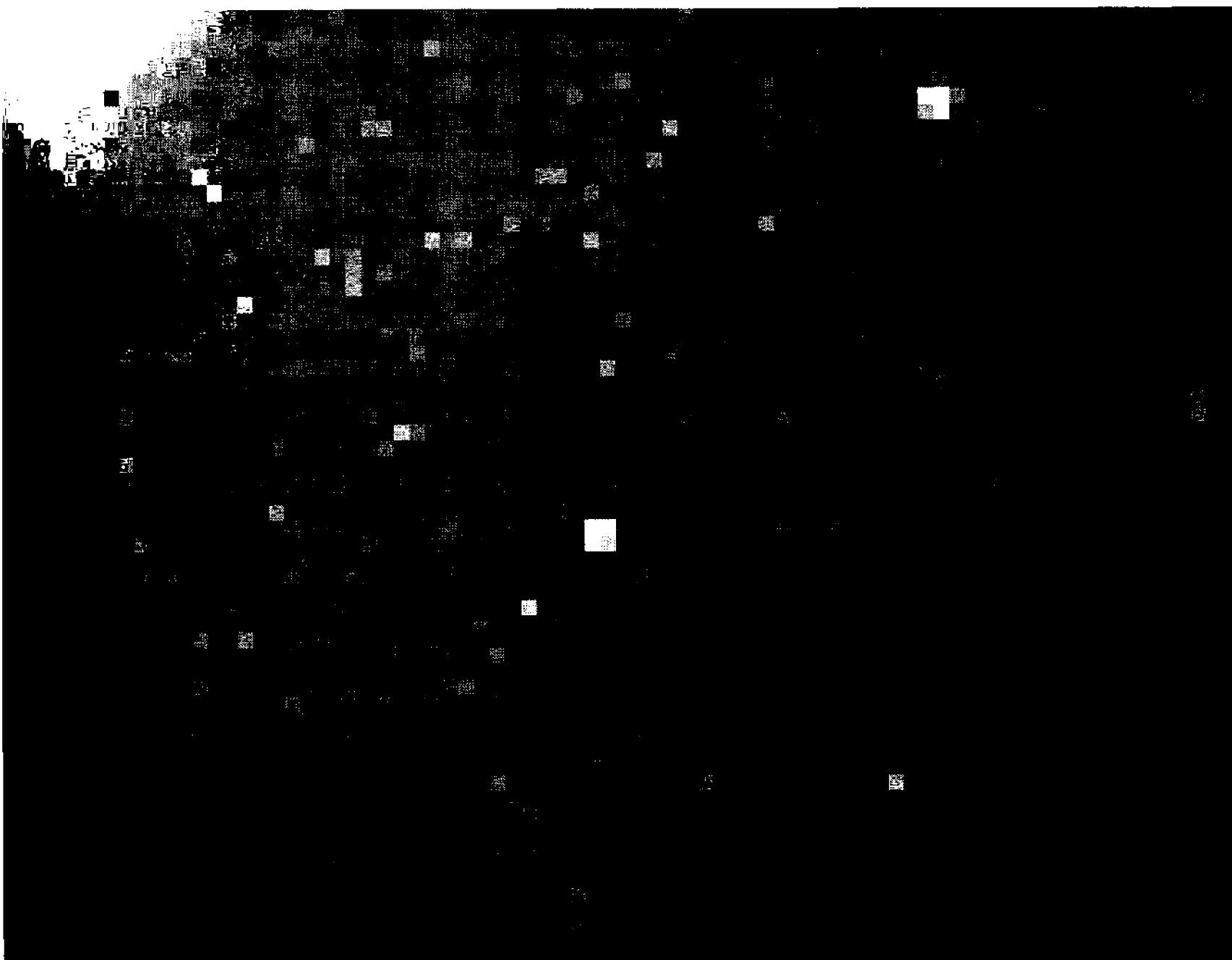


**Figure 6.25** Stacking images makes use of more photons, thereby increasing the signal-to-noise ratio. In this image, 44 integrations of 60 seconds each reveal the Horsehead in all its glory. Features that were barely visible in one 60-second integration stand out strongly when many integrations are combined.

images and the dark frames have the same bias value, no harm is done; but when the bias changes with time, significant errors can result.

- If the bias value in the raw image is higher than in dark frames, pixel values in the image will be too high. When a properly made flat-field is applied, vignetting is only partially corrected. This problem is difficult to spot unless flat-fielding under the standard or advanced protocols produces undercorrected flat-fielding.
- If the bias value in the dark frames is higher than it is in the raw images, pixel values in the images will be low. If the drift is severe, the entire sky background can have a zero or negative value. If the drift is small, when a properly-made flat-field is applied, vignetting is overcorrected. A completely black sky, bright corners, or bright dust shadows are clear indicators of bias increase.

CCD cameras that feature drift subtraction or automatic bias correction are immune to this problem. In these units, the bias is measured each time the CCD is read out, and pixel values in the image are adjusted to have a constant bias value.



**Figure 6.26** This enlarged section of an image displays both “hot” and “cold” pixels after calibration. This is evidence that the image frames and dark frames were taken at different temperatures, or that the CCD was operated with an active antiblooming gate, resulting in dark frames that were nonlinear.

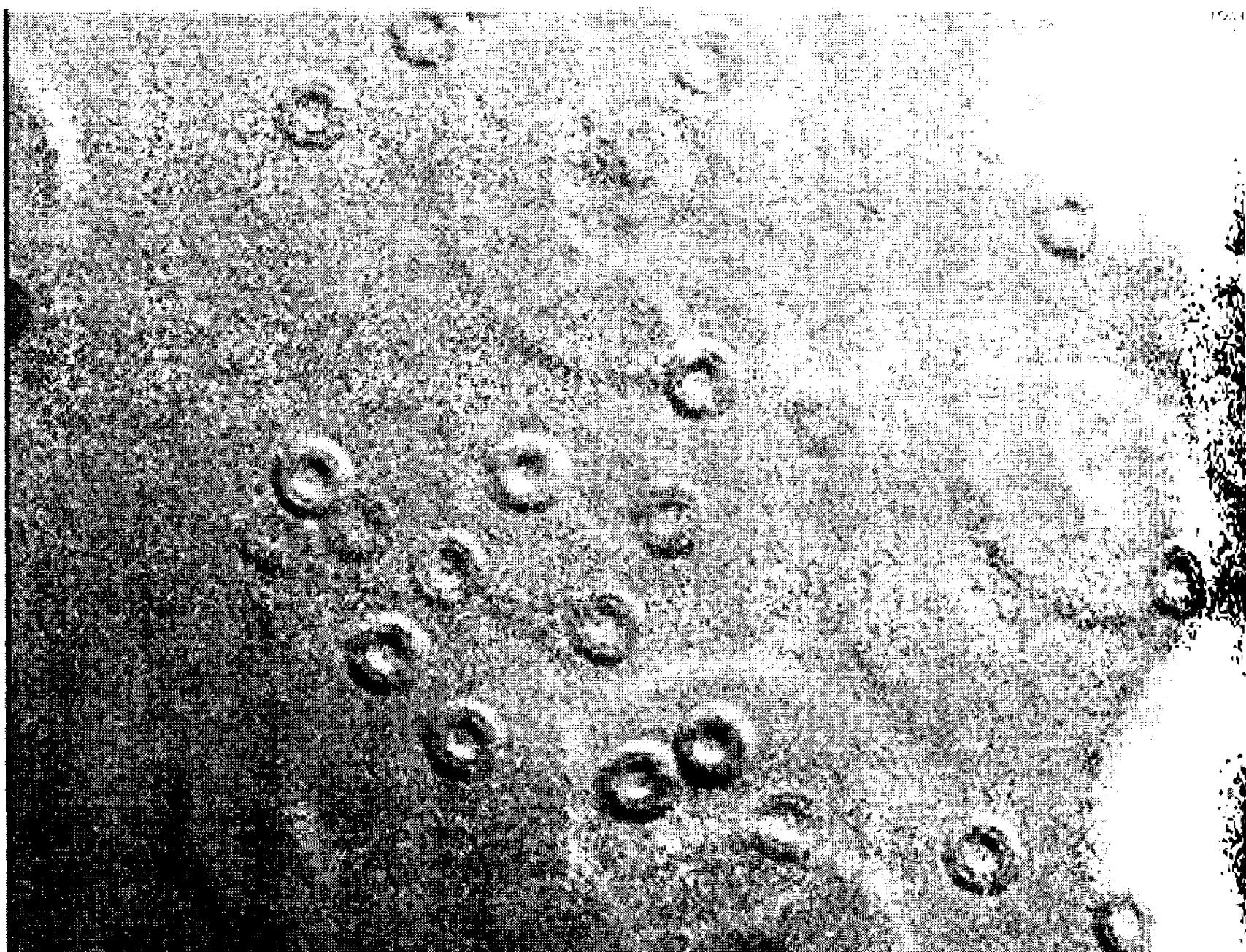
#### 6.4.4.2 Changing CCD Temperature

If the temperature of the CCD changes between the time you take the raw image and the time you take your dark frames, the dark current will change. Dark current is quite sensitive to the temperature of the CCD, so a change of one or two degrees Celsius can make a big difference.

- If the raw image was taken with the CCD at a higher temperature than the dark frames, the master dark frame will have recorded too little dark current; and you will see a residual population of hot pixels after dark subtraction. A residue of hot pixels points to temperature drift in the CCD.
- If the dark frames were taken with the CCD at a higher temperature than the raw images, the master dark frame contains pixel values that are too high for the raw image; and you will see a population of oversubtracted, or “cold” pixels. Cold pixels are a clear diagnostic for CCD temperature drift.

CCD cameras with actively stabilized CCD chip temperature are obviously desirable, but the observer should be aware that a poorly designed temperature stabilization circuit may result in a temperature that oscillates between too high and

## Chapter 6: Image Calibration



**Figure 6.27** Even though “nothing was changed” from one night to the next, something did change. Here is a flat-field, made on December 22, that has been flat-fielded using a flat-field made on December 21. What probably happened is that the CCD camera shifted slightly when it was refocused on December 22.

too low. For cameras that use water cooling without active temperature stabilization, use the largest practical volume of coolant water. For air-cooled cameras, use a fan to promote a constant rate of heat dissipation.

“Dark matching” partially corrects CCD temperature drift because it computes a dark-current scaling factor based on the properties of the hot pixel population. The characteristic appearance of a temperature drift that has been corrected by dark matching is a mixture of hot and cold pixels in the same image.

### 6.4.4.3 Changing Optical Configuration

Good flat-fielding demands an optical and mechanical system that stays the same between the time that images are taken and the time that flat-field frames are made. Other factors being equal, screw-thread couplings are preferable to slide-in adaptors, and refractors are preferable to reflectors and catadioptrics. However, the most important factors in getting good flat-fields are an observer who pays close attention to the mechanics of the telescope and CCD camera, and frequent cross-checks to insure optical stability and repeatability.

Here are some common problems that you may encounter in flat-fielding:

## Section 6.5: Defect Mapping and Correction

- Center-to-edge gradients in flat-fielded images usually result from scattered or stray light either in the flat-fielding setup or in the normal imaging setup. To correct this, blacken all interior surfaces and install baffles to prevent low-angle reflections inside the focuser tube, filter assembly, and camera housing.
- Right-to-left and top-to-bottom gradients in flat-fielded images can often be traced to non-uniform illumination of the dome screen or the light box. If you have a light box, compare flat-fields made before and after rotating it; if you have a dome screen, change the illumination and make comparison images.

When you flat-field raw images, examine the results critically and with an open mind. Remember that overcorrected and undercorrected flat-fielding will result if the bias level changes in your raw images. However, don't blame everything on bias frames; good flat-fielding techniques take some time to work out, and you need to check yourself and your equipment. If your flats aren't working right, try these tests with light-box and dome flats:

- Take data for two master flats, one right after the other without changing anything. Flat-field the first master flat with the second one; the result should be perfectly flat. Any departure from a uniformly illuminated image suggests that your flat-making technique or processing of the data is wrong.
- Take data for one master flat, point the telescope in all different directions, and then take data for a second one. When you flat-field the first with the second, the resulting image should be perfectly flat. Variation suggests movement of elements in the optical system.
- Take data for one master flat, rotate the telescope tube 90 degrees, and then take data for a second master. Any difference suggests that gravity is causing one or more elements in the optical system to sag or shift.
- Take data for one master flat at the beginning of your observing session and then take data for a second at the end of the session. Flat-field the first with the second flat. Expect some movement of dust particles. Any other changes suggest movement in the optical system.
- If you leave your CCD camera on the telescope all the time, compare master flats taken a few days or weeks apart. Changes should be minor; on the order of 1%.

## 6.5 Defect Mapping and Correction

Strictly speaking, defect mapping and correction are not part of the calibration process, but because defect correction is sometimes performed immediately after

## Chapter 6: Image Calibration

calibration, we include it in this chapter. CCDs are not perfect: they have single photosites, clusters of photosites, and even whole columns that either do not respond to light, or that respond in a markedly different way than normal photosites. These defects pass through calibration because the photosites do not behave normally, and appear in calibrated images as dark or bright pixels, clusters of pixels, or columns. They appear in every image frame regardless of exposure.

Defect mapping consists of identifying the defective pixels, generating a list of the defects, or creating a defect map—a special image in which the pixel values classify the defects by type. For example, a value of 1,000 might mean the problem is a single pixel defect; a pixel value of 2,000, that the pixel is in a defect cluster; and a value of 3,000, that the pixel is part of a defective column of pixels.

To correct the defective pixels, image processing software scans the defect map and applies a corrective action to the image based on the pixel values of the defect map. Upon detecting the code for a single defective pixel, the software might determine the median value of pixels surrounding the defect, and replace its value with the new one. For a cluster, the software could determine the median of the surrounding region and replace each pixel in the cluster with the new value; and for a defective column, the software could repair each pixel in the column with the median of pixels from adjacent columns.

Strictly speaking, defect correction does not actually “correct” or “repair” defective pixels—it simply discards the original “defective” pixel value and finds a value characteristic of the neighborhood to replace it. Because of this, defect-corrected images should not be used with those intended for astrometry or photometry. However, they certainly look much nicer than images with pixel, cluster, and column defects.

---

# 7 Image Analysis

---

Locked within the numerical values that make up a calibrated CCD image is a staggering amount of information. To access it, we use a series of software tools—routines crafted to extract the specific types of information that astronomers want from images. These fall into three categories:

- information about *pixels*,
- information about the *entire image*, and
- information about *regions of interest*.

Pixel information consists of location and pixel value—the address and telephone number for each individual speck in an image.

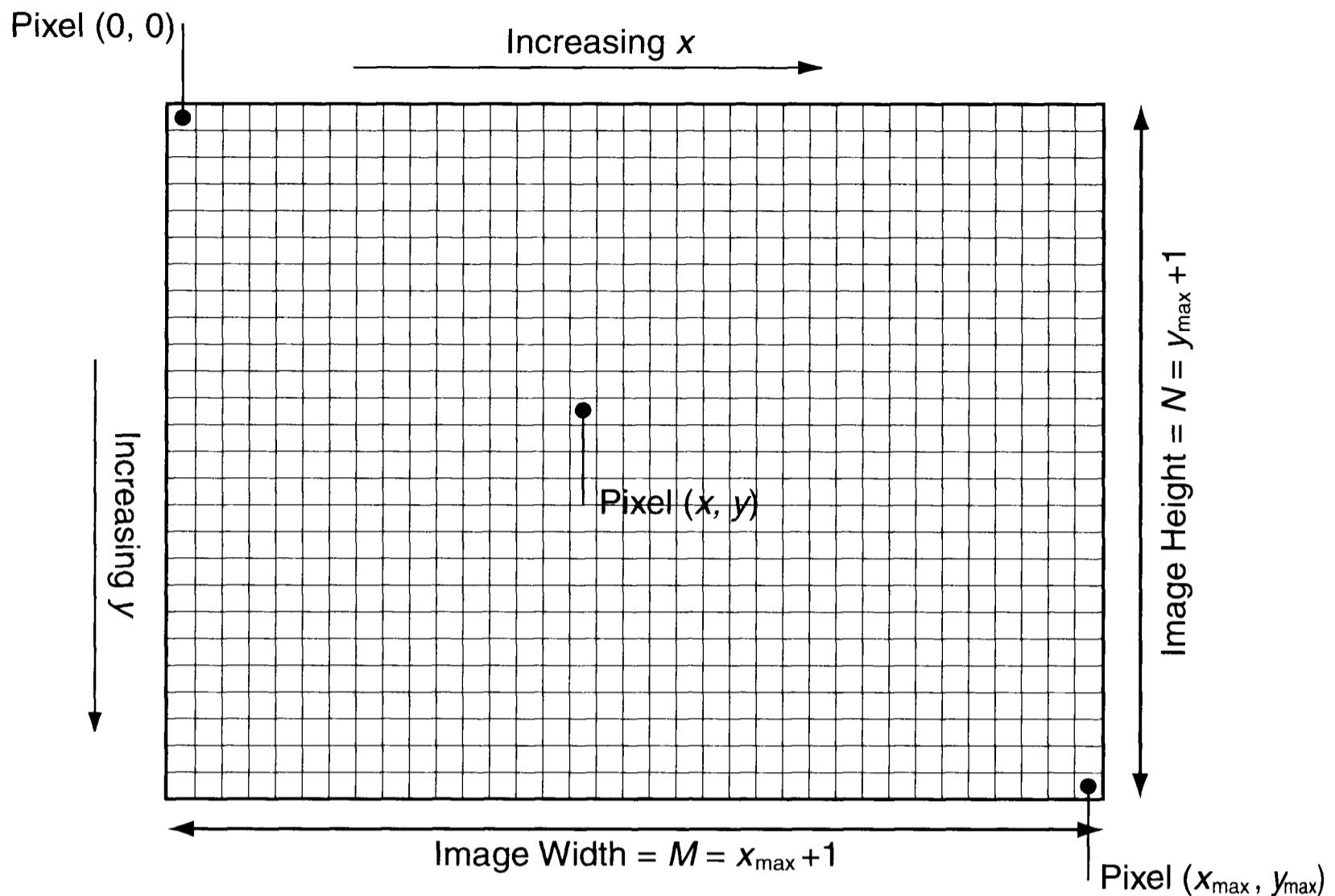
Information about the entire image is like a survey of all the people in a community—it's about pixel demographics.

The pixels in a region of interest are related, like people in a family. A pixel family might be a star image; the information the family shares is the star's brightness and position.

In this chapter, you will learn what the standard image measurement tools are, what they measure, and what you can do with that information. Much of the material might seem at first dry and statistical; but as you learn more, you will begin to understand and interpret information about individual pixels, their communities, and their families.

## 7.1 Pixel Measurements

An image is an organized collection of values. Each value that makes up a monochrome image is characterized by three quantities: the two numbers that specify the pixel's position in the image, and the numerical value of the pixel. The first two values are *pixel coordinates*, and the third is the *pixel value*. Color images are characterized by five or more quantities: two *pixel coordinates*, and three or more *color pixel values*. The color channels may represent red, green, and blue *color planes*, or they may represent color in a *luminance/chrominance color space*, as luminance, color hue, and color saturation.



**Figure 7.1** A digital image is an organized array of pixels. Each pixel occupies a specific column and line. An image that is  $M$  pixels wide by  $N$  pixels deep contains a total of  $M \times N$  pixels. In computer applications, the upper left pixel occupies the  $(0, 0)$  location, and the lower right occupies  $(x_{\max}, y_{\max})$ .

### 7.1.1 Pixel Coordinates

These specify the location of a pixel in an image. Pixel coordinates in raw images, calibration frames, and calibrated but otherwise unprocessed images refer to a corresponding location—a specific photosite—on the detector.

The coordinate assigned to the first sample in the first line read from a CCD is  $(0, 0)$ , the second sample is  $(1, 0)$ , then  $(2, 0)$ , and so on to pixel  $(x_{\max}, 0)$ . On the CCD, the array is shifted down one line, and the first sample in the second line is  $(0, 1)$ , the second sample of the second line is  $(1, 1)$ , and so on. The last pixel in the last line read from the CCD is  $(x_{\max}, y_{\max})$ . The number of samples per line,  $M$ , is  $x_{\max} + 1$ , the number of lines in an image,  $N$ , is  $y_{\max} + 1$ , and the total number of pixels in the image is  $MN$ , or  $(x_{\max} + 1)(y_{\max} + 1)$ .

In the literature of CCD imaging, the nomenclature for labeling the coordinate axes varies considerably. When referring to the entire image, the horizontal, column, or sample, axis is most often called the  $x$ -axis. The vertical, row, or line, axis is most often called the  $y$ -axis. When referring to a small portion of an image, such as the region immediately surrounding pixel  $(x, y)$ , offsets in the  $x$ -axis are designated  $i$ , and offsets in the  $y$ -axis are designated  $j$ . Thus, pixels in the neighborhood surrounding pixel  $(x, y)$  will be named  $(x \pm i, y \pm j)$ , where the variables  $i$  and  $j$  run through some range of values.

There are two conventions for displaying images: the Cartesian convention in which the  $y$ -axis increases upward, and the computer convention in which the  $y$ -axis increases downward. In both cases, the  $x$ -axis increases from left to right. Although books on image processing sometimes use the Cartesian display convention, virtually all software for small computers uses the computer display convention.

Although pixels have integer coordinates, image processing routines often must interpolate to find values “between” pixels. Because the numerical value of each pixel contains the integrated light of the whole photosite, the center of the pixel is the location  $(x, y)$ , and the four “corners” of the pixel are  $(x-0.5, y-0.5)$ ,  $(x+0.5, y-0.5)$ ,  $(x-0.5, y+0.5)$ , and  $(x+0.5, y+0.5)$ .

- **Tip:** ***AIP4Win** provides tools for measuring pixel positions: the Image Display Control and the Magnifying Glass Tool. The Image Display Control gives the pixel coordinates and pixel value under the cursor in the current image whenever the left mouse button is down. The Magnifying Glass Tool shows an enlarged region surrounding the cursor position, with a readout of the pixel coordinates and pixel value.*

### 7.1.2 Pixel Value

Pixel value is the numerical value of a pixel, and it can be an integer or floating-point number. CCD cameras produce output in which the pixels have integer values; after processing, CCD images may contain integer or floating-point values. Pixel values can express almost any unit of measurement. They can represent the raw digital output of a CCD in analog-to-digital units (ADUs), but can also represent photoelectrons, radiometric units, units of energy ( $\text{ergs cm}^{-2} \text{ sec}^{-1}$ ), or magnitudes per square arcsecond.

The unit of measurement found in raw CCD images, calibration frames, and calibrated but otherwise unprocessed images is the analog-to-digital unit, or ADU. ADUs are directly related to the statistically useful number of photoelectrons through the conversion factor,  $g$ . After several processing steps, this linear relationship may have become distorted, so it is usually desirable to regard pixel values in processed images as dimensionless numbers.

Although images from CCD cameras begin with integer pixel values, calibration and image processing soon convert these to non-integer values. If the noise level in the image is greater than one ADU, it is possible to work with integer pixel values with minimal loss of information. However, treating pixel values as floating-point numbers reduces loss to negligible levels and gives the user access to a wide dynamic range.

- **Tip:** ***AIP4Win** treats all pixel values as floating-point values. The maximum value that **AIP4Win** allows is  $1 \times 10^{32}$ , and the minimum is  $-1 \times 10^{32}$ . The smallest allowed pixel value is  $\pm 1 \times 10^{-32}$ ; smaller values are treated as zero or “not a number.” Over this wide*

## Chapter 7: Image Analysis

*dynamic range, numerical values are carried with a precision of approximately one part in 8,400,000.*

### 7.2 Whole-Image Analysis

There are three basic reasons for looking at the statistical properties of a digital image: (1) to determine whether the image is technically sound; (2) to establish the range of values in the image, in order to determine the best way to display it; and (3) to determine the optimum brightness scaling to use in processing and displaying a final image.

#### 7.2.1 Image Statistics

Image statistics are usually computed by pixel-by-pixel analysis of the entire image. The procedures pays no attention to the subject of, or features in, the image—all they “see” are pixel values. However, the inventory of pixel values resulting from performing the statistical procedures can ferret out a great deal of information that is invisible in a simple inspection of the image.

- **Tip:** *AIP4Win’s Statistics Tool computes and displays the statistical properties of the whole image. This provides information that you can use to evaluate the quality and integrity of your images.*

##### 7.2.1.1 Minimum Pixel Value

This value defines the bottom of the range of pixel values found in an image. In a raw image, the minimum should be the pixel value of the bias plus dark current plus the pixel value of the background sky—a positive number. In a calibrated image, bias and dark have been removed, so the minimum value should be the pixel value of the background sky—a positive number. A minimum pixel value of zero probably means one of three things:

1. lines or columns are masked and receive no light, or
2. pixel values in the sky are very low, and a few are zero, or
3. calibration was performed incorrectly, truncating negative values.

Masked lines and columns are common on many CCDs, so a zero value after calibration is normal. If the sky background is low, the normal variation due to noise occasionally produces a zero or negative value, which is normal. Incorrect calibration might involve a dark frame taken at a different integration or temperature than the image, generating negative values in dark subtraction. In this case, you should find the cause.

##### 7.2.1.2 Maximum Pixel Value

The maximum is the top of the range of pixel values found in an image. This can serve as a useful diagnostic, especially if it is close to either the minimum or maximum possible pixel value.

## Section 7.2: Whole-Image Analysis

If the range (i.e., the maximum minus the minimum pixel value) is small compared to the maximum possible pixel value (255 for 8-bit integers, 4095 for 12-bit integers, and 65,535 for 16-bit integers), then the image has either received very little exposure; or it shows a very bland subject, such as a flat-field image.

If the maximum is close to the maximum possible value, then some of the pixels represent non-linear or fully saturated photosites on the CCD. A high maximum pixel value warrants investigation, especially if the image will be measured for astrometric or photometric information.

### 7.2.1.3 Mean Pixel Value

An image that is  $M$  pixels wide by  $N$  pixels deep contains a total of  $MN$  pixels. The mean pixel value is the sum of the pixel values divided by  $MN$ :

$$\overline{P_{\text{image}}} = \frac{1}{MN} \sum_{i=1}^{MN} P_i \quad (\text{Equ. 7.1})$$

where  $\overline{P_{\text{image}}}$  is the arithmetic mean of the pixel values in the image. The mean pixel value is usually somewhat greater than the sky pixel value because every object in the image that is brighter than the sky value raises the mean value.

### 7.2.1.4 Median Pixel Value

The median is that pixel value for which  $MN/2$  pixels in the image have greater values, and  $MN/2$  pixels have lower values. In deep-sky images, the median pixel value is usually quite close to that of the sky. The median is useful because it is largely independent of low values that occur along the edges and extreme high values due to stars and other bright objects in the image.

### 7.2.1.5 Standard Deviation

The standard deviation is a measure of the distribution of pixel values from the mean pixel value of the image; that is, how greatly the collection of pixels in the image stray from the mean pixel value. Given an image with dimensions  $M$  by  $N$  pixels and a mean pixel value of  $\overline{P_{\text{image}}}$ , the standard deviation,  $\sigma$ , is:

$$\sigma = \sqrt{\frac{1}{MN-1} \sum_{i=1}^{MN} (P_i - \overline{P_{\text{image}}})^2} \quad (\text{Equ. 7.2})$$

where  $P_i$  is the  $i$ th pixel in the collection of  $MN$  pixels.

Standard deviation is a more useful measure of the distribution of pixel values found in an image than is the range between the minimum and maximum, because it is not influenced by a small population of unusually low or high values, but by the properties of all pixels in the image.

## Chapter 7: Image Analysis

### 7.2.1.6 Low-Point and High-Point Pixel Values

The pixel values in an image are distributed between the minimum and maximum values. The minimum value is usually considerably less than that of the bulk of the pixels, and the maximum is often much greater than the bulk of the pixel values. If the pixel values are simply mapped into the range of grays available on a computer monitor, the image will appear dark and muddy. The low point and high point provide considerably more useful information.

The percent-point pixel value is the value that lies above some fraction of pixels in an image. The minimum and maximum are the extreme pixel percentiles—the minimum is the 0.00-point pixel value because none lie below the minimum. Likewise, the maximum is the 1.00-point pixel value because it is greater than that of all the pixels in the image. The 0.50-point pixel value has a value greater than half of the pixels in the image.

Percent-point pixel values are extremely valuable in displaying images because they allow the user to determine how many pixels will saturate to pure black or pure white on the display. If the black level (low point) in the display is set to 0.01, then 1% of the pixels in the image will be black. If the white level (high point) is set to 0.99, then  $100\%-99\% = 1\%$  of the pixels will saturate white.

For example, an image with a minimum pixel value of 0 and a maximum pixel value of 3545 may have a 0.01 low-point pixel value of 356 and a 0.99 high-point pixel value of 510. Using these pixel values to display the image means that 2% of the pixels will be saturated either black or white (probably the dark corners of the image and the cores of the brightest stars), while the remaining 98% of the pixels will display in shades of gray.

- **Tip:** *AIP4Win uses percent-point values to make the automatic brightness settings in the Image Display Control, and to select low- and high-point values for the Brightness Scaling Tool.*

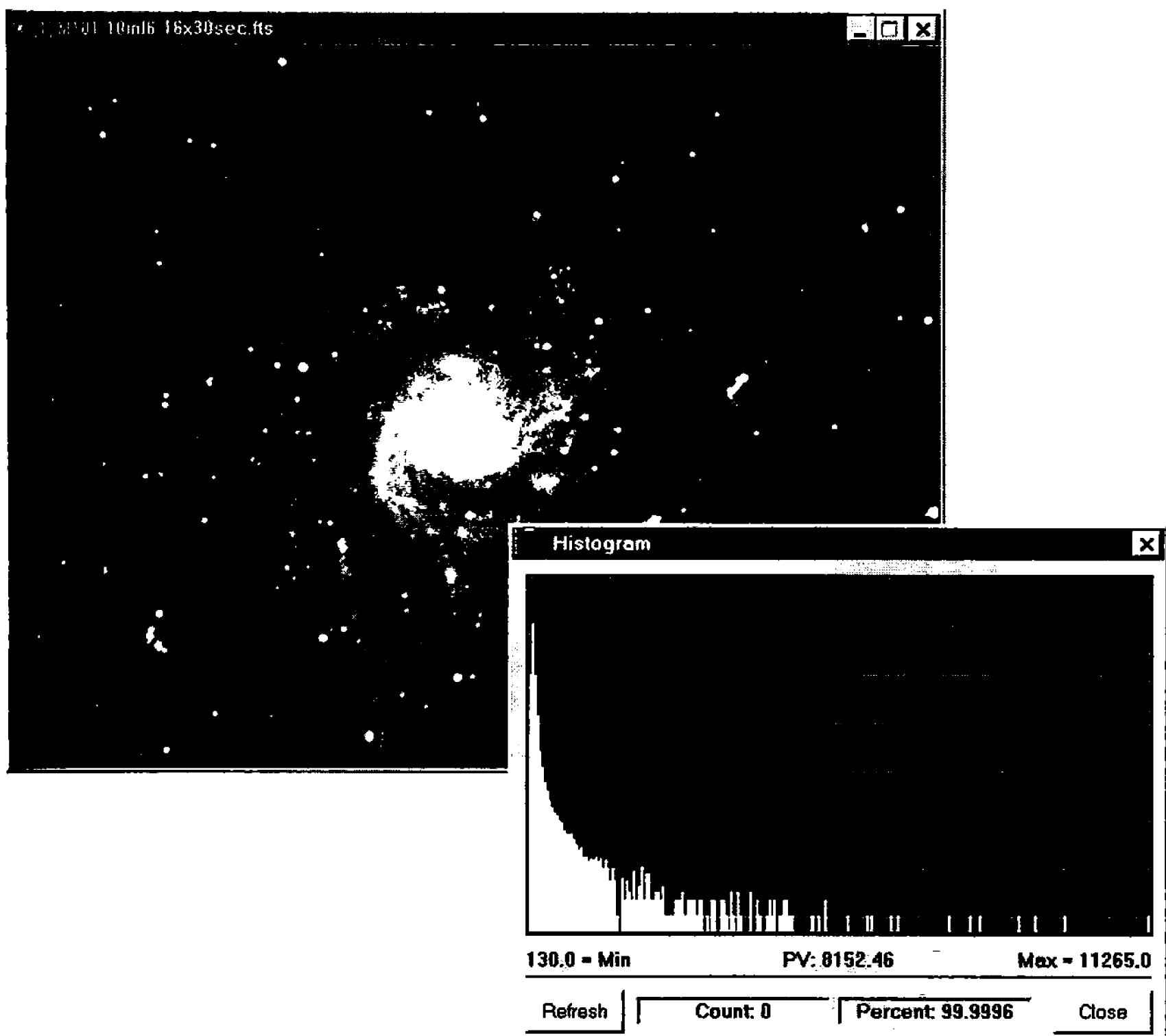
## 7.3 The Image Histogram

A histogram is an inventory of the pixel values found in an image. It is a graph that plots the number of pixels of each pixel value on the  $y$  axis *versus* pixel value on the  $x$  axis. Although a histogram contains no explicit information about the image contents, the distribution of pixel values implies a great deal about the image.

Because of the enormous range of brightness captured in astronomical images, the histogram is usually plotted as the logarithm of the number of pixels versus pixel value. In a linear plot, the number of pixels is plotted on a scale that might run 0, 20,000, 40,000, 60,000, 80,000, and 100,000—in equal steps of 20,000. In a logarithmic plot, the scale units are 1, 10, 100, 1,000, 10,000, and 100,000. In a linear histogram, a feature consisting of 100 high-value pixels might not even be visible; whereas in a logarithmic histogram, an isolated 100-pixel feature appears as a respectable bump.

As a quick image diagnostic, you can't beat the histogram. Without even seeing the image, you can tell at a glance what type it is. The bias frame of a healthy

## Section 7.3: The Image Histogram



**Figure 7.2** The histogram of an image is a plot of pixel value versus the number of pixels having each value. It provides insight as to what range of pixel values holds the most useful information. Although pixel values here range from 130 to 11265, values are strongly concentrated near that of the background sky.

CCD camera, for example, has a characteristic spike at a low pixel value and virtually no pixels outside the spike. In contrast, a dark frame shows a narrow spike but with an exponentially decaying tail of high-value pixels. The spike comes from the vast majority population of well-behaved normal pixels, while the tail reflects the presence of a minority population of hot pixels.

Healthy flat-field frames usually show a broad peak centered on the average value of the flat field—but a sharp cutoff on the high side reveals saturation, a pathological condition for a flat-field frame.

With astronomical images, of course, you see a broad spike from sky pixels, and you can tell whether the values cluster at the low end (underexposure) or fill a reasonable portion of the dynamic range of the CCD. Pixels that are parts of star images form a tail that declines exponentially toward high values. Humps and bumps superimposed on the exponential tail are generally from the pixels in galaxies and nebulae. A sharp cutoff at high pixel values marks the onset of saturation in the CCD chip, usually affecting only the cores of bright star images.

Planetary images often display bimodal distributions with one peak from the sky surrounding the planet, and a second peak at higher pixel values from its disk.

## Chapter 7: Image Analysis

Mars often shows a spike at the high end of its peak from the high-value pixels in the polar cap.

### 7.4 Image Feature Analysis

The human eye and brain perceive “objects” and “features” in images—stars, nebulae, and galaxies—but the computer sees nothing but numbers. Nonetheless, the computer is a valuable aide and ally in the analysis of those clusters of pixels that make up what we humans see. To enlist the computer’s help, however, we must define an area—a region of interest—that is large enough to contain the feature or object, yet small enough that the statistical differences that it causes make the region distinctively different.

#### 7.4.1 Pixel Statistics

The pixel statistics of small regions of interest are the raw material of feature analysis, from the analysis of the noise in the CCD to the measurement of stellar magnitudes. The first step in analyzing a group of pixels is to define the location of the region of interest; the second is to perform the mathematical procedures; and the third is to interpret the statistical results.

Of these steps, interpretation is the most difficult. You can use pixel statistics

- to explore the performance of the CCD itself,
- to derive positions of image features or their brightness, or
- to color-balance a set of color separation images.

The following section, however, deals only with the first two: fundamental pixel-level selection of a region, and the computation of its statistical properties.

##### 7.4.1.1 Defining a Region of Interest

“Region of interest” is a fancy term for a collection of pixels with properties that you want to measure. Membership in the collection must be determined by some kind of rule, such as nearness to a feature that you want to know more about. Regions of interest are usually defined as including pixels in a circle or square that includes the feature, or a ring that surrounds the feature of interest. Once the pixels in the region of interest have been selected, the program can determine useful things like the average value of the pixels that comprise it.

By definition, a circular region of interest includes all pixels at a distance less than or equal to the specified radius,  $R$ , from a central point,  $(x_0, y_0)$ . Any pixel  $(x, y)$  that satisfies the condition

$$R \geq \sqrt{(x - x_0)^2 + (y - y_0)^2} \quad (\text{Equ. 7.3})$$

belongs to the region of interest; any that lies outside it is ignored.

When the radius is small, this definition generates blocky circles. A circular region of interest with a radius of 1 contains 5 pixels; with a radius of 2, it contains

13; and with a radius of 3, 29 pixels—assuming that  $(x_0, y_0)$  are integer values. As the radius becomes larger, the region better approximates a circle.

An annular, or ring-shaped region of interest, includes all pixels that satisfy a double criterion of lying inside an outer radius  $R$  and outside an inner radius  $r$ :

$$R \geq \sqrt{(x - x_0)^2 + (y - y_0)^2} > r . \quad (\text{Eqn. 7.4})$$

When the radii are small or close in value, the ring they define is quite blocky; but as the radii grow, the borders of annular regions become fairly smooth. Such regions are especially valuable for estimating the statistical properties of the background sky pixels near an interesting object or feature, by sampling those that surround it.

A square region of interest includes all pixels that lie within a specified distance (called the “radius” even for non-circular regions) in the  $x$  and  $y$  axes:

$$\begin{aligned} R &\geq (x - x_0) \\ R &\geq (y - y_0) . \end{aligned} \quad (\text{Eqn. 7.5})$$

A square region of interest with a radius of  $R$  pixels is a box  $2R + 1$  pixels on a side containing  $4R^2 + 4R + 1$  pixels. A region of interest with a radius of 1 pixel is 3 by 3 pixels and contains 9 total; a region with a radius of 10 pixels is 21 on a side and contains a total of 441 pixels.

Like the annulus, a hollow square region of interest contains pixels inside an outer radius,  $R$ , and outside an inner radius,  $r$ :

$$\begin{aligned} R &\geq (x - x_0) > r \\ R &\geq (y - y_0) > r . \end{aligned} \quad (\text{Eqn. 7.6})$$

Because pixel space is quantized, for square regions of interest,  $R$  and  $r$  can only hold integer values.

- **Tip:** *AIP4Win includes a Pixel Tool for measuring the statistical properties of square, circular, and annular regions of interest. You can select the shape, and set both inside and outside radii. When the inside radius is zero, all pixels inside the outer radius are included in the statistics.*

### 7.4.1.2 Minimum Pixel Value

The minimum pixel value serves as a diagnostic for abnormal noise or unusual conditions that might be present in a region of interest, interpreted within the context of the other pixel statistics. For example, if it is far below a value that you might reasonably expect, the situation warrants checking.

Consider this example: you select a region containing a few moderately bright stars. The median pixel value is 1000, which probably represents the pixel value of the background sky, and the maximum pixel value is 3144, presumably the brightest pixel in one of the star images—but the minimum pixel value is zero!

## Chapter 7: Image Analysis

A reasonable minimum would be somewhat less than the median, such as 850. The zero suggests there is a problem. Check for rare events—such as a cosmic ray in the dark frame—that might have caused the abnormally low minimum value.

### 7.4.1.3 Maximum Pixel Value

This parameter is another useful diagnostic. It can tell you the highest pixel value found in a star image, check for pixels with abnormally high thermal noise (i.e., “hot” ones), and verify that all of the pixels in a star image are within the linear range of the CCD.

The range of the maximum and minimum pixel values is an indicator of the dispersion of pixel value from the norm—but it’s a poor indicator because it takes only one abnormally low or high pixel value to substantially expand the range.

### 7.4.1.4 Mean Pixel Value

This measure is the sum of the pixel values divided by the number of pixels in the region of interest:

$$\bar{P} = \frac{1}{n} \sum_{i=1}^n P_i , \quad (\text{Equ. 7.7})$$

where  $\bar{P}$  is the arithmetic mean of  $n$  pixels in the given region.

Because the mean value includes *every* pixel in the region of interest and can therefore be skewed by a small number of abnormally low- or high-value pixels, the mean is a risky way to determine the pixel value of a sky background.

### 7.4.1.5 Variance

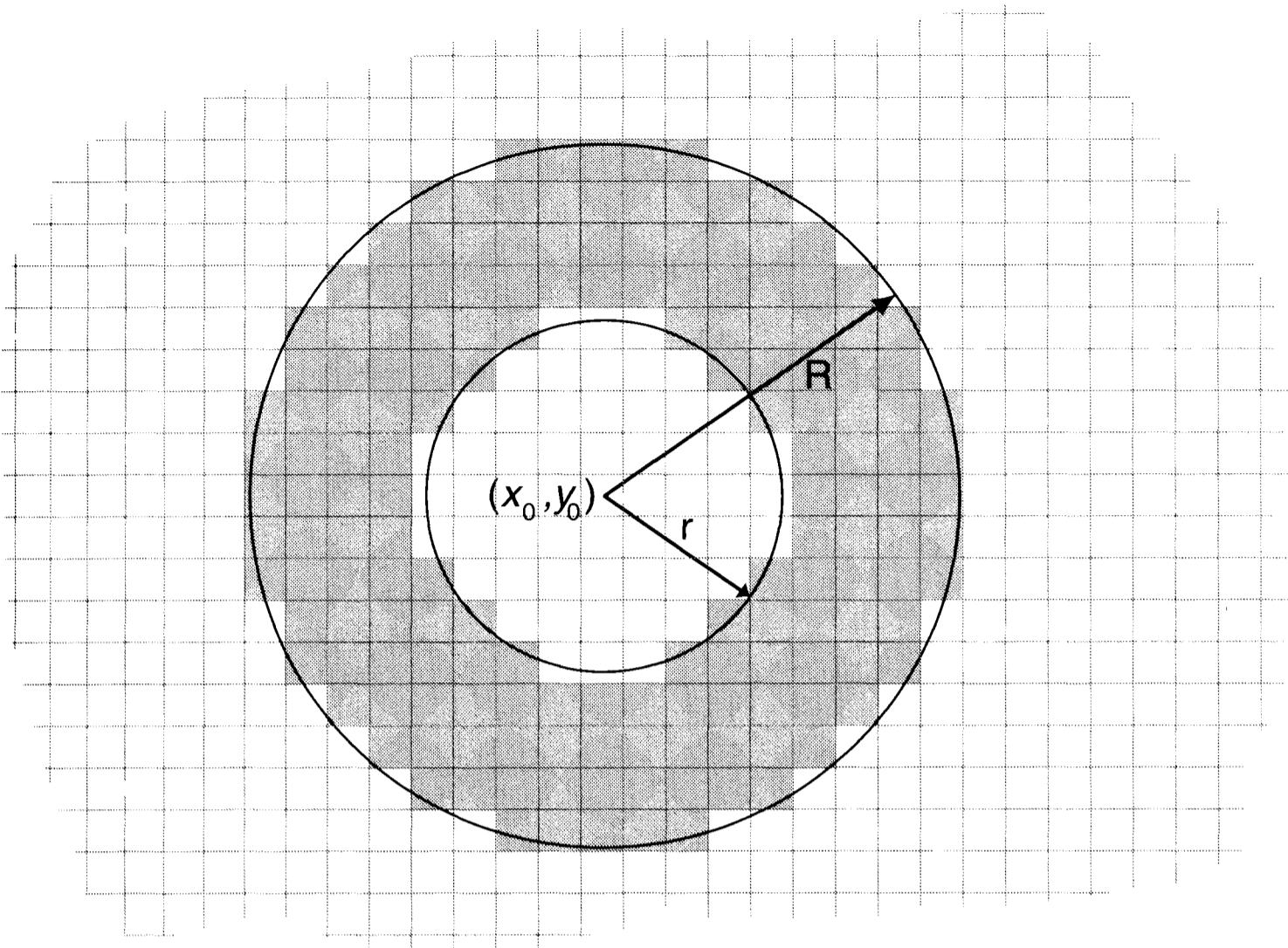
The variance,  $\sigma^2$ , is a measure of the dispersion of a set of measurements from the mean value of the measurements:

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (P_i - \bar{P})^2 , \quad (\text{Equ. 7.8})$$

where  $\bar{P}$  is the mean value and  $P_i$  is the  $i$ th pixel in a region of interest containing  $n$  pixels. The square root of the variance is the standard deviation. The most immediate application for the variance in CCD imaging is to characterize the CCD’s readout noise and conversion factor (i.e., electrons per ADU) from test images.

### 7.4.1.6 Standard Deviation

The standard deviation,  $\sigma$ , is the square root of the variance. Given a collection of pixel values,  $P_1, P_2, \dots, P_n$ , in a region of interest, the standard deviation is computed from:



**Figure 7.3** Regions of interest can be square, rectangular, circular, elliptical, or as shown above, annular. Pixels in the region of interest lie at or inside the outer radius,  $R$ , but outside the inner radius,  $r$ . The annular region of interest is useful for determining the sky brightness in CCD stellar photometry.

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (P_i - \bar{P})^2}, \quad (\text{Equ. 7.9})$$

where  $\bar{P}$  is the mean value and  $P_i$  is the  $i$ th pixel in a region of interest containing  $n$  pixels.

In a set of random numbers, 68% of them should lie between  $\bar{P} - \sigma$  and  $\bar{P} + \sigma$ . The standard deviation is the characteristic width of a Poisson or Gaussian distribution; thus in a random sample, the standard deviation is a direct measure of what we loosely call “noise.”

#### 7.4.1.7 Signal-to-Noise Ratio

To estimate the signal-to-noise ratio (SNR) for a small, uniform part of an image, measure the mean and the standard deviation for that region. If the mean pixel value is 600 and the measured standard deviation is 20, the SNR is  $600/20 = 30$ . Because the SNR depends on the amount of light striking a CCD, you cannot define a meaningful SNR for an image, but only for a specific pixel value, such as the brightness of the sky background, found in the image.

Consider an example in which the sky background is 600 ADU and the SNR is 30. The sky brightness varies randomly with a  $\sigma$  of 20 ADU. In this image, it

## Chapter 7: Image Analysis

would be difficult to detect a star less than 20 ADU brighter than the mean sky value.

The strength of a signal such as a star image is sometimes rated in terms of how many “sigmas” it departs from the background. Continuing the example above, if a weak star image has a pixel value of 660 ADU—60 more than the mean sky—then it is said to be a “three-sigma” object. The greater the departure from the mean, the higher the probability that the object is real—and not just a random cluster of high pixels. The more pixels the star image contains, the more certain the detection becomes; because the probability that multiple three-sigma events would occur side by side is very low.

### 7.4.1.8 Median Pixel Value

The median is the middle value in a sorted set of values. In a region of interest, an equal number of pixels has values greater than the median and less than the median. In the sequence: 5, 5, 5, 5, 7, 8, 11, 13, 99, the median value is 7—as there are four lower values and four higher values. The mean value is 17.5. The median rejects extreme values that the mean does include. In astronomical images, the median is a powerful tool for estimating the value of the sky background without including light from faint stars that raise the mean value.

### 7.4.1.9 Mean of Median Half

The “mean of the median half” is a hybrid statistic designed to use information from as many pixels in the region of interest as possible while still excluding the most extreme values. The mean of the median half is the mean of the middle half of a set of sorted pixel values. In the sequence: 5, 5, 5, 5, 7, 8, 11, 13, 99, the mean of the median half is 7.2. The bottom quartile and top quartile were rejected in forming this value.

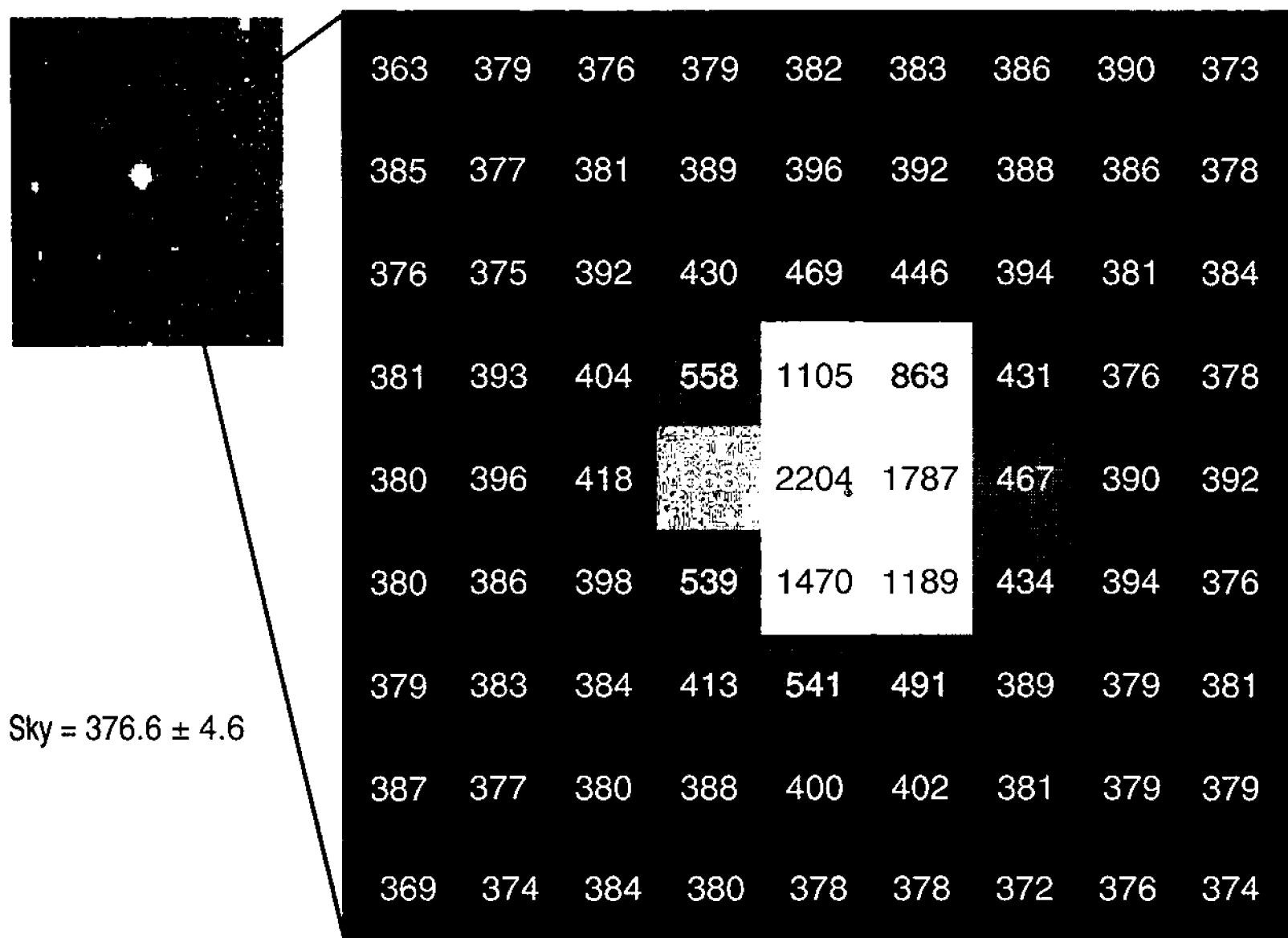
The mean of the median half is useful in astronomy for measuring the sky background pixel value that rejects faint background stars, and at the same time generates a background sky value that avoids being pigeon-holed to the nearest integer value of a pixel in the region of interest.

## 7.5 Determining a Centroid

The centroid of a physical object is its barycenter, or center of gravity. It is the point at which the object—suspended from a thread or teetering on a fulcrum—balances. In an image, the centroid is the point at which the values of all surrounding pixels are “balanced.” What makes the centroid interesting is that even though light from a star is spread over many pixels, the distribution of light among them enables us to recover the exact center of the star image *within a small fraction of a pixel*. The process is called determining a centroid.

The value of a star’s centroid lies not in knowing exactly where on the CCD the light from a particular star happened to fall—which depends on factors that we cannot reproduce—but of determining the *relative* positions of the stars in the im-

## Section 7.5: Determining a Centroid



**Figure 7.4** Blocky star images contain precise information about the location of the star that formed them. In this star image, pixel values record how much light fell on each photosite. The dot under the “4” in 2204 shows the location of the computed centroid, and its diameter shows the probable error in its position.

age, since the star images were formed at the same time. Even though every star image has been enlarged by diffraction, spread by atmospheric turbulence, and smeared by guiding errors, a precise record of their relative positions is stored in the image.

On a CCD image, a star appears as a cluster of pixels having values greater than those of the surrounding sky. Images of bright stars are unmistakable, but that of a faint one may be nearly indistinguishable from random noise in the background sky. The human eye is remarkably good at picking out clusters of pixels, but extracting the centroid of a star image from a CCD image with a computer requires a surprisingly large amount of analysis and number crunching. To accomplish what the eye does with such apparent ease, the computer must:

1. determine which pixels belong to a region of interest centered on the approximate location of the star image;
2. determine the mean brightness within the region of interest;
3. determine (assuming that the pixels that make up the star image are on average brighter than pixels of the surrounding sky) which pixels might belong to the star image;
4. determine which candidate star pixels belong in a cluster, presumably a star image; and

## Chapter 7: Image Analysis

5. compute the centroid of the pixels that belongs to the star image.

Computing a centroid begins when an astronomer selects a star for centroiding. Since the cluster of pixels that make up a star image is roughly circular, the region of interest includes all pixels within a defined radius, typically two or three times larger than the visible disk of a bright star. Given an initial center at  $(x_0, y_0)$ , the region of interest includes all pixels  $x_i, y_i$  that satisfy the condition:

$$r \leq \sqrt{(x_0 - x_i)^2 + \left(\frac{(y_0 - y_i)}{\text{aspect ratio}}\right)^2} \quad (\text{Equ. 7.10})$$

where  $r$  is the radius of the region of interest in pixels, and aspect ratio is the pixel width divided by the pixel height.

The centroid procedure next determines the mean pixel value of the pixels within the region of interest. Assuming this region actually contains a star image, pixels belonging to it should be brighter than the mean pixel value. If necessary, the program applies a “fudge factor” to the mean pixel value to get a threshold value that distinguishes star pixels from sky pixels.

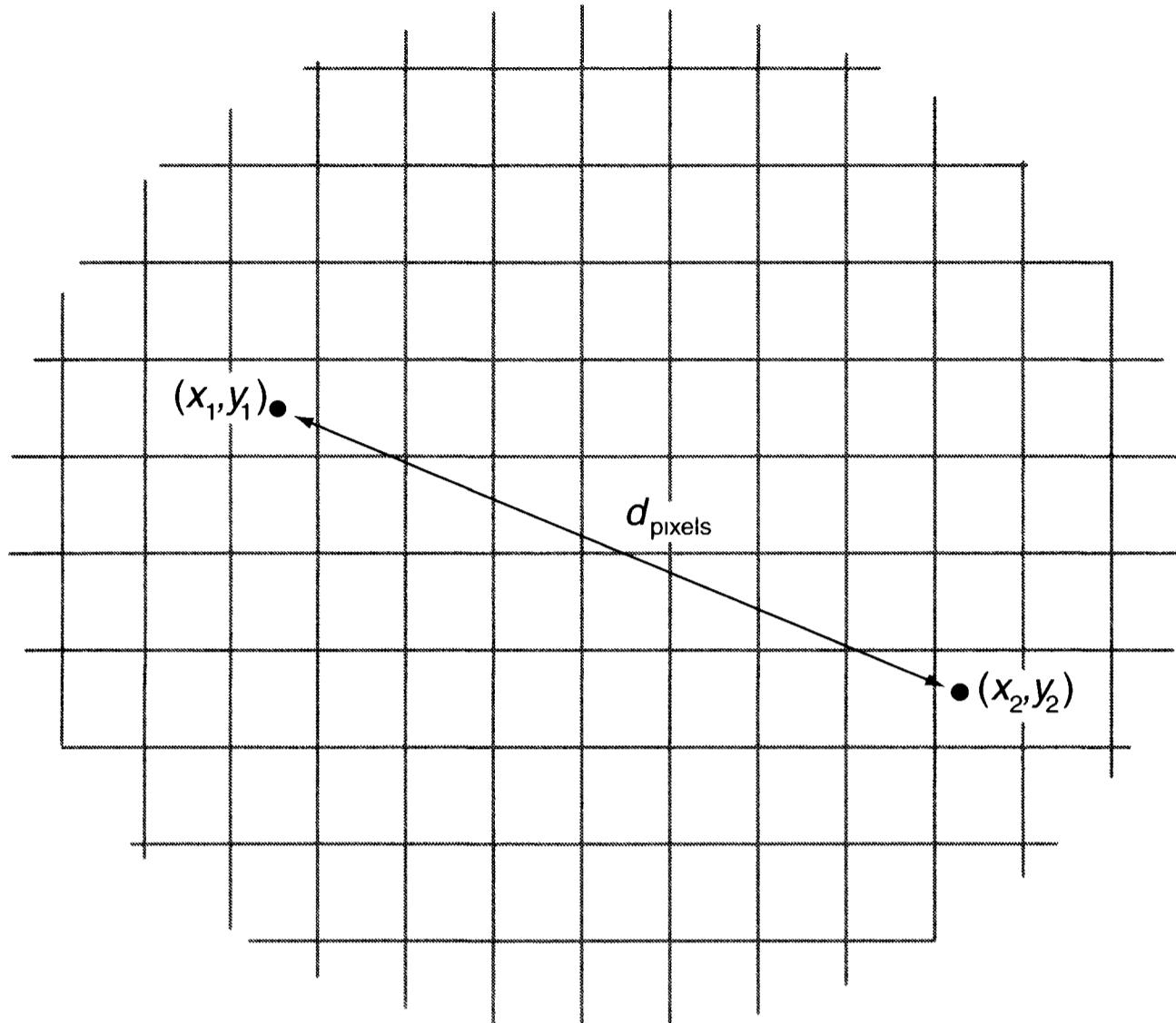
The procedure now identifies pixels belonging to the star image using a double test: the pixel value must be greater than the threshold, and some number of neighboring pixels must also be star pixels. The neighborhood requirement insures that the procedure rejects individual sky pixels that exceed the threshold because of random noise. (Requiring three neighbors does a good job of selecting between star pixels and random noise.) At the same time, the procedure determines the mean pixel value of non-star pixels (i.e., those that fall below the threshold value) to use as the sky background brightness.

Finally, the procedure computes the centroid using the moment equation for the  $x$ -axis and  $y$ -axis centroids:

$$\bar{x} = \frac{\sum x_i(P_i - S)}{\sum (P_i - S)} \quad \text{and} \quad \bar{y} = \frac{\sum y_i(P_i - S)}{\sum (P_i - S)} \quad (\text{Equ. 7.11})$$

where  $P_i$  is the pixel value of the pixel at  $(x_i, y_i)$ , and  $S$  is the mean background sky brightness. The moment equation weights each pixel along the two axes by the amount of starlight that has fallen on that pixel. The result is  $(\bar{x}, \bar{y})$ —the centroid of the star image.

- **Tip:** *The Star Image Tool gives you direct access to stellar centroids for specialized projects. The Distance Tool, the Astrometry Tool, the Photometry Tools, and the image registration and stacking tools all rely on accurate stellar centroids. The centroid procedure also works remarkably well on galactic centers, comet nuclei, and with a sufficiently large radius setting, on irregularly shaped objects.*



**Figure 7.5** Knowing their pixel coordinates, you can find the distance between two points in an image. The points can be pixel coordinates, or coordinates found by computing the centroids of two star images. The distance between two star images can be determined to sub-pixel accuracy.

## 7.6 Distance on a CCD Image

The distance between two points on a CCD image is the very simplest form of astrometry, the science of measuring the positions of heavenly bodies. The points can be integer pixel positions or the non-integer centroids of star images.

The distance  $d$  between two points on the surface of a CCD is:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (\text{Equ. 7.12})$$

where  $(x_1, y_1)$  and  $(x_2, y_2)$  represent the two locations. The  $x$ -axis is conventionally taken as the sample direction, and the  $y$ -axis as the line direction on the CCD.

On a sensor with square pixels (i.e., equal pixel width and pixel height), the distance is in pixels. On one with differing pixel width and height, the raw pixel positions must be corrected for the pixel aspect ratio of the chip. The pixel aspect ratio is defined as the pixel width divided by the pixel height, so the distance is:

$$d = \sqrt{(x_1 - x_2)^2 + \left( \frac{(y_1 - y_2)}{\text{aspect ratio}} \right)^2}, \quad (\text{Equ. 7.13})$$

with the result in units of pixel width.

The angular separation,  $\vartheta$ , in radians, between the two measured points in

## Chapter 7: Image Analysis

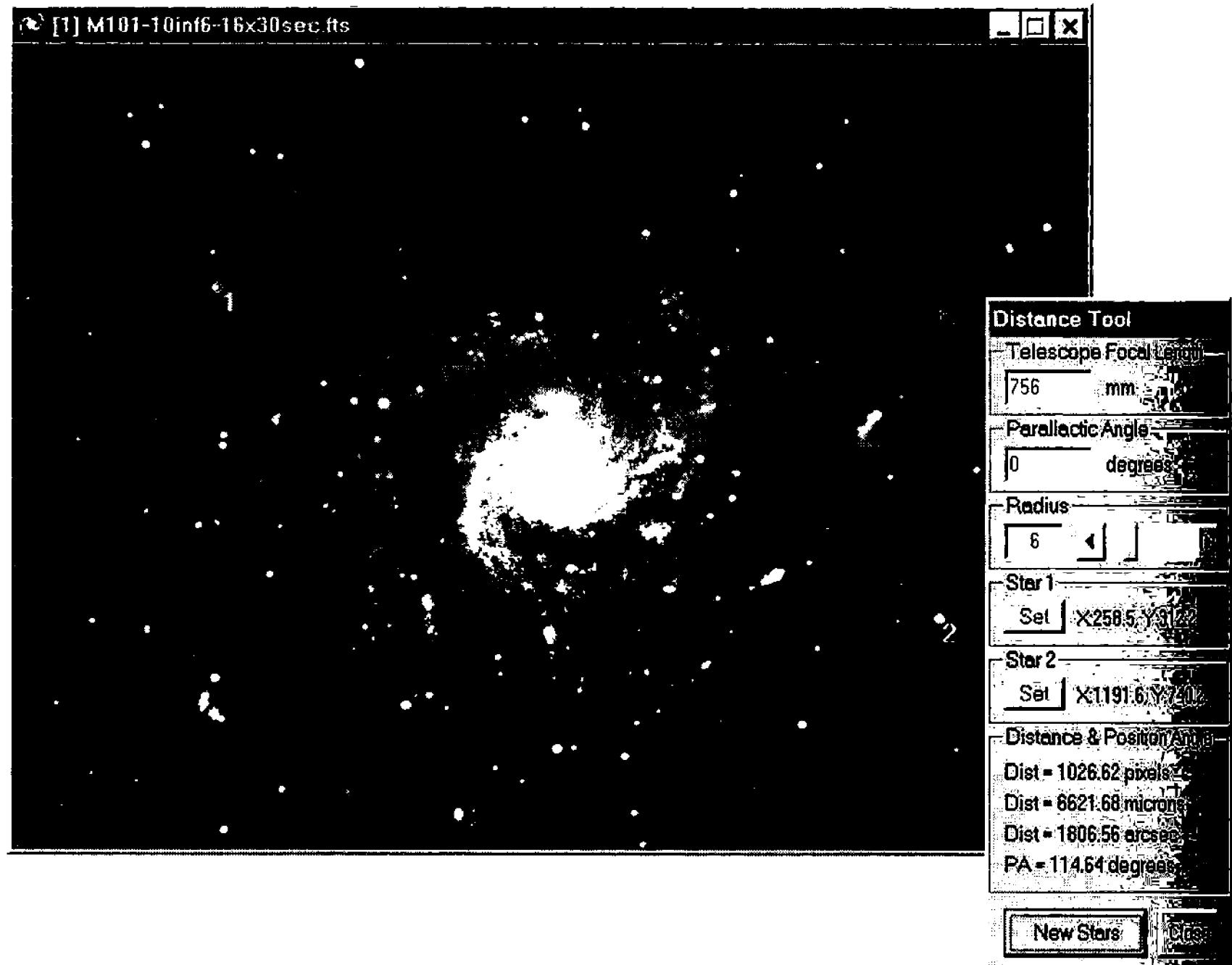


Figure 7.6 The Distance Tool in **AIP4Win** computes the distance between two star images. If you know the focal length of the telescope and the orientation of the CCD image, you can measure the angular separation and the position angle of the second star with respect to the first.

the focal plane of a telescope is:

$$\vartheta = \arctan(dd_{\text{pixel}}/F) \text{ [radians]}, \quad (\text{Eqn. 7.14})$$

where  $F$  is the focal length of the telescope objective,  $d$  is the separation in pixels, and  $d_{\text{pixel}}$  is the pixel width in the same units as the focal length.

## 7.7 Image Profiles

Profiles are one-dimensional slices through images, a graph of pixel values across a single line or column of pixels. Profiles are a huge help in sorting out and understanding brightness relationships in images. Although the human eye is good at extracting features, it is quite poor at estimating brightness. Humans perceive spiral galaxies, for example, as dominated by the spiral arms—but a profile through a typical spiral galaxy reveals that the nucleus overwhelms the arms, which are mere bumps superimposed on the radial decline in brightness from the nucleus. Profiling planetary images also reveals the importance of limb darkening, which observers accustomed to eyeballing images hardly notice.

- **Tip:** The *Profile Tool* in **AIP4Win** extracts a profile of pixel values along an arbitrary line drawn through an image.

## 7.8 Astrometry

Astrometry is the science of measuring the positions of celestial objects. In CCD images, the first step is to determine the centroids of a set of stars (called *reference stars*) with known spherical coordinates, and also the centroids of the images of the objects whose position you wish to measure (that is, the *target stars*). If possible, the array of reference stars should surround the target star or stars. The theory and practice of astrometry are covered in full in Chapter 9.

## 7.9 Photometry

Astronomers refer to the determination of a celestial object's brightness as photometry. Even though a star image may sprawl over many pixels, because CCDs are linear, they accurately record the total amount of light in each pixel, and therefore, also record the total light in any feature. Star images present something of a problem, however, because the starlight is mixed with light from the background sky. Extracting an accurate measurement of the total light in a star image is tricky, but it is one of the most important “instruments” in the digital astronomer’s tool kit. Chapter 10 covers photometry in detail.

### 7.9.1 Defining the Star Image

Photometry involves calculating the statistics of small regions of interest; specifically, the region encompassing a star image and the dark-sky region immediately surrounding it. However, the two regions of interest must be defined in terms of their dimensions in the focal plane of the telescope rather than the pixel space, somewhat complicating the analysis for CCDs with pixel widths and heights that are not the same.

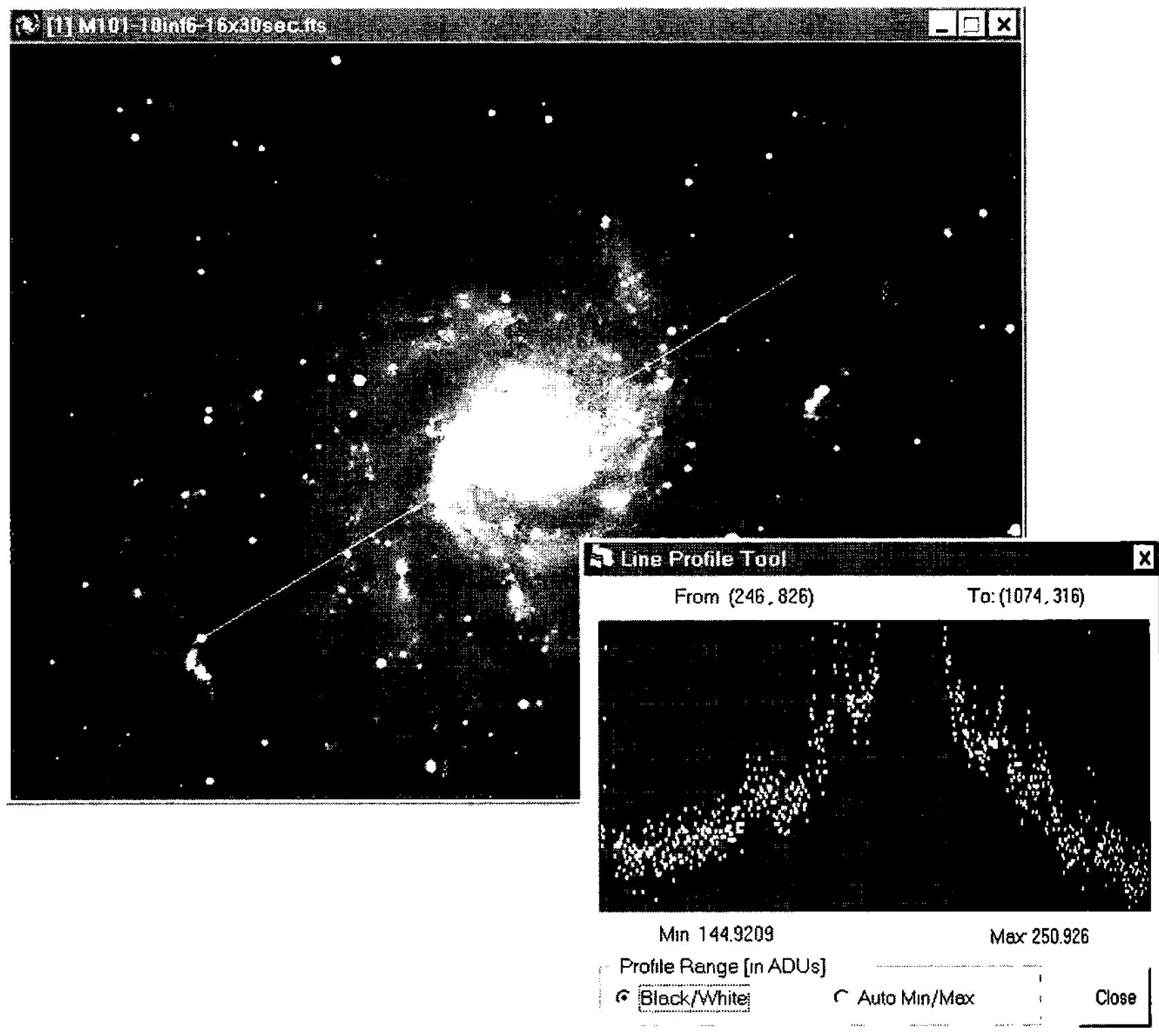
Star images are comprised of three regions:

1. a core consisting of the diffraction disk and the inner diffraction rings, containing perhaps 90% of the star’s total light;
2. an aureole several times larger than the core containing light that, to the eye, is either hidden by the dazzle of the core or is too faint to see, containing perhaps 90% of the remaining light, and;
3. the surrounding sky, containing not only sky light, but also a tiny fraction of light from the star itself, parasitic light such as field flooding, and possibly a few faint background stars.

Both the core and the aureole also contain background sky light, and they also quite possibly contain unwanted background stars.

The basic task of stellar photometry is to determine the total amount of light contained in the image of the star. To accomplish this, not only must the starlight

## Chapter 7: Image Analysis



**Figure 7.7** Although features can fool the human eye, they do not fool the image profile. The profile reveals that spiral arms, which dominate the visual impression in this image, are really rather minor ripples in the overall radial decline in brightness outward from this galaxy's nucleus.

be totaled, but the light from the sky must also be measured and subtracted from the starlight. On the image, therefore, the procedure must measure two regions of interest, one containing the star (the *aperture*), and a second consisting of a representative sample of sky (the *annulus*). The aperture contains the star plus the sky background, and the annulus contains only the sky. Subtracting the area-weighted annulus from the aperture should yield the total starlight.

1. From an initial rough estimate of the star's location, determine the centroid of the star image;
2. determine which pixels belong to a region of interest (i.e., the star aperture) centered on the centroid of the star image, containing both its core and aureole;
3. count the number of pixels and add up the total pixel value contained inside the star aperture;
4. determine which pixels belong to a ring-shaped region of interest (i.e., the sky annulus) surrounding the aperture; or alternatively, locate and

- determine which pixels belong to a nearby region of interest (i.e., a sky aperture) containing only sky pixels;
5. make a meaningful determination of the sky pixel value in the annulus or sky aperture;
  6. from the total pixel value in the aperture, subtract the product of sky pixel value times the number of pixels in the aperture. The result is the total pixel value of the star.

The radius of the star aperture should be four to five times the full-width half-maximum of the star image on the CCD, or between 5 and 20 arcseconds—depending on the seeing quality, focal length of the telescope, tracking quality, and size of the pixels on the CCD. For typical amateur instruments and CCD cameras, the optimum radius usually lies between 4 and 10 pixels, so that the region of interest contains between 60 and 360 pixels.

The most difficult part of measuring stellar brightness is determining a meaningful pixel value of the sky. For an aperture containing 100 pixels, an error of one ADU in the sky brightness generates an error of 100 ADUs in the total pixel value of the star. Thus, a key requirement in digital photometry is an accurate determination of the sky brightness.

The radius of the annulus is typically twice the radius of the aperture, and contains approximately three times as many pixels. The larger the number of sky pixels, the greater the statistical accuracy of the sky background measurement, but the greater the risk of encountering contaminating background stars.

The average brightness of the annulus is one obvious possibility, but the average is sensitive to contamination from background stars. Their light adds to the sky brightness and drops the measured star brightness. The median pixel value of the sky background is another option, good because the scattering of high pixel values from faint background stars will have a negligible affect on the median. The median value of a set of pixels, however, is quantized into steps of one ADU, so that there is an automatic error of  $\frac{1}{2}$  ADU built into the photometry.

Similar to the median, but free of quantization, is an average built from the middle of the distribution of sky pixel values. Excluding the top and bottom 20% of sky pixel values and taking the average of the middle 60%, this “mean of the median half” method avoids both contamination from stars in the annulus and quantization effects. Repeated measurements on images with uniform sky backgrounds suggest that the “mean of middle” method of measuring the sky background is consistent to better than 0.1 ADU.

- ***Tip:*** **AIP4Win’s Star Image Tool extracts the basic measurements used in measuring the properties of star images. For stellar photometry, the suite of photometry tools is designed to carry out high-quality measurements of star brightness.**

### 7.9.2 Photometric Image Profile

To measure the total light from a star with any accuracy, the aperture must contain

## Chapter 7: Image Analysis

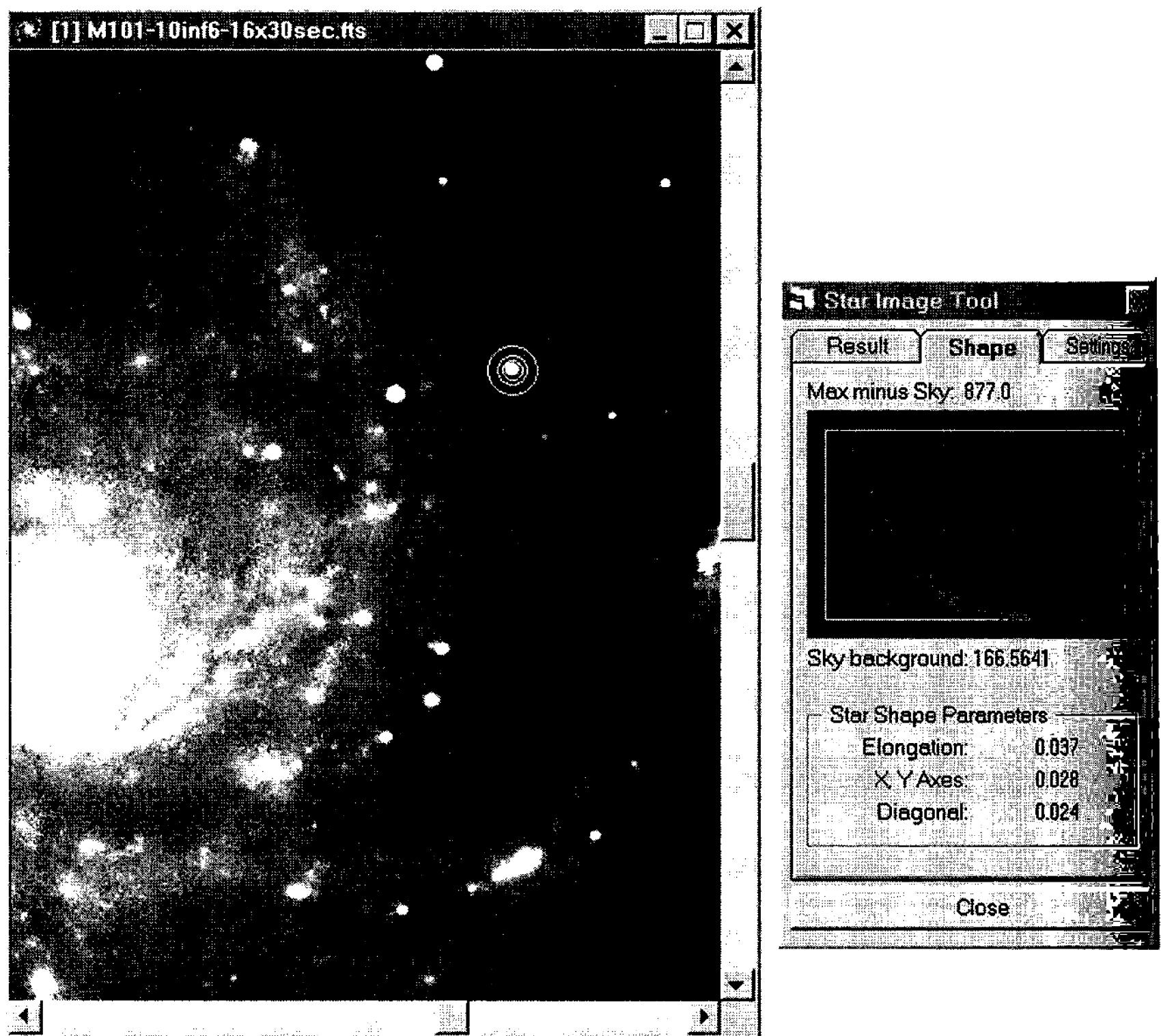


Figure 7.8 The profile of a star image shows a smooth decline in brightness from its centroid. From the profile, you can estimate the radius (or “half-width”) of the star image at half maximum as about 1.1 pixels. The curve of growth shows how much of the star’s light falls inside a circle with a given radius.

as much of its light as possible. Superficially, a star in a CCD image looks like a blocky circle with a distinct edge; but in reality, the image is bright in the center (the core) and fades smoothly into the background sky (the aureole). Good photometry requires measuring not only the light in the core but also the light in the aureole.

To form a realistic estimate of the extent of the star image, you can examine its profile. The profile is simply a graph of pixel values around a star image—the pixel value versus the radial distance of the pixel from the centroid of the star image. With the exception of bright stars near the saturation level of the CCD, *all star images in a given CCD image show exactly the same profile independent of the star’s brightness*. A typical profile falls rapidly from a maximum value, then more gradually approaches the sky value until the contribution of the starlight is lost in the noise of the sky background. This profile closely resembles the Gaussian, or normal, curve of distribution.

From the star profile plot, you can easily estimate the radius at which the

## Section 7.10: Spectroscopy

star's light falls to half the peak. This radius is called the “half-width at half-maximum,” or HWHM; the diameter of the image is its full-width at half maximum, or FWHM. If the profile were truly Gaussian, then the radius containing 99% of the star's light is  $2.2 \times \text{HWHM}$ ; and the radius containing 99.9% of it is  $2.8 \times \text{HWHM}$ . To be sure of including all the starlight and minimizing the star's contribution to the sky background, many observers prefer to set the radius of the aperture to at least  $4 \times \text{HWHM}$ .

- **Tip:** *The Star Image Tool in **AIP4Win** displays a photometric profile of a star image.*

## 7.10 Spectroscopy

Spectroscopy is the science of learning about celestial objects from their spectra. In CCD images, spectroscopic analysis begins with defining the location of the spectrum in the CCD image, and then extracting the integrated light at each wavelength from the image. Further analysis is carried out using the extracted data on intensity versus wavelength. For a description of spectroscopy, see Chapter 11.

## Chapter 7: Image Analysis

---

# 8 Measuring CCD Performance

---

In this chapter, you will learn two methods for assessing the performance of your CCD camera. The first—basic CCD testing—is so simple that you can make the necessary observations in a few minutes during the course of your normal observing program, and the second method—advanced CCD testing—yields a complete performance profile on your CCD.

Five characteristics define the performance of a healthy CCD camera:

- the conversion factor, or gain;
- the readout noise;
- the CCD's linearity;
- the CCD's uniformity; and
- the dark current.

Basic CCD testing checks only the first two items—the conversion factor and the readout noise—while advanced testing checks all five characteristics, plus allows you to spot radio-frequency interference and “charge skimming” as well.

- **Tip:** *AIP4Win includes a complete suite of functions needed to determine the performance of a CCD camera from a set of test images.*

## 8.1 Goals in CCD Testing

The primary goal of characterizing your CCD camera’s performance is to verify that it is operating correctly. Once you have established that the noise level is at or better than specification, your goal in testing will be to check its performance from time to time, to be sure that nothing has changed or degraded. In this way you can rest assured that your CCD camera is in tip-top shape.

When should you test your CCD? Wait until the thrill of shooting your first images wears off, when you have started to wonder how to squeeze the maximum possible performance from your camera. If you do discover a problem, it can probably be corrected. At regular intervals—every six months or so—run a quick check to make sure the readout noise and dark current continue to meet the camera’s specifications.

The characteristics that you can test for are:

## Chapter 8: Measuring CCD Performance

**Conversion Factor.** The conversion factor, or gain, ties arbitrary pixel values in analog-to-digital units (ADU) to the physically meaningful number of electrons generated at photosites on the CCD. Measuring the conversion factor enables you to characterize other aspects of the performance of the CCD, such as the readout noise and dark current, as physically significant numbers of electrons.

**Readout Noise.** Readout noise is the irreducible bottom line for noise in a CCD chip. It is the random variation in its output when no signal is present. It is customarily expressed as the root-mean-square variation in the number of electrons detected by the CCD. Determining the readout noise allows the performance of a CCD to be compared with the manufacturer's specifications, and also with other CCDs. If the measured readout noise is worse than specification, you can locate and possibly correct this problem.

**Linearity.** Ideally, the pixel value is directly proportional to the light that has fallen on the CCD. When this is the case, images from the camera can be precisely dark-subtracted and flat-fielded for clean, good-looking images, as well as used for precise astrometric and photometric measurements. As the charge wells on the CCD approach saturation, however, many CCDs become nonlinear. The goal of linearity testing is to verify the linearity of the CCD over its full dynamic range, or to determine the range over which it is linear so that you can optimize the amplifier gain.

**Uniformity.** The photosites on the CCD vary in their sensitivity to light. Between adjacent pixels the variations are typically less than 1%, and the variation may reach 10% across the entire CCD. Characterizing its uniformity establishes that a CCD is operating within normal parameters.

**Dark Current.** The accumulated charge from thermally generated electrons grows linearly with time. Electrons are generated in the neutral bulk silicon, in the charge depletion region, and in surface states at the interface between the bulk silicon and the silicon dioxide insulation layer. Cooling the CCD reduces the rates at which thermal electrons are generated. In cameras that can operate with inverted biasing, dark current can be reduced significantly. Operating the CCD with inverted bias (also called "multi-pinned phase" mode) reduces the contribution from surface states, but also reduces the full-well capacity. The goal of this test is to verify the correct operation of inverted bias mode and to measure the dark current in physical units (electrons per second per pixel) for comparison with the manufacturer's specifications and with other CCDs.

**Interference and Noise Pickup.** Although readout noise is intrinsic to a CCD, noise from a variety of other sources is not. CCDs can pick up noise from many sources in and around the telescope. If noise is found in the test set, its characteristics can be used to identify the source so that it can be reduced or eliminated.

**Charge Skimming.** Sometimes photosites that contain defects or impurities can trap (or "skim") electrons and subsequently release them when the CCD is read out. Since the skim charge is typically fewer than 100 electrons, charge skimming is troublesome only when signal levels are quite low.

## 8.2 Basic CCD Testing

To be practical, CCD tests must be simple enough to be carried out without fancy test equipment. You can do basic testing without even taking your CCD off the telescope! This consists of shooting two bias frames, two flat-field frames, and one dark frame. That's it. The analysis takes about five minutes. Details are given in Section 8.2.1.

Advanced CCD testing is more complex. It requires building a low-level light source ( $L^3S$ ), and with it shooting nine bias frames, nine low-level flat-field frames, 32 flat frames taken with differing integration times, and three long-integration dark frames. The test sequence can be done any place that can be darkened, including indoors on a cloudy night or in your observatory. Shooting a set of advanced test images takes about three hours, and analyzing the results takes about an hour. You will find a complete description of advanced testing in Section 8.3.

### 8.2.1 How to Make Basic CCD Test Images

The purpose of making initial test images is to get a quick measurement of the conversion factor, readout noise, and dark current of your CCD camera. This method gives you accurate results, and it takes only a few minutes. Furthermore, you can test the camera while it is attached to your telescope, so you can easily fit it into your schedule during a night's observing.

To make the flat-fields, you will need a light box like the  $L^3S$  described later in this chapter, the evening sky, or your usual method for making flats. Make the bias frames and the dark frame immediately before or after the flats so there's no time for the CCD or electronics to drift or change.

Turn on the camera and allow it to reach thermal equilibrium. Using the readout mode that you use to make celestial images, shoot the following:

**Two Flat Frames.** Make two flat frames using the same integration time. Adjust the integration time to produce average pixel values one-third to halfway to the maximum output of the camera. For 12-bit cameras, the flat should have an average value between 1,500 and 2,000 ADUs; and for a 16-bit ones, the flat should average around or 20,000 to 32,000 ADUs. Save the flat-frame files as BCT-FF1 and BCT-FF2.

**Two Bias Frames.** With the dark slide closed or with telescope capped, set the integration time of your camera to the shortest available, and then make two bias frames. Save them as BCT-BF1 and BCT-BF2.

**One Dark Frame.** Close the dark slide or cap the telescope, and make a dark frame using a 60-second integration time. Save the dark frame as BCT-DF.

### 8.2.2 Basic Test Analysis

These images contain enough information to determine the conversion factor, the readout noise, and the dark current.

## Chapter 8: Measuring CCD Performance

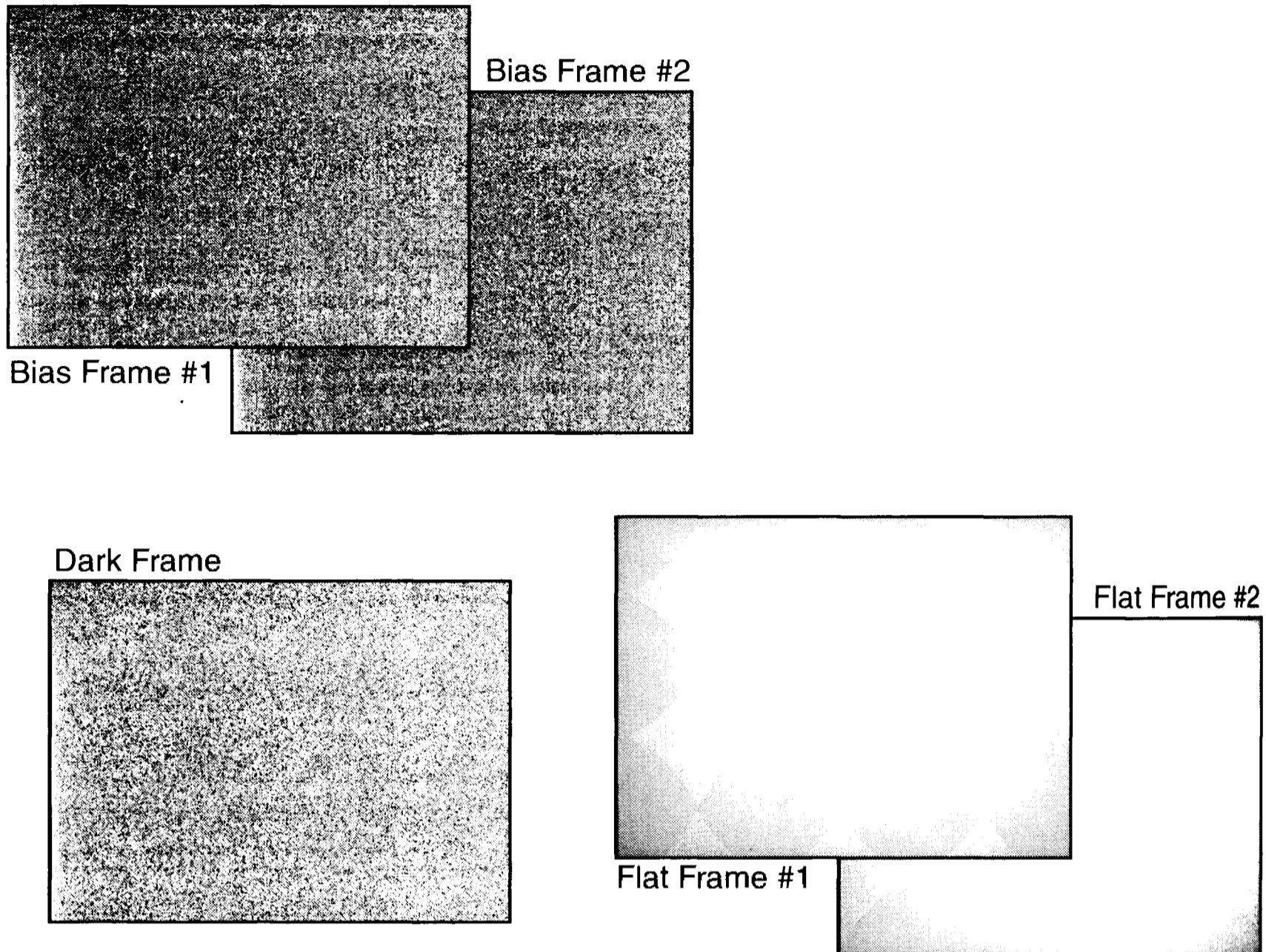


Figure 8.1 It takes only five images to determine your camera's conversion factor and readout noise. You can obtain a set of images like these without taking your CCD camera off the telescope during a regular observing session. Checking your camera assures that you are getting the performance you expect.

### 8.2.2.1 Step 1: Mean and Standard Deviation in the Bias

Load the two bias frames, BCT-BF1 and BCT-BF2. Add them together and measure the mean value of a region near the center of the image. The size of the region is not critical; an area 100 pixels on a side is entirely adequate. The mean value is the quantity  $\bar{B}_1 + \bar{B}_2$ . (The notation  $\bar{B}$  is read as "B-bar," and the bar over the  $B$  says that it's a mean value.)

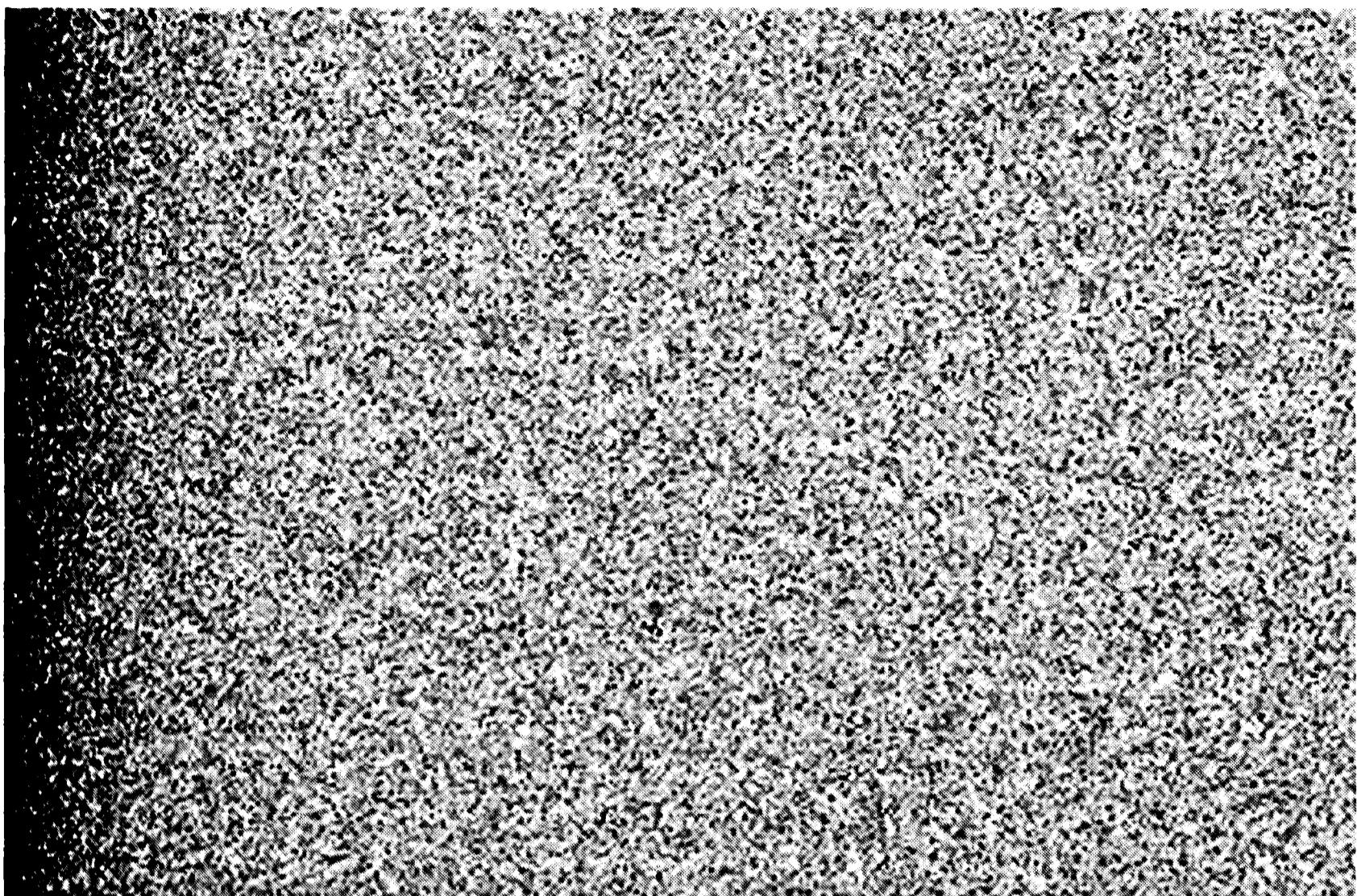
Next, subtract one bias frame from the other, and measure the standard deviation in pixel value for the same region at the center of the frame. Subtracting the two bias frames removes any fixed bias patterns, leaving just the noise from the two bias frames,  $\sigma_{B_1 - B_2}$ , which is  $\sqrt{2}$  times the noise of one bias frame.

- *Tip:* In **AIP4Win**, you can add and subtract images using the *Image Math tool*. Perform the measurements with the *Pixel Tool*.

### 8.2.2.2 Step 2: Mean and Standard Deviation of the Flats

Load the two flat frames BCT-FF1 and BCT-FF2. Add them together and then measure the mean value of a region near the center of the image. The result is  $\bar{F}_1 + \bar{F}_2$ .

Next, subtract one frame from the other, and measure the standard deviation in pixel value for a region near the center of the frame. Subtracting the two flat



**Figure 8.2** The difference between two seemingly identical flat-field frames is noise—in this case, the random variation in the number of electrons generated at each photosite. This image has been stretched by a factor of 20 relative to the two flat-frames, shown in Figure 8.1, that were subtracted to create it.

frames removes any features in the flat, leaving the total noise in the two flat frames,  $\sigma_{F_1 - F_2}$ , and the noise you have measured is  $\sqrt{2}$  times the noise in a single flat frame.

- *Tip:* In **AIP4Win**, you can add and subtract images using the *Image Math* tool. Perform the measurements with the *Pixel Tool*.

### 8.2.2.3 Step 3: Measure the Dark Current

Load the dark frame, BCT-DF, and bias frame BCT-BF1. Subtract the bias frame from the dark frame. Measure the mean pixel value for a region at the center of the frame. This is the dark current,  $D_{\text{ADUs}}$ , that accumulated during a 60-second integration, measured in ADUs.

- *Tip:* In **AIP4Win**, you can add and subtract images using the *Image Math* tool. Perform the measurements with the *Pixel Tool*.

### 8.2.2.4 Step 4: Compute the Conversion Factor

You have measured the mean of a high signal level and its standard deviation in ADUs. Because you expect the signal to display Poisson statistics measured in electrons, you expect  $\sigma_{\text{electrons}} = \sqrt{F_{\text{electrons}}}$ . However, since both  $\sigma$  and  $F$  have been multiplied by the conversion factor,  $g$  electrons per ADU, you have actually measured  $g\sigma_{\text{electrons}} = \sqrt{gF_{\text{electrons}}}$ . Solving then for  $g$ :

## Chapter 8: Measuring CCD Performance

$$g = \frac{F_{\text{electrons}}}{\sigma_{\text{electrons}}^2} \text{ [electrons/ADU].} \quad (\text{Equ. 8.1})$$

In fact, you have done an even better job because you have measured the bias mean and noise, so you can remove their influence by subtracting them.

To compute the conversion factor, evaluate:

$$g = \frac{(\bar{F}_1 + \bar{F}_2) - (\bar{B}_1 + \bar{B}_2)}{(\sigma_{F_1 - F_2}^2) - (\sigma_{B_1 - B_2}^2)} \text{ [electrons/ADU].} \quad (\text{Equ. 8.2})$$

In 16-bit cameras, the conversion factor is usually close to unity.

### 8.2.2.5 Step 5: Compute the Readout Noise

The only source of noise in a bias frame should be the readout noise. You have measured the sum of readout noise in two bias frames.

To compute the readout noise, evaluate:

$$\sigma_{\text{Readout}} = \frac{g\sigma_{B_1 - B_2}^2}{\sqrt{2}} \text{ [electrons r.m.s.].} \quad (\text{Equ. 8.3})$$

In 16-bit cameras, readout noise usually lies between 3 and 20 electrons r.m.s. The lower the readout noise, the better.

### 8.2.2.6 Step 6: Compute the Dark Current

Convert the dark current measured in ADUs to electrons per pixel per second by evaluating:

$$D_{\text{electrons}} = \frac{D_{\text{ADUs}}}{60} \text{ [e}^- / \text{pixel / sec].} \quad (\text{Equ. 8.4})$$

In a cooled CCD camera, the dark current is typically around 1 electron per pixel per second. However, in some CCDs the dark current may be as low as 0.001 electrons per pixel per second. For these low-dark-current CCDs, it may be necessary to use an integration time much longer than 60 seconds to obtain a measurably large accumulation of dark current.

The values you get should be in accord with those specified by the manufacturer for this operating mode of your camera.

## 8.2.3 What Results to Expect

The conversion factor is the number of electrons per ADU. It is determined by your CCD camera's electronics. The designer's goal is usually to divide the full-well capacity of the CCD chip into the number of steps resolved by the analog-to-digital converter:

**Table 8.1 Sample Basic CCD Test Data**

Frame	Sum	Std Dev'n of the Difference	Mean
Bias Frames	213.3	32.84	—
Flat Frames	39675	247.7	—
Dark – Bias Frame	—	—	6.43

$$g = \frac{(\bar{F}_1 + \bar{F}_2) - (\bar{B}_1 + \bar{B}_2)}{(\sigma_{F_1 - F_2}^2) - (\sigma_{B_1 - B_2}^2)} = \frac{39675 - 213.3}{247.7^2 - 32.84^2} = 0.65 \text{ [e}^- / \text{ADU]}$$

$$\sigma_{\text{Readout}} = \frac{g \sigma_{B_1 - B_2}^2}{\sqrt{2}} = \frac{0.65 \times 32.84}{\sqrt{2}} = 15.2 \text{ [e}^- \text{ r.m.s.]}$$

$$D_{\text{electrons}} = \frac{D_{\text{ADUs}}}{60} = \frac{32.84}{60} = 0.55 \text{ [e}^- / \text{pixel / sec}]$$

$$\frac{\text{full-well capacity}}{\text{levels available}} = \text{conversion factor.} \quad (\text{Equ. 8.5})$$

For example, for a commercial-grade chip with a full-well capacity of 80,000 electrons and a 16-bit analog-to-digital converter, most camera designers would set the conversion factor at some value close to  $80,000/65536 = 1.2$  electrons per ADU.

Readout noise is primarily determined by the CCD's on-chip amplifier. Science-grade chips have levels of 3 electrons r.m.s. or better; top-notch commercial-grade models have readout noise around 6 electrons r.m.s.; and CCDs made for video camera applications and digital SLRs typically have a level between 8 and 30 electrons r.m.s. If your camera has a CMOS sensor, you can expect the conversion factor and readout noise to be larger than you might find in a comparable CCD, but nonetheless in the same range you would expect for a high-quality video camera.

Dark current depends critically on the intrinsic properties of the sensor and the temperature of the sensor. A CCD that has a dark current of 1 electron per pixel per second when it's cooled to  $-30^\circ\text{C}$  will have between 100 and 1000 times that amount when operating at room temperature. Low-dark-current technologies can push the dark current down to roughly 10 electrons per pixel per second at room temperature; and these devices, when cooled to  $-10^\circ\text{C}$ , turn in quite remarkable dark current figures. CCDs made by Sony employ a proprietary process that produces very low dark currents at room temperature, and extraordinarily low levels—on the order of 0.002 electrons per pixel per second—when cooled to  $-10^\circ\text{C}$ . With such low dark currents, dark subtraction may not immediately appear to be necessary, but remember that dark-frame subtraction also removes the bias value, which is necessary for proper flat-fielding.

## Chapter 8: Measuring CCD Performance

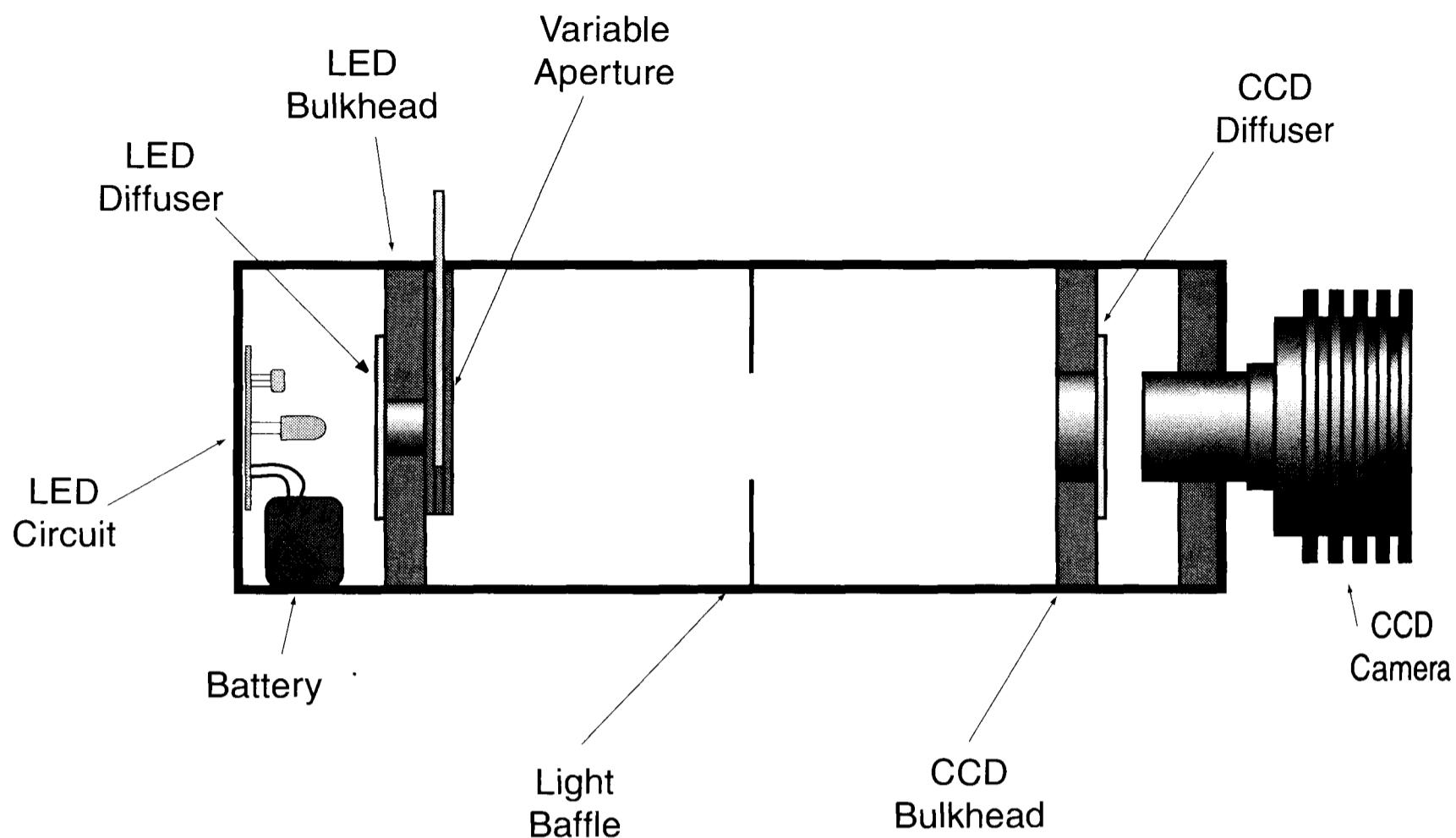


Figure 8.3 This low-level light source (L<sup>3</sup>S) provides a stable, faint source of light for advanced CCD testing. A circuit-stabilized LED emits light that is diffused by an opal-glass or milk plastic diffuser. The variable aperture controls how much light goes to the second diffuser and reaches the CCD.

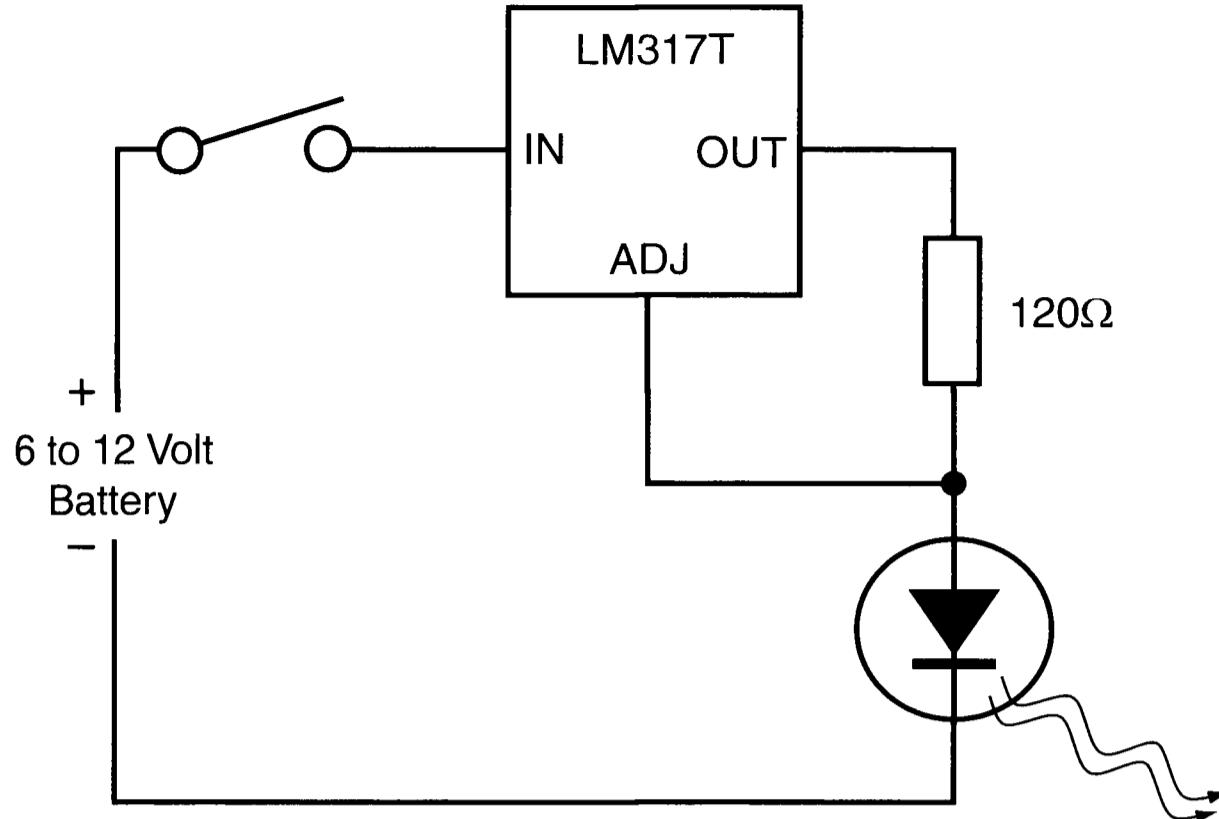
## 8.3 Advanced CCD Testing

A full set of test images for advanced CCD testing contains enough information for you to determine the conversion factor, readout noise, linearity, uniformity, and dark current—as well as to check for interference and charge skimming. Advanced CCD testing is a must if you want to know for sure that your camera is delivering the high-quality images that you expect.

### 8.3.1 Constructing a Low-Light Level Source

The purpose of the low-level light source (or L<sup>3</sup>S) is to provide the stable and uniform source of faint illumination that you will need to make flat-field frames for the advanced CCD test sequence. The apparatus is similar to that used by professional astronomers to assess the performance of the megapixel CCDs on giant telescopes, yet it is surprisingly simple to construct. If you have experience building optical devices (such as telescopes), you may wish to modify the design to suit your needs.

The L<sup>3</sup>S is a box made of  $\frac{1}{4}$ -inch plywood 6 inches on a side and 18 inches long. A light-emitting diode (LED) supplies light at one end, and the CCD is attached to the other. At the LED end of the box is a  $5\frac{1}{2}$ -inch square piece of  $\frac{3}{4}$ -inch plywood that holds a diffusing screen. In front of the diffusing screen, you construct a holder for cardboard slides that carry holes drilled in thin aluminum or brass. The size of the hole determines how bright the L<sup>3</sup>S appears. The center of



**Figure 8.4** This simple electronic circuit keeps the LED in the L<sup>3</sup>S at a constant brightness by maintaining a constant current through it. This protects your measurements against the LED's dimming as the batteries run down. Because the circuit draws very little current, batteries will last a long time.

the box is empty except for a blackened baffle to reduce scattered light. At the CCD end of the box, a second bulkhead made of ¾-inch plywood holds another diffusing screen. A third bulkhead supports the CCD camera. Figure 8.3 shows an overview of its construction.

The light source is an LED run by any battery from 6 to 24 volts, including four AA cells, a 6-volt lantern battery, a 9-volt transistor battery, or an automobile battery. To insure that it is always the same brightness, the current through the LED is held at a constant 10 milliamperes by an LM317T adjustable voltage regulator. The choice of LED is not critical. Any red, orange, yellow, or green LED with a forward voltage of 2 volts and rated for currents between 10 and 20 milliamperes should serve equally well.

Note: The voltage drop across the resistor depends on the current that flows from the OUT pin and through the resistor and LED. The LM317 adjusts the voltage on the OUT pin to maintain a constant 1.25 volts difference between the OUT and ADJ pins. The LM317 therefore controls the LED current,  $I_{LED}$ , according to the formula  $I_{LED} = 1.25/R$ . When you set  $R$  to 120 ohms, the LM317 increases the voltage on the OUT pin until a current of 10.4 milliamperes flows through the LED. The current through the LED does not depend on the voltage applied to the IN pin, so you can run the circuit with any voltage greater than 4.5 volts. This circuit is designed for battery operation, and the current drawn is so small that virtually any battery will run the LED at a constant brightness for many hours.

**The Lamp Section.** The light source is separated from the midsection of the L<sup>3</sup>S by the bulkhead that carries the LED diffuser and the aperture slide. The amount of light that reaches the CCD depends on the distance between the LED

## Chapter 8: Measuring CCD Performance

**Table 8.2 Aperture Slides for a Low-Level Light Source**

Aperture	Number of Holes	Diameter (inches)	Relative Illumination
A1	1	1	1024
A2	2	1/2	512
A3	1	1/2	256
A4	2	1/4	128
A5	1	1/4	64
A6	2	1/8	32
A7	1	1/8	16
A8	2	1/16	8
A9	1	1/16	4
A10	2	1/32	2
A11	1	1/32	1

and the diffuser; therefore, build the lamp section to allow you to place the lamp immediately behind the diffusing screen or considerably further back. Drill a hole for an on-off switch, and install clamps to hold the battery firmly in place.

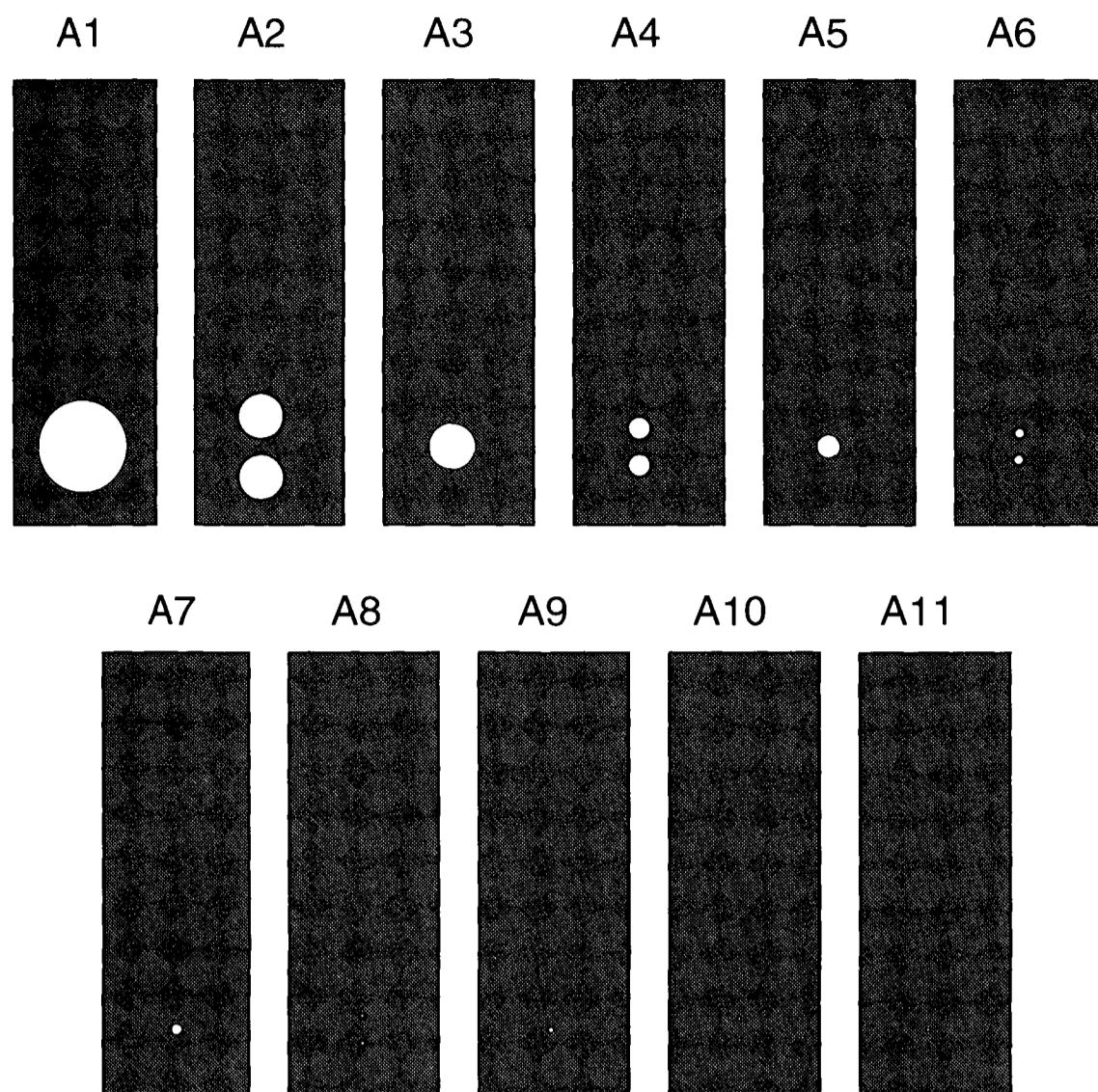
**The LED Diffuser.** This screen is a piece of opal glass or milk plastic roughly two inches on a side. Opal glass and milk plastic are translucent materials that completely scatter light. Both materials look like milk—smooth and uniformly white. Mount the diffuser on the LED side of the LED bulkhead.

**The Slide Holder.** This is mounted on the aperture bulkhead so that you can insert slides from the top of the box. Make the holder from heavy cardboard or 1/8-inch plywood. Paint the parts black to minimize light leakage.

**The Aperture Slide.** The purpose of the aperture slide is to pass the right amount of light to make well-exposed flat-frames over the range of exposure times needed. A complete set of aperture slides provides a 1:1000 range of intensity, but you need only one (the one that's the right size) to test your digital sensor.

If you want to simplify the design by making only one slide, follow the plan below but make one slide instead of eleven. Aperture A5 is right in the middle of the brightness range, and is probably the one you will need.

To make a complete set of slides, make 11 slides 2 inches wide by 6 inches long from heavy cardboard or light plywood. One inch from the bottom end of each slide, chisel a recess half the thickness of the material and 1 1/4 inches square; and in the center of the recess, drill a hole 1 inch in diameter. (See Figure 8.5.) Number the slides from A1 to A11. They carry a set of holes drilled through thin aluminum or brass shim 1 1/4 inches on a side. Drill the holes as shown in Table 8.2, and glue each piece of thin metal into the recess in the slide. Each slide passes half the light of the next larger one in the series.



**Figure 8.5** A full set of eleven L<sup>3</sup>S apertures allows you to set the light level in a completely repeatable way over a range in brightness of 1000 to 1, but you only need to make as many as you need to get good test images. The apertures are holes drilled in thin aluminum or brass shim stock.

**The Midsection.** Between the LED bulkhead and the CCD bulkhead, insert a cardboard or thin plywood baffle to reduce scattered light. Cut a hole 2 inches in diameter in the center of the baffle and install it in the box. The LED bulkhead and the CCD bulkhead will be approximately 10 inches apart.

**The CCD Bulkhead.** Cut a 2½-inch hole in the bulkhead. On the side of the CCD bulkhead that faces the camera, mount a second diffuser 3 inches square made of opal glass or milk plastic. The fully illuminated area of the diffuser should be at least 2¼ inches in diameter.

**The CCD Holder.** Provide a solid mounting for your camera on the end of the L<sup>3</sup>S box. It should be held well enough that it will not wiggle or shift if you need to move the L<sup>3</sup>S. If your CCD camera accepts 2-inch O.D. accessories, for example, the mounting can be as simple as a 2-inch diameter hole lined with black felt. Try to keep the separation between the diffusing screen and the CCD as small as practical to insure even and shadow-free illumination of the chip. Line the hole with black felt so that when you insert the camera tube into the opening, you have a light-tight seal.

Assemble the L<sup>3</sup>S box with glue and wood screws. Carefully seal around the aperture bulkhead and slide holders to prevent light leaks. Paint the interior flat black. Using bright light, check that no light sneaks around the edges of the aper-

## Chapter 8: Measuring CCD Performance

ture slide. When you turn on the LED with the largest aperture in place, you should just be able to see a dim glow on the second diffusing screen.

It is not necessary to adhere rigidly to these dimensions and materials. Any method you can work out for providing an adjustable, low-level, diffuse illumination will serve for advanced CCD testing.

### 8.3.2 Setting the Low-Level Light Source for Use

Setting the brightness of a newly built L<sup>3</sup>S usually involves making trial-and-error adjustments of the lamp placement. The goal is to place the lamp behind the first diffusing screen at a distance such that using a 100-second integration with the aperture A5 (the one in the middle of the brightness range), the CCD generates a signal of about one-fourth of the full-well capacity—about 1,000 ADUs for a 12-bit camera and 16,000 ADUs in a 16-bit camera, above the bias level. These settings mean that the L<sup>3</sup>S will be able to generate the brightest and faintest levels you will need in testing.

Install the CCD in the L<sup>3</sup>S. Swaddle the end of the box and the CCD in black cloth, then turn on the camera's electronics and cooling system and allow it to reach equilibrium—at least 15 minutes. With air-cooled CCD cameras, be sure to allow the free circulation of air around the cooling fins.

Set the software to the readout mode that you most often use to make celestial images. With aperture A5 in place and the LED turned off, make a dark-frame integration of 100 seconds, and save it as a test dark frame. Turn on the LED and make a second integration of 100 seconds; save it as a test bright frame.

Subtract the test dark image from the test bright image, then measure the average pixel value generated by the light from the L<sup>3</sup>S. Repeat the process of making and measuring images, altering the position of the lamp as necessary, until with the aperture in place, the average pixel value in a 100-second integration is  $1000 \pm 200$  ADUs for a 12-bit camera or  $16,000 \pm 3,000$  ADUs for a 16-bit model. Once you have set the output, you are ready to begin testing.

If you cannot get the right brightness using aperture A5, then try larger and smaller apertures until you find the one that gives you the right brightness.

- **Tip:** *In AIP4Win, load the test bright frame and the test dark frame. Use the Multi-Image tool to subtract the test dark frame from the test bright image. Use the Pixel Tool to measure the average pixel value of a region near the center of the frame.*

### 8.3.3 How to Make Advanced CCD Test Images

Make test images after the CCD has been running long enough to reach thermal equilibrium. If in doubt, allow the camera to run for at least 60 minutes in the L<sup>3</sup>S. The purpose of waiting this long is to minimize temperature changes during the test. For your initial runs, use the readout mode that you most often use to make celestial images. You may later wish to test your camera in readout modes that you use less often.

The full set of test images includes multiple bias frames, multiple skim frames, multiple flat-field frames at different integration times, and multiple dark frames. All of the images should be taken in a single session. Work in an area with very low ambient light levels, and cover the L<sup>3</sup>S and CCD camera with black cloth to prevent any light leaks. Give yourself two to three hours to take the complete set of images.

### 8.3.3.1 Make the Bias Frame Set

Bias frames are integrations made with no light falling on the CCD for the shortest integration time your camera's software allows. The only contributions to a bias frame should be the zero-point offset (i.e., the image bias), readout noise from the CCD's amplifier, and any electronic interference that is present.

Make the bias frames as follows:

1. turn off the LED so no light falls on the CCD;
2. set the shortest integration your CCD allows;
3. shoot nine bias frames;
4. save the bias frames as BIAS001 through BIAS009.

### 8.3.3.2 Make the Skim Frame Set

To detect charge traps in the CCD, you need flat frames made at a very low light level. The total charge that accumulates during integration should be less than 300 electrons, and the integration should be sufficiently short that hot pixels do not exceed this value. If the L<sup>3</sup>S is properly set up using aperture A5, an integration time of 5 seconds with aperture A9 ( $\frac{1}{16}$  as much light) should produce the correct number of signal electrons.

1. Turn on the LED;
2. shoot nine skim frames;
3. save the skim frames as SKIML001 through SKIML009;
4. turn off the LED;
5. shoot nine skim dark frames;
6. save the skim darks at SKIMD001 through SKIMD009.

Note: since you know the conversion factor between pixel value measured in ADUs and the number of electrons, divide 300 electrons by the conversion factor to obtain the signal level in ADUs.

### 8.3.3.3 Make the Flat Frame Set

To determine the conversion factor (electrons per ADU) and test the linearity of the CCD, make a series of flat frames of increasing exposure. As a check against the brightness of the LED changing during the test, make the flats in two sets, one with increasing integration times and a second set with decreasing integration times. Use aperture A5 to give the desired signal levels with integration times from 0.5 to 100 seconds.

## Chapter 8: Measuring CCD Performance

For the increasing series, execute steps 1 through 5 below with integration times of 1, 3, 5, 9, 15, 25, 45, and 80 seconds. Name the flat frame integrations FLATxxA and FLATxxB, and name the flat darks FLATxxD, with “xx” designating the integration time. For example, call the group of 1-second integrations FLAT01A, FLAT01B, and FLAT01D.

As soon as you reach the last of the increasing series, begin the decreasing series. Execute steps 1 through 5 with integration times of 60, 30, 20, 12, 6, 4, 2, and 0.5 seconds. Use the same file naming system.

1. Turn on the LED;
2. set the integration time to xx seconds;
3. take two flats;
4. save them as FLATxxA and FLATxxB;
5. turn off the LED;
6. take one dark frame;
7. save the dark frame as FLATxxD.

Remember that each set consists of *three* frames per integration time—two flats and one dark. It is easy to get confused when you are carrying out boring and repetitive operations. Work carefully and stay alert.

### 8.3.3.4 Make the Dark Frame Set

The final step is to make a set of dark frames with an integration time long enough to produce dark current in all pixels, but short enough that the hottest hot pixels do not saturate. An integration time of 500 seconds should satisfy these requirements.

1. Turn off the LED;
2. shoot three dark frames;
3. save the dark frames as DARK001, DARK002, and DARK003;
4. shoot a bias frame;
5. save the bias frame as DARKBIAS.

This completes the set of test images. You will have nine bias frames (BIAS001–BIAS009), nine skim frames (SKIML001–SKIML009), nine skim dark frames (SKIMD001–SKIMD009), 32 flats and 16 flat darks (FLATxxA, FLATxxB, FLATxxD, etc.), three long-exposure dark frames (DARK001–DARK003), and one dark bias (DARKBIAS).

### 8.3.4 Analyzing the Advanced Test Image Set

The most important thing to bear in mind as you analyze the test image sets is to look at them with an eye that is both analytical and quantitative. Pattern noise in the bias frame might look like a mountain range but have an amplitude of 0.05 ADU and therefore be insignificant, while a uniform doubling of the readout noise is devastating. Remember that your ultimate goal is making astronomical images.

**Table 8.3 Sample Bias Frame Data**

Frame	Mean PV $\bar{S}_{pv}$	Minimum PV	Maximum PV	Standard Deviation $\sigma_{pv}$
BIAS1	100.21	99	102	0.761
BIAS2	100.17	99	102	0.779
BIAS3	100.24	99	102	0.773
...	...	...	...	...
BIAS-MED	100.21	99	102	0.771

### 8.3.4.1 Check the Bias Frames for Noise and Interference

Load the nine bias frames in the bias frame set. Measure the mean pixel value,  $\bar{S}_{pv}$ , and the standard deviation of the pixel value,  $\sigma_{pv}$ , in each bias frame. Tabulate the values as shown in Table 8.3.

- **Tip:** In **AIP4Win**, measure the mean pixel value and standard deviation using the Pixel Tool. Set the outer radius to 25 pixels, and the sample shape to square. Click Get Statistics and read the mean value. Measure the same part of each bias frame.

Make a median bias frame and save the resulting image as BIAS-MED.FTS.

- **Tip:** Use the Median Image Tool in **AIP4Win** to generate the median bias frame.

Examine each of the bias frames and the median bias frame. An ideal bias frame has a salt-and-pepper scattering of pixel values.

The readout noise in an ideal bias should vary randomly about an average value (the bias value) with a Gaussian distribution. The standard deviation of the variation is the readout noise. Apply a linear scaling that strongly accentuates the noise and interference present; then make a histogram of each bias frame. The histogram should show a Gaussian distribution.

- **Tip:** In **AIP4Win**, use Enhance, Brightness Scaling to set the linear stretch thresholds just below the lowest pixel value in the image and just above the highest pixel value in it. Examine the image histogram to see if the distribution of pixel values is reasonably similar to Gaussian. On a logarithmic plot, a Gaussian distribution appears as an inverted parabola.

On visual examination, bias frames often show variations from top to bottom and left to right, patterns of various kinds, as well as random and pseudo-random variations in the intensity of the lines and columns. In each of the bias frames, look for the following:

- Wavy patterns with sinusoidal or repetitive structure, either across the columns (relatively high frequency) or down the

**Table 8.4 Sample Transfer Curve and Linearity Data**

Flat Set	Integration (seconds)	Mean PV $\bar{S}_{pv}$	Variance $\sigma_{pv}^2$	Count Rate (ADUs/sec)
FLAT01	1	59.2	2.98	59.20
FLAT03	3	171.2	7.35	57.06
FLAT09	9	469.8	19.05	52.20
FLAT15	15	769.8	31.05	51.32
FLAT25	25	1377.3	49.78	55.09
FLAT45	45	2417.4	83.14	53.72
FLAT80	80	saturated	—	—
FLAT60	60	saturated	—	—
FLAT35	35	1865.9	65.67	53.31
FLAT20	20	1101.8	41.33	55.09
FLAT12	12	685.1	26.26	57.09
FLAT06	6	340.0	13.46	56.67
FLAT04	4	224.1	9.40	56.03
FLAT02	2	112.2	5.25	56.10
FLATP5	0.5	24.3	1.86	48.60
<b>Data for Cookbook 245, 378-wide External Binning, Low-Dark-Current Mode</b>				

lines (relatively low frequency). If a pattern is present, it may vary randomly from line to line, or it may be fixed from one frame to the next. Patterns may originate inside the camera or may be interference from a nearby computer or monitor. Note whether the pattern is the same from one frame to the next (and therefore generated in the CCD electronics), or whether it changes from frame to frame (and therefore comes from a source that is not synchronized with the CCD and its associated electronics). Power supplies that exhibit ripple or switching transients are prime suspects. If the pattern repeats exactly, it can be averaged and subtracted out. If the pattern is different in each bias frame, it is important to identify and eliminate the cause.

- Burst noise will most likely appear as an isolated band across one or more bias frames. It may be caused by electrical equipment such as a motor operating in the house, in the observing area, or elsewhere on the same electrical circuit. The source of burst noise should be found, if possible.
- Popcorn noise (also called  $1/f$  noise) usually appears as a semi-random variation in the pixel values from line to line. It may originate in the camera electronics. Since popcorn noise does

not repeat from one bias frame to the next, it cannot be averaged and subtracted.

- Slopes are systematic variations across the bias frame from top to bottom or left to right. They may result from the bias injection part of the CCD electronics. Slopes are typically small in amplitude and highly repeatable, so they are subtracted during image calibration.
- Hot regions are caused by the CCD's on-chip amplifier, as is the case in the TC211 chip, or by charge leakage during readout. Hot regions that repeat precisely are routinely subtracted out during image calibration.
- Dark pixels are defects in the chip manufacturing process. The total number should be less than the total of “point defects” specified by the quality grade of the CCD.

Scrutinize the median bias in the same way you did each of the individual frames. Features of the bias frame that remain the same from frame to frame will be present in the median bias, but features that change should be absent. This is one of the best ways to determine whether a noise source is coherent (related to the camera's frame-taking process) or incoherent and therefore probably external to the camera system.

The outstanding characteristic of a good bias frame is blandness. The pixel value is nearly constant across the frame, and the random variation of the pixel values should itself be very bland. Repeatable noise features less than one ADU in amplitude are seldom cause for concern since they are subtracted during routine image calibration. However, random noise and burst noise sources should be identified and removed.

#### **8.3.4.2 Charge Skimming Check**

Charge skimming occurs when an abnormal photosite retains a few hundred electrons during readout. The characteristic signature of charge skimming is a dark pixel at the location of the charge “trap” and a dark tail of pixels following the trap.

Because the signal levels are so low in this check, it is necessary to shoot multiple frames and combine them to reduce the noise level. Furthermore, because of the hot pixels present even in short-integration flat fields, it is necessary to make and subtract a master dark frame to remove the hot pixels.

To check for charge skimming, make a median of the nine skim dark frames and save it as SKIMDMED.FTS. Make another median of the skim frames and save this frame as SKIMLMED.FTS. Add 1,000 to SKIMLMED and then subtract SKIMDMED from it. Examine this image carefully for dark pixels and their associated tails.

You can obtain some feel for the number of electrons caught in the trap by measuring how much lower the dark pixel is than the surrounding pixels, or by

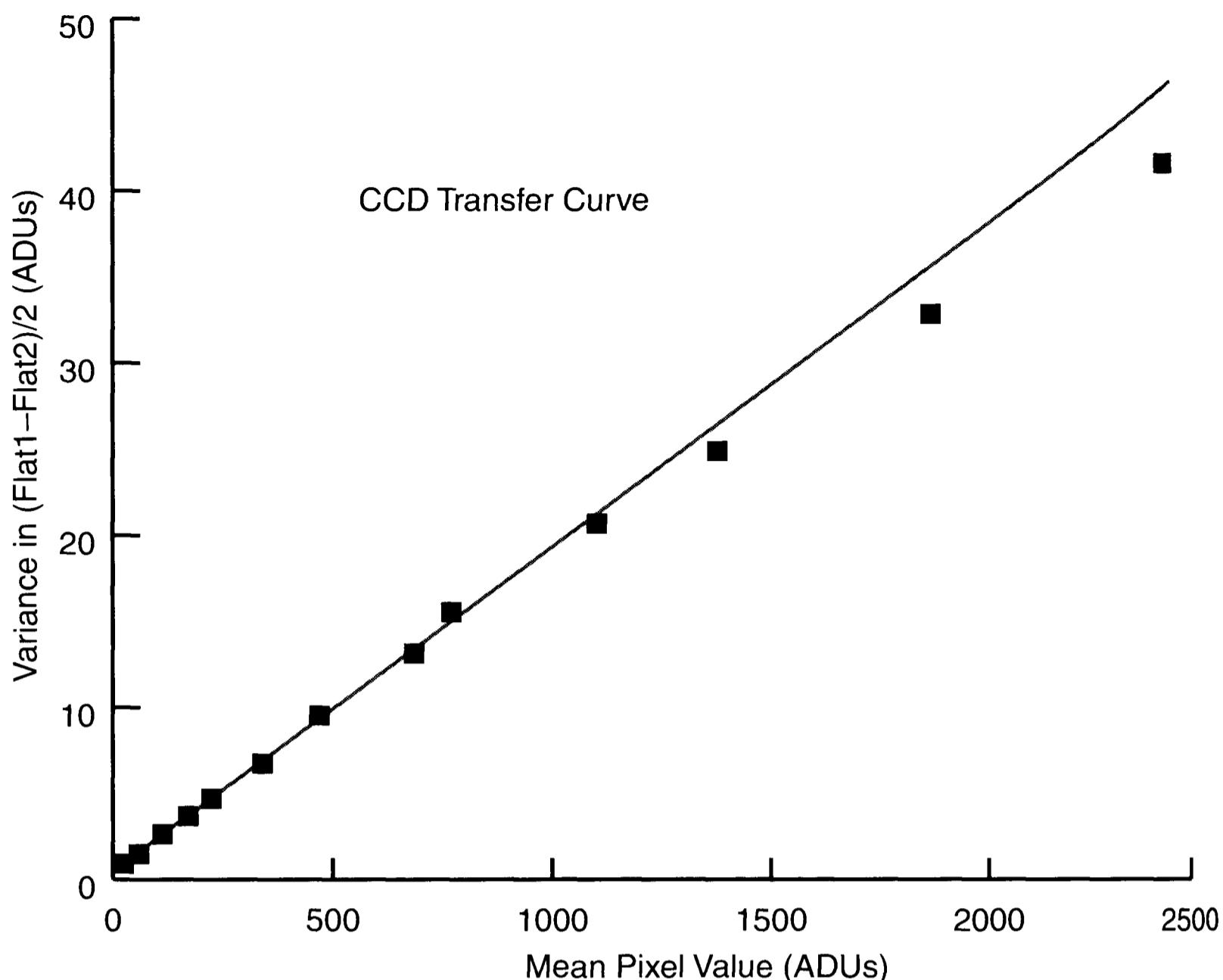


Figure 8.6 The CCD transfer curve graphs the relationship between mean pixel value of a uniformly illuminated region and the variance from the mean pixel value in the region. The inverse of the slope of the line is the conversion coefficient between ADUs and electrons, about 55 electrons per ADU in this example.

summing the excess pixel values in the tail. To convert the measured pixel value from ADUs to electrons, multiply by  $g$ , the conversion factor.

### 8.3.4.3 Plotting the Transfer Curve

The transfer curve and the test for linearity use the same set of measurements from the flat-field data, but the two tests use it in different ways.

The transfer curve is a graph of the variance,  $\sigma_{\text{pv}}^2$ , as a function of the average pixel value,  $\bar{S}_{\text{pv}}$ , for a set of flat-field images. The variance at different levels of pixel value is found by subtracting the pairs of flat-field frames. From the transfer curve, you can determine the conversion factor of your CCD in electrons per pixel value and the readout noise of the CCD. The theory behind this determination is really quite elegant.

Deducing the transfer factor relies on the counting statistics of events such as the generation of photoelectrons having a Poissonian distribution; that is, the standard deviation in the number of photoelectrons,  $\sigma_{\text{pe}}$ , from each sampling interval equals the square root of the average number of photoelectrons,  $\bar{S}_e$ . For example, if on the average, the light falling on a given photosite generates 10,000 photoelectrons per integration period, then the standard deviation will be

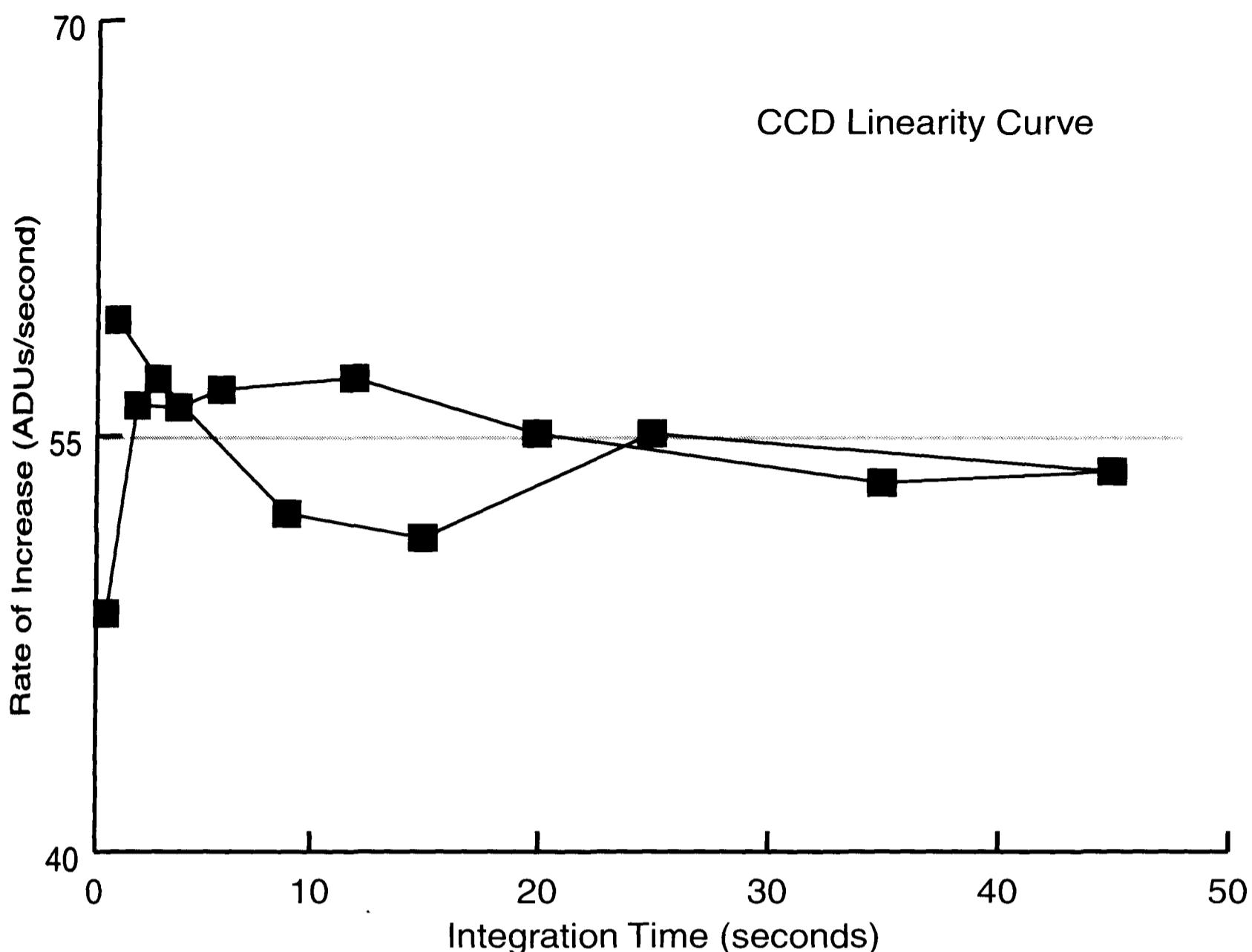


Figure 8.7 The gap in this CCD linearity curve suggests that the light source used in this test run—an ordinary tungsten bulb—probably did *not* remain constant in brightness when the data were taken, underscoring the importance of using an LED run at a constant current for advanced CCD testing.

$\sqrt{10,000}$  photoelectrons, and the number of photoelectrons expected to arrive during a given sampling period will be  $10,000 \pm \sqrt{10,000}$ .

In a healthy CCD, the signal consists of a combination of the number of electrons plus the random noise from Poisson counting statistics plus the readout noise of the CCD. Since standard deviations add in quadrature, we expect the standard deviation of the combined signal,  $\sigma_e$ , in units of equivalent electrons at the detection node of the CCD, to be:

$$\sigma_e = \sqrt{\sigma_{pe}^2 + \sigma_{ron}^2} \quad (\text{Equ. 8.6})$$

where  $\sigma_{pe}$  is the standard deviation in the number of photoelectrons and  $\sigma_{ron}$  is the readout noise expressed in electrons.

For each of the flat-field images, we can directly measure the average pixel value,  $\bar{S}_{pv}$ , in ADUs. The mean number of photoelectrons at the detection node,  $\bar{S}_e$ , equals  $\bar{S}_{pv}$  times  $g$ , the conversion factor for the number of electrons per pixel value unit:

$$\bar{S}_e = g\bar{S}_{pv}. \quad (\text{Equ. 8.7})$$

## Chapter 8: Measuring CCD Performance

For the photoelectron component of the electrons reaching the charge detection node, the standard deviation,  $\sigma_{pe}$ , equals  $\sqrt{\bar{S}_e}$ , so the variance,  $\sigma_{pe}^2$ , equals  $\bar{S}_e$ . Thus, by substitution:

$$\sigma_{pe}^2 = g\bar{S}_{pv}. \quad (\text{Equ. 8.8})$$

Another quantity we can measure is the standard deviation of pixel values in a flat-field frame,  $\sigma_{pv}$ . Even though flat-fields are never perfectly “flat,” by subtracting one flat-field image from another made under identical conditions, we can eliminate almost all nonuniformities, and thus determine the quadratic sum of the noise. The noise measured in electrons is scaled by the conversion factor into pixel value units:

$$\sigma_e = g\sigma_{pv}. \quad (\text{Equ. 8.9})$$

Note that the *measured* variance using the difference-of-two-flats technique is twice the variance of a single flat-field frame. You must divide each measured variance by 2 to obtain the variance of a single flat frame.

Recall that noise adds in quadrature, so that the total noise measured is the square root of the sum of the standard deviations of the photoelectron noise and the readout noise:

$$\sigma_e = \sqrt{\sigma_{pe}^2 + \sigma_{ron}^2},$$

so it follows that:

$$\sigma_e^2 = \sigma_{pe}^2 + \sigma_{ron}^2. \quad (\text{Equ. 8.10})$$

Substituting the equalities developed earlier into the noise equation:

$$g^2\sigma_{pv}^2 = g\bar{S}_{pv} + \sigma_{ron}^2. \quad (\text{Equ. 8.11})$$

Multiplying through by  $1/g^2$ , we obtain:

$$\sigma_{pv}^2 = \frac{1}{g}\bar{S}_{pv} + \frac{1}{g^2}\sigma_{ron}^2. \quad (\text{Equ. 8.12})$$

This is a linear equation in the standard form  $y = mx + b$ . You can measure  $(x, y)$  pairs,  $\sigma_{pv}^2$  and  $\bar{S}_{pv}^2$ , directly from the flat-field images. The slope of the line,  $m$ , is the inverse of the conversion factor,  $1/g$ , and the  $y$ -axis intercept,  $b$ , corresponds to the term:

$$\frac{1}{g^2}\sigma_{ron}^2,$$

which is the square of the readout noise in ADUs.

This relationship says that when you graph the variance of the signal against the signal itself, you should expect a straight line plot. Remember to divide the

## Section 8.3: Advanced CCD Testing

measured variance of [FLAT1 – FLAT2] by two, to obtain the variance for one flat-field frame.

The easiest way to handle these data is to enter them into a spreadsheet. Use the method of least squares to solve for the best-fitting value of  $1/g$ . Although it is possible to solve for the readout noise from the intercept of the transfer curve, it is easier and more reliable to measure the standard deviation from the bias frames and convert it into electron units.

### 8.3.4.4 Checking the Linearity of the CCD

You can check the linearity of the CCD by comparing the rate at which the pixel value increases for different integration times. In a camera that is linear, the rate of increase of the pixel value is the same for all integration times.

To compute the rate of increase, divide the mean pixel value,  $\bar{S}_{pv}$ , by the integration time. If the CCD was perfectly linear, the graph of rate of increase against the integration time would be a horizontal line. However, if the CCD becomes nonlinear at high pixel values as it nears saturation, the counting rate at the high-value end of the graph will drop off. If the lamp in the L<sup>3</sup>S changed in brightness while you were taking the flat fields, the count rates you took in the ascending exposure series of flats will not overlap the count rates you took during the descending exposure series.

Many CCDs become nonlinear when the signal is greater than about half the full-well capacity. CCDs with active anti-blooming gates are intentionally made nonlinear in order to prevent excess electrons from blooming. If your CCD tests as nonlinear below half of the full-well capacity, its photometric performance will be poor; and it may not be possible for you to perform an accurate flat-field correction of your images.

### 8.3.4.5 Determining the Dark Current

The charge stored on the photosites of the chip decays with time through a leakage called dark current. Cooling the CCD and setting its bias voltage strongly affect the dark current. Although this test is designed to measure the dark current under normal operating conditions, it can also be employed to measure the optimum bias voltage during integration.

In many cameras, the dark current depends strongly on the bias applied to the anti-blooming gate during integration. When the CCD is run with the bias inverted during integration—variously called multi-pinned phase or low-dark-current mode—the dark current at most photosites drops by a factor of roughly 100. However, a residual population of photosites with surface-state defects continues to accumulate dark current at near the original level. During inverted operation, “hot” photosites show much greater dark currents than the average dark current per pixel—thus increasing the measured average dark current. To characterize the dark currents from such a device, you must determine both the average dark current and the dark current generated by “hot” photosites.

## Chapter 8: Measuring CCD Performance

To determine the dark current statistics, load the dark frames DARK001 through DARK003. Subtract the bias frame DARKBIAS from each of them, and then take the median of the three dark frames. To determine the dark current for any given pixel, multiply the pixel value in ADUs times the conversion factor; and then divide by the integration time in seconds.

### 8.3.4.6 Check the Uniformity of the CCD

Load one of the more fully exposed flat-field images and subtract the appropriate dark frame. Apply a stretch to highlight small differences in the image. Despite all of your efforts to illuminate the CCD uniformly, you will almost certainly see considerable structure in the flat-field image. Much of this probably originates from variations in the silicon boule from which your chip was made. Note the low and high scaling values; these typically differ by less than 5%.

- **Tip:** *To examine the uniformity of the CCD with high precision, make a large set of identical flat-field frames and flat-dark frames. Use the Create Median function to make a single flat-field image with very low noise, and examine it for structure. Unless the intrinsic nonuniformities in the CCD exceed 10% across the chip and 1% over short distances, the standard flat-fielding procedures will entirely remove the effects of these nonuniformities from your images.*

# 9 Astrometry

---

Astrometry is the science of measuring the positions and motions of celestial objects. Normal applications include measuring the position of newly discovered variable stars, novae, and supernovae (so that other astronomers can accurately locate and observe the new object), and measuring the position of newly discovered comets and asteroids (because precise positions are needed to determine their orbits). Observers today can measure CCD images or digitized photographs to obtain precise coordinates of celestial objects with unprecedented speed and ease.

Astrometry offers additional practical benefits to observers interested in celestial imaging and processing. With a simple three-star solution, you can identify an celestial object precisely and without resorting to finder charts—a benefit particularly valuable to variable star observers. In addition, astrometric calibration of an image returns the precise focal length of the telescope used for taking it, and its precise scale and orientation, allowing the observer to make accurate measurements of objects in the image.

In this chapter you will learn how astrometry draws on reference catalogs of stars with accurately known positions, gain insight into the theory behind measuring reference stars and deriving coordinate solutions, and explore the practical benefits of measuring images with electronic cameras.

## 9.1 Astrometric Catalogs and Coordinates

The foundation of astrometry is a twenty-three-century-long tradition of measuring stellar positions. The earliest star catalog known was compiled by the Greek astronomer Hipparchus that came to us in modified form in Ptolemy's great book, the *Almagest*. When European astronomy came back to life after the Dark Ages, the noble Dane Tycho Brahe produced a star catalog, as did the great Polish amateur observer Hevelius. They routinely achieved accuracies of a few minutes of arc using instruments with open sights (like rifle sights). Telescopes and advances in making precisely divided circles eventually pushed errors down to a few arcseconds in the work of John Flamsteed and Ole Römer.

Astrometry in the 19th century was accomplished with transit telescopes, by measuring the altitude and precisely timing the transit of stars across the celestial meridian. These precise measurements defined a *fundamental* reference frame

## Chapter 9: Astrometry

upon which other, less fundamental measurements of much larger numbers of stellar positions could be based.

### 9.1.1 The Astrometric Dilemma

Astronomical coordinates are based on the orientation of the rotational axis of our planet. Declination, analogous to terrestrial latitude, is defined as the angular distance north or south of the celestial equator of the star's apparent position. Right ascension is a longitude-like coordinate, traditionally measured as the length of time between the transit of the vernal equinox point and the star in question (which is why right ascension is expressed in hours, minutes, and seconds of time rather than degrees). Because the Moon and Sun pull on the equatorial bulge of Earth, its rotation axis precesses through space, circling the ecliptic pole with a period of some 26,000 years. Thus, the vernal equinox, the point of intersection between the celestial equator and the ecliptic, is not fixed in space; so both the declination and the right ascension of a star change with time. The Greek astronomer Hipparchus was the first to measure precession with reasonable accuracy.

Precession creates problems. When an astronomer measures the declination and right ascension of a star for a fundamental catalog, the data are direct measurements of the position of the star in the sky relative to the orientation of Earth's spin axis at the date of observation. Those coordinates apply only to that particular moment in time, called the *epoch* of the observation. Precession is reasonably well-behaved, however, so that even if an astronomer measures the position of a star on July 15, 2007, we can "reduce" its coordinates to what they would have been had they been measured on January 0.5, 2000, a standard epoch called J2000. The right ascension of a star near the celestial equator increases by 3.07 seconds of time per year, and the declination can increase or decrease annually by as much as 20 arcseconds, depending on the star's location.

Other effects come into play also. Earth's axis has another wobble (induced by the gravitational attraction of the Moon) called nutation that has an amplitude of 9.203 arcseconds and a period of 18.6 years. Another effect called aberration (due to the finite speed of light which causes the apparent direction of starlight, arriving at 299,300 km per second to change as the Earth orbits the Sun at a velocity of 30 km per second), amounts to 9 arcseconds. Finally, nearby stars appear to shift slightly due to parallax as Earth circles the Sun. The *apparent positions* that astronomers measure include all of these effects; when they are removed and formalized in a fundamental catalog of stellar positions, the result is called an *astrometric position*.

Fortunately for everyday bread-and-butter astrometry, precession, nutation, aberration, and parallax are almost exactly the same over any small section of the sky, so their effects cancel out; and very few stars are nearby so that parallax almost never becomes an issue.

What poses the dilemma for astrometry is the apparent motion of stars across the celestial sphere—*proper motion*. With a few dramatic exceptions (such as Barnard's Star, which has a proper motion of 10.4 arcseconds per year), most stars

## Section 9.1: Astrometric Catalogs and Coordinates

take decades, centuries, or even millennia to move one second of arc. Unless astronomers have accurate positions separated by several decades, there is no way to know which stars have large proper motions and which do not.

Because they have been measured over a long time span, stars in a fundamental catalog have accurately determined proper motions. The catalog will thus contain five pieces of information for each star: the right ascension, declination, epoch, and annual proper motion in each of the two coordinates. With this information, it is possible to calculate the astrometric position of a star for any desired epoch. The stars in a fundamental catalog thus comprise a *coordinate reference frame* for everything else in the sky.

For the mass production of astrometric data, however, suitable pairs of photographs taken 20 to 50 years apart are not available. Of necessity, astronomers have measured positions of stars in a fundamental catalog on plates taken at some date, solved for the plate constants (as explained later in this chapter), and then computed coordinates of all the other stars on the plates. Even though the coordinates may be given for epoch J2000, the positions of the stars remain what they were when the plate was taken.

### 9.1.2 Astrometric Catalogs and Reference Frames

For many years, the ultimate arbiter of stellar positions was the *FK4*, the *Vierte Fundamental Katalog*, a compilation of some 1,536 star positions derived from precise meridian circle measurements. The absolute accuracy of the *FK4* was about 0.1 arcsecond. The positions of the much larger numbers of stars in later astrometric compilations such as the *Smithsonian Astrophysical Observatory Catalog* (*SAO*), with 259,000 stars, and *Sky Catalog 2000.0*, with 45,000 stars, traced their accuracy directly to the *FK4* or the earlier *FK3*. Unfortunately, these catalogs contain too few stars to be really useful to observers with CCD cameras. Even though the 259,000 stars in the *SAO* sounds like an enormous number, its coverage averages out to 6 stars per square degree. Any field of view significantly smaller than a square degree runs the risk of having too few reference stars for astrometry. In the late 1980s, the *FK4* was superseded by the *FK5*. Astrometric positions in the *FK5* are given for epoch J2000.

The year 1992 brought a boon to astrometry, when the Space Telescope Science Institute issued the *Guide Star Catalog* (*GSC*). Intended for use with the fine-guidance system in the Hubble Space Telescope, the *GSC* lists over 15 million stars down to magnitude 14. The *GSC* was based on digitized photographic plates taken with the Palomar Oschin Schmidt telescope and the UK Schmidt telescope in 1975. The positions of stars on the Schmidt plates were derived with respect to reference stars in the *FK3*, *SAO*, and *Cape Photographic Catalog*. The probable error lies in the range of 0.5 arcsecond, better in some parts of the sky and worse in others. However, the insurmountable problem with the *GSC* is that even though the *coordinates* are given for epoch J2000, the epoch of the stellar *positions* is 1975 or 1982.

As the *Guide Star Catalog* was being released, the European Space Agency

## Chapter 9: Astrometry

cy's Hipparchos satellite (named to honor the ancient Greek astronomer and *not* incidentally an acronym for "HHigh-Precision PARallax COllecting Satellite") was making astrometric history. Free from the distorting effects of Earth's atmosphere on the apparent positions of the stars and the distorting effect of Earth's gravity on its special telescope, Hipparchos measured star positions with unprecedented accuracy. After years of intensive data reduction, the positions of the 118,000 stars in the Hipparchos catalog are known to an accuracy of 0.002 arcsecond, and those in the larger Tycho catalog (1,058,000 stars magnitude B = 10 to 11 or brighter) are reliable to 0.05 arcsecond.

Both the Hipparchos and the Tycho catalogs use a new fundamental reference system adopted by the International Astronomical Union (IAU) in 1995, called the International Celestial Reference System (ICRS). The ICRS finally provides a reference frame that will not change, because it is based on long-baseline radio interferometry of cosmic radio sources that are too distant to show any trace of proper motion. Coordinates in the Hipparchos and Tycho catalogs are given in the ICRS reference frame for J2000, with a 1991.25 epoch of position.

Even these two catalogs do not contain enough stars to serve as references for amateur CCD astrometry, but two catalogs from the United States Naval Observatory do. These are the *USNO-A2.0* and *USNO-SA2.0* astrometric catalogs. Tied to the Hipparchos data, the *USNO-A2.0* provides a dense coverage (with 526,000,000 stars it averages more than 12,000 stars per square degree), and excerpts from this precise astrometric catalog can be downloaded over the Internet.

Superseding these catalogs is the USNO CCD Astrograph Catalog (UCAC). In 1997, the U. S. Naval Observatory built a special CCD-equipped telescope to image the entire sky; by May 2004 all observations for UCAC had been completed, and the release of the final catalog is expected in 2006. A partial release of data from the south celestial pole to +40° became public in 2003, and is called UCAC2.

The stellar positions in UCAC are referenced to Hipparchos data, and are accurate to 0.020 seconds of arc for stars in the 10 to 14 magnitude range. At the limiting magnitude—16th magnitude in red light—the catalog positions have a standard error of 0.070 seconds of arc. Best of all, the UCAC includes proper motions derived from data of various earlier epochs, so star coordinates can be updated to the epoch of observation. Check the USNO website for current availability.

For demanding amateur astrometry, the USNO catalogs are the first choice; however, for routine locating and measuring celestial objects to a bit better than one second of arc, the *Guide Star Catalog* remains entirely adequate.

### 9.2 Astrometric Theory

Astrometric accuracy depends on having a precise reference frame within which the position of celestial objects can be measured. Using a typical amateur's telescope equipped with a CCD camera, observers routinely make measurements to a precision of 0.2 arcsecond or better. The key concept is that astrometry measures the relative location of a star in an image that covers only a small section of the

sky with a precision of 1 part in 5,000 relative to a background of stars whose positions are accurately known.

### 9.2.1 Standard Coordinates

When astronomers speak of “the plane of the sky,” they mean a section of the celestial sphere that is sufficiently small that it can be treated as a plane surface. Standard coordinates take the “plane of the sky” concept literally, defining it as a plane that is tangent to the celestial sphere at a point  $(\alpha_0, \delta_0)$  on the sky. The  $X$ -axis is aligned with right ascension ( $\alpha$ ), the  $Y$ -axis is aligned with declination ( $\delta$ ), and the origin, i.e.,  $(0, 0)$ , lies at the point of tangency. The coordinates  $(X, Y)$  are expressed in terms of the unit radius of the celestial sphere.

It is helpful to think of this sphere as a transparent globe with a tiny bright lamp at its center, and the plane containing standard coordinates as a sheet of stiff cardboard touching the outside of the globe at some point. Star symbols on the globe cast (or “project”) shadows on the cardboard, and the locations of the shadows are the positions of the stars in standard coordinates. This geometry is called a central, or gnomonic, projection.

It is straightforward to compute the locations of stars in standard coordinates located on the plane tangent to the celestial sphere at the point  $(\alpha_0, \delta_0)$ . Given the right ascension and the declination of a star, its standard coordinates are:

$$X = \frac{\cos \delta \sin(\alpha - \alpha_0)}{\cos \delta_0 \cos \delta \cos(\alpha - \alpha_0) + \sin \delta_0 \sin \delta} \quad (\text{Equ. 9.1})$$

$$Y = \frac{\sin \delta_0 \cos \delta \cos(\alpha - \alpha_0) - \cos \delta_0 \sin \delta}{\cos \delta_0 \cos \delta \cos(\alpha - \alpha_0) + \sin \delta_0 \sin \delta}. \quad (\text{Equ. 9.2})$$

Given the standard  $(X, Y)$  coordinates for a star, the right ascension and declination are computed from:

$$\alpha = \alpha_0 + \arctan\left(\frac{X}{\cos \delta_0 - Y \sin \delta_0}\right) \quad (\text{Equ. 9.3})$$

$$\delta = \arcsin\left(\frac{\sin \delta_0 + Y \cos \delta_0}{\sqrt{1 + X^2 + Y^2}}\right). \quad (\text{Equ. 9.4})$$

As we shall see, standard coordinates are useful because they mimic the formation of images in a telescope. When an ideal telescope forms an image of the sky, parallel light from sources on the celestial sphere passes through its optics and comes to focus *not* on a spherical surface behind the lens, but instead on the flat surface of a photographic plate or a CCD chip (see Figure 9.1).

The image of a star with an angular distance  $\vartheta$  from the telescope’s optical axis forms an image at a linear distance  $r$  from that axis:

$$r = F \tan \vartheta, \quad (\text{Equ. 9.5})$$

## Chapter 9: Astrometry

where  $F$  is the focal length of the telescope objective. Thus, star images are projected from locations on the celestial sphere to locations on a plane tangent to the celestial sphere at the aim-point of the telescope; i.e., as a gnomonic projection. Because the angles involved are small (typically, the field of view in an astrometric image is less than  $1^\circ$ ), the small-angle approximation  $\tan \vartheta \approx \vartheta$  is valid, where  $\vartheta$  is measured in radians. For example, at an angle of 0.01 radian, the tangent equals 0.010003333. Thus, we can simplify the relationship to:

$$r \approx F\vartheta, \quad (\text{Equ. 9.6})$$

stipulating that  $\vartheta$  be given in radian measure. Virtually all telescopes used for astronomy approach the ideal so closely that deviations from this relationship are negligible over a  $1^\circ$  full field.

### 9.2.2 Plate Coordinates

When an astrometric image is taken, the astronomer does not know the exact coordinates of the center of the image  $(\alpha_0, \delta_0)$ . After the plate is developed or the CCD read out to a computer, the locations of stars on the image are measured in  $(x, y)$  coordinates. Photographic images have customarily been measured using a precision measuring engine, and the location of each star image recorded in millimeters. With CCD camera, however, the centroid of each star image is measured in units of pixels and, from their known dimensions, converted to millimeters.

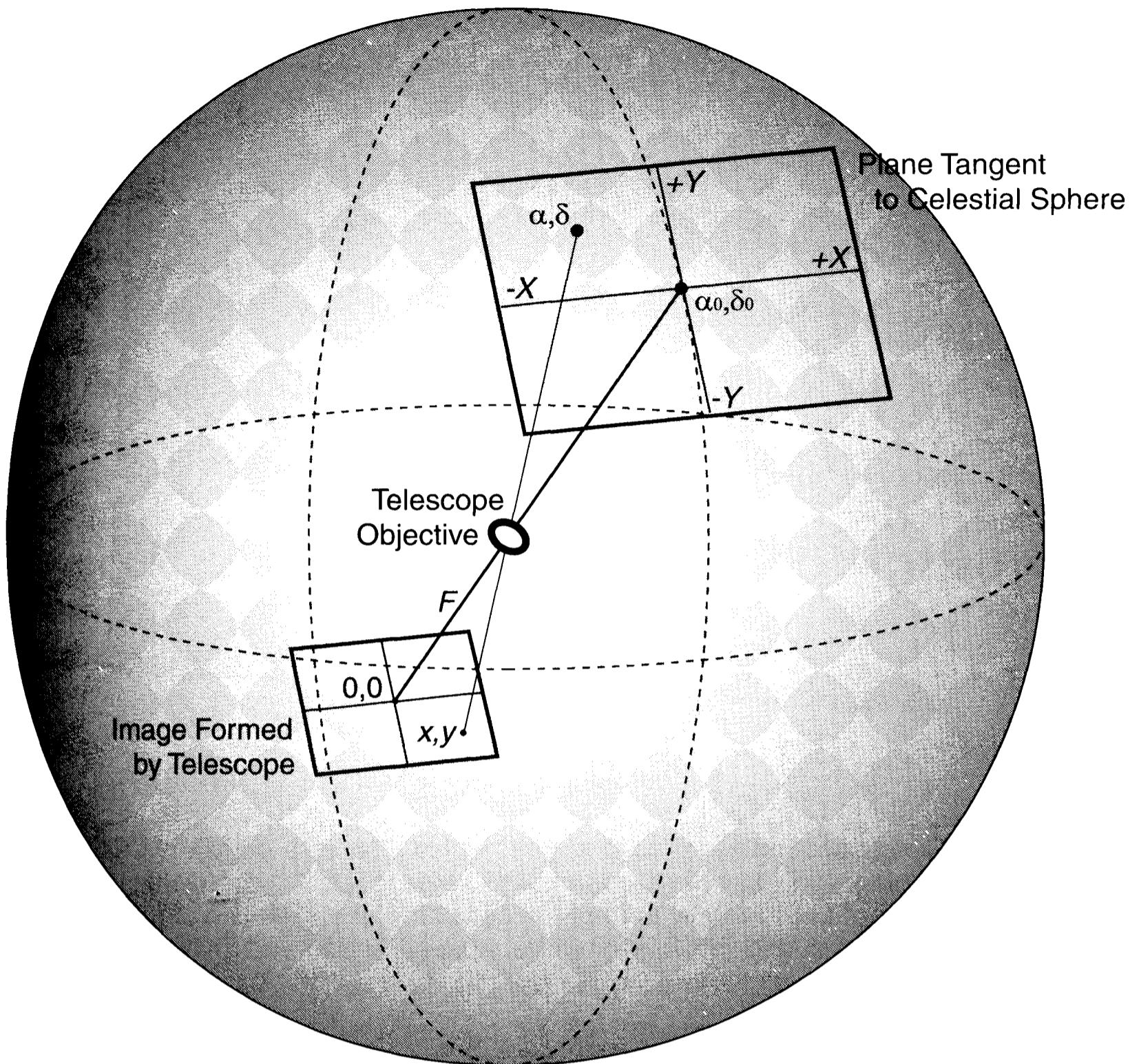
In an ideal situation, the  $x$  and  $y$  axes used in measuring the image would be perfectly aligned with its right ascension and declination axes, and the coordinates of the exact center of the image  $(\alpha_0, \delta_0)$  would be known. If they were known, then the measured  $(x, y)$  coordinates could be converted into standard coordinates by dividing by the focal length of the telescope:

$$\begin{aligned} X &= x/F \\ Y &= y/F. \end{aligned} \quad (\text{Equ. 9.7})$$

In real life, however,  $(\alpha_0, \delta_0)$  is inevitably somewhat off the center of the image, and the  $x$  and  $y$  axes of the detector will be rotated through some angle (see Figure 9.2). If the departure from the aim-point is  $(x_{\text{offset}}, y_{\text{offset}})$  and the detector is rotated through an angle  $\rho$ , the relationships between plate coordinates and standard coordinates become:

$$\begin{aligned} X &= \frac{\cos \rho}{F} x - \frac{\sin \rho}{F} y - \frac{x_{\text{offset}}}{F} \\ Y &= \frac{\sin \rho}{F} x + \frac{\cos \rho}{F} y - \frac{y_{\text{offset}}}{F}. \end{aligned} \quad (\text{Equ. 9.8})$$

In addition to displacement and rotation, the detector may be slightly tilted relative to the incoming light; the manufacturer's figures for the pixel dimensions might be slightly inaccurate at the CCD's operating temperature; and in the case of scanned photographs, the axes may not be perfectly orthogonal and equally



**Figure 9.1** Telescopes project a section of the celestial sphere onto a flat photographic plate or a CCD chip. Shown are standard ( $X, Y$ ) coordinates on a plane tangent to the celestial sphere, and ( $x, y$ ) coordinates measured from a photographic plate or CCD located at the focus of the telescope.

scaled. These errors, provided that they are small, can be included as further terms involving  $x, y$ , and offset.

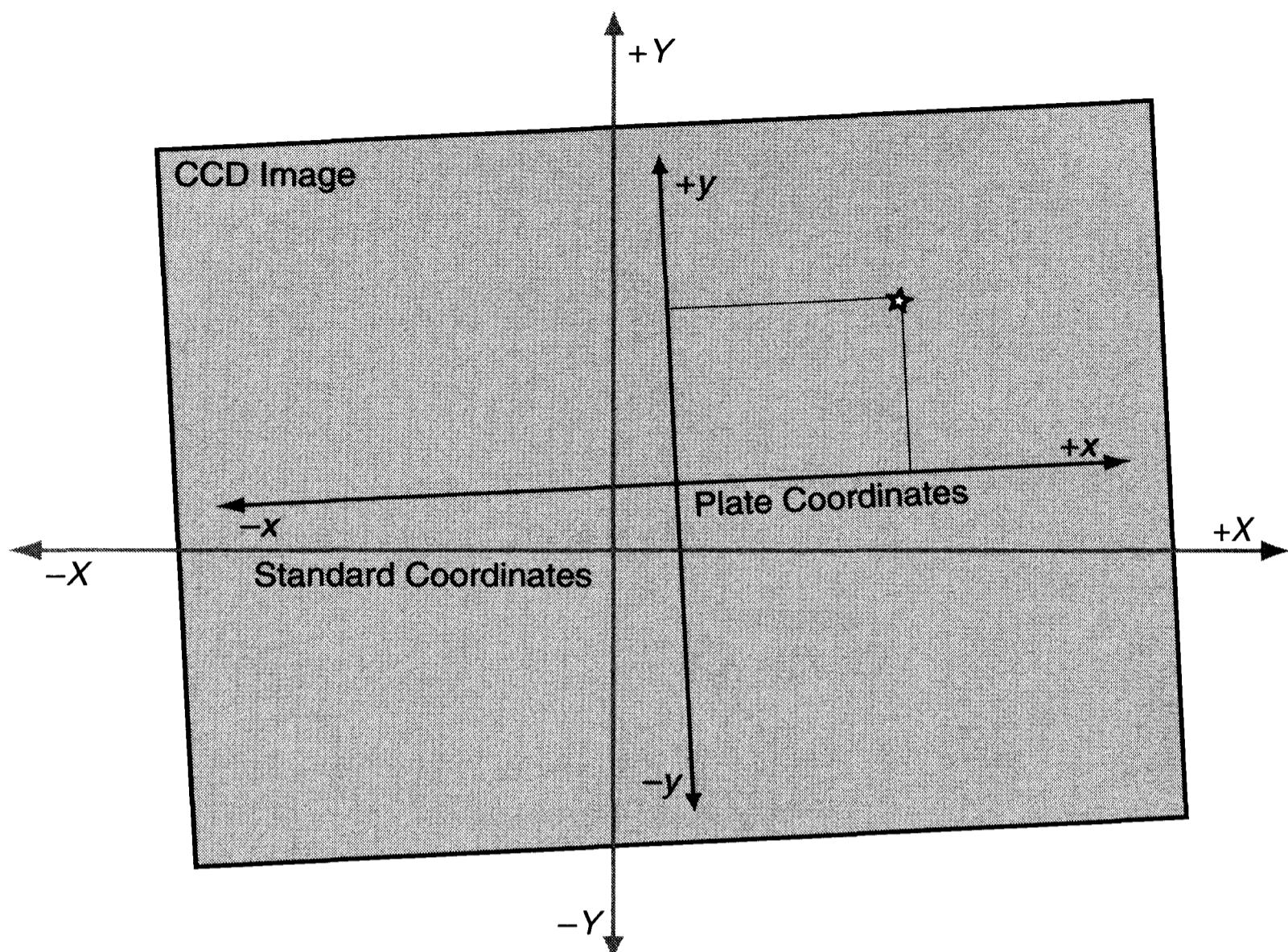
However, rather than try to take into account every peculiarity of the telescope and optics, we simply rewrite the relationship between the two coordinate systems as a general linear transformation:

$$\begin{aligned} X &= ax + by + c \\ Y &= dx + ey + f. \end{aligned} \quad (\text{Equ. 9.9})$$

The terms  $a, b, c, d, e$ , and  $f$  are called plate constants. The plate constants can be determined empirically from the image itself. The trick is to measure the  $x, y$  locations of three or more reference stars and to compute the standard coordinates of these stars, and then solve the resulting linear equations for the plate constants.

The equations in  $X$  for three stars are:

## Chapter 9: Astrometry



**Figure 9.2** Plate coordinates refer to the  $(x, y)$  location of a star in a CCD image or photograph—standard coordinates refer to  $(X, Y)$  coordinates on a plane tangent to the sky at the right ascension and declination of the plate center. The two coordinate systems are usually rotated and shifted relative to one another.

$$\begin{aligned} X_1 &= ax_1 + by_1 + c \\ X_2 &= ax_2 + by_2 + c \\ X_3 &= ax_3 + by_3 + c. \end{aligned} \quad (\text{Equ. 9.10})$$

This gives us three independent equations with three unknowns,  $a$ ,  $b$ , and  $c$ , which can be solved using standard algebraic methods.

The equations in  $Y$  for three stars are:

$$\begin{aligned} Y_1 &= dx_1 + ey_1 + f \\ Y_2 &= dx_2 + ey_2 + f \\ Y_3 &= dx_3 + ey_3 + f. \end{aligned} \quad (\text{Equ. 9.11})$$

Again, by measuring three reference stars, we have three independent equations with three unknowns,  $d$ ,  $e$ , and  $f$ , to solve. In this formulation, the plate constants are approximately equal to or smaller than  $1/F$ .

Interestingly, it is not necessary to know the focal length of the telescope with any degree of precision. The small-angle approximation  $\tan \vartheta \approx \vartheta$  implies that  $\tan n\vartheta \approx n\vartheta$ ; so if the focal length is, for example, too long by a factor  $n$ , the plate coordinates will remain proportionately correct even though they are small by a factor of  $n$ . This is, however, an approximation. In computer-assisted astro-

metric reductions, the focal length can be determined precisely from the true angular distances between the reference stars and the known distance between the same stars on the plate or CCD; run the solution again, this time using the accurate focal length.

### 9.2.3 Improved Plate Constants

Only three reference stars are necessary to solve for plate constants, but more reference stars may be available on the image. To take advantage of the potential for a more accurate set of plate constants from the additional stars, we can execute a least-squares solution to produce the most probable solution to an overdetermined set of equations.

We assume that each of the equations contains a residual: that is, if we were to place all of the terms on the right hand side of the equal sign and evaluate the equation, the result would not be zero—there would instead be a small remaining difference. For example, the equation that sets up this condition for  $i$ th star in a set of reference stars is:

$$\begin{aligned} X_i - x_i/F &= ax_i + by_i + c \\ Y_i - y_i/F &= dx_i + ey_i + f, \end{aligned} \tag{Equ. 9.12}$$

in which the residual is  $X_i - x_i/F$ ; i.e., the difference between the measured place of the star and its computed place. (Note that each  $x_i$  is converted from millimeter to radian measure by dividing by the focal length.) Assuming that the errors are distributed normally (i.e., that a plot of the errors would form a bell-shaped curve), the best solution occurs when the sum of the squares of the residuals is smallest. This condition gives the method of least squares its name.

To find the minimum value of the sum of the squares of the residuals, the equations of condition are differentiated with respect to  $x$ , producing three so-called normal equations in which  $a$ ,  $b$ , and  $c$ , and  $d$ ,  $e$ , and  $f$  are the unknowns, and their coefficients are terms in  $x$  and  $y$ . Since the normal equations are linear, their solution is straightforward.

After solving for the plate constants, the residuals in each axis are computed for each reference star, converted into arcseconds, and displayed. The quadratic sum of the residuals is also computed and displayed as the standard deviation expected in the coordinates of the target object. If the residuals and standard deviation are significantly larger than the standard deviation for reference star positions, the current position of one or more of the reference stars is suspect.

### 9.2.4 Solving for Position

Once the plate constants have been determined, either from a simple three-star solution or from a least-squares fit involving four or more reference stars, the position of the target object in the image can be determined from its  $(x, y)$  plate coordinates. With all terms on the right side of both equations now known,  $X$  and  $Y$  are readily determined:

## Chapter 9: Astrometry

$$\begin{aligned} X &= ax + by + c \\ Y &= dx + ey + f. \end{aligned} \quad (\text{Equ. 9.13})$$

The final step is to convert the standard coordinates of the target back to right ascension and declination. With both  $(\alpha_0, \delta_0)$  and  $(X, Y)$  now known, its coordinates are computed from:

$$\delta = \arcsin\left(\frac{\sin\delta_0 + Y\cos\delta_0}{\sqrt{1 + X^2 + Y^2}}\right) \quad (\text{Equ. 9.14})$$

$$\alpha = \alpha_0 + \arctan\left(\frac{X}{\cos\delta_0 - Y\sin\delta_0}\right). \quad (\text{Equ. 9.15})$$

Results are normally carried about one-half decimal place beyond the computed residuals so as neither to lose potential information nor to overstate the accuracy of the result; in amateur astrometry, this usually means 0.1 arcsecond in declination and 0.01 second in right ascension.

## 9.3 Practical Astrometry

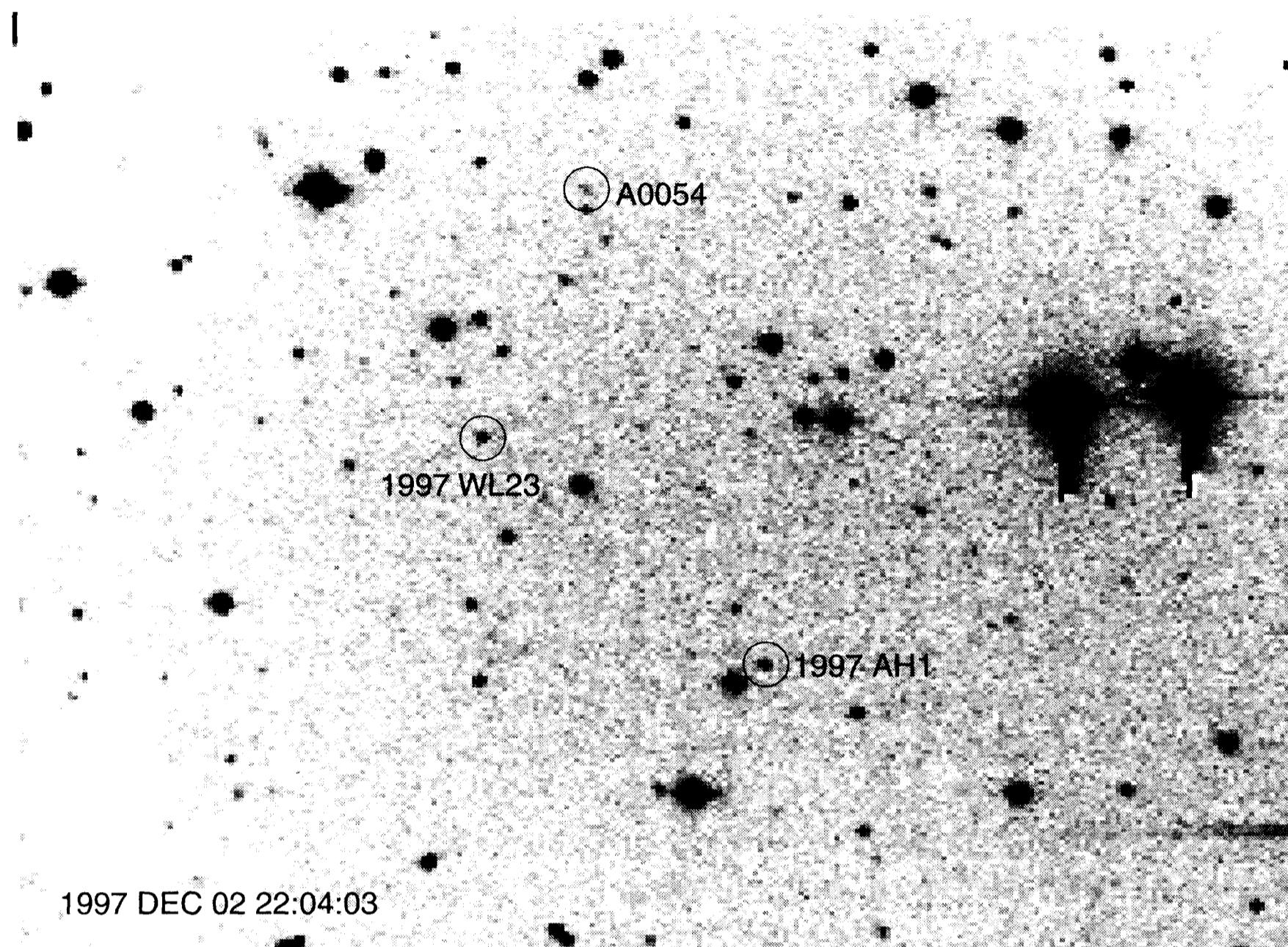
In the preceding sections, we have seen that astrometry is a four-step process: first measuring  $(x, y)$  locations for each reference star and collecting the corresponding right ascensions and declinations; second, computing plate constants for the image; third, measuring the  $(x, y)$  positions of target objects; and last, computing the coordinates of the target objects.

Software makes the computationally intensive second and fourth steps nearly invisible to the user, making good astrometry a two-step process of identifying a set of reference stars and then of identifying the target objects. However, every astrometry session begins with suitable images.

### 9.3.1 CCD Images for Astrometry

You can make astrometric measurements from virtually any CCD image, but the best results—the smallest residuals—come from those made with adequate exposure times, careful calibration, and good tracking. Bear in mind, however, that high precision is not always necessary. If you are following up on a newly-discovered asteroid, any reasonably good position is better than none—so positions with residuals of 2 and even 3 arcseconds are useful in recapturing the object at a later apparition.

**Accurate Timing.** For top-notch astrometry of comet and asteroid positions, where determining the orbit of the object is top priority, you must report the time of the mid-exposure to within one second, if possible. CCD software usually notes either the beginning or the end of integration in a data file, so you will need to determine which, and then either add or subtract half of the integration to obtain the time of mid-exposure. To insure accurate timing, set the clock in your comput-



**Figure 9.3** On November 30, 1997 while making astrometric images of 1997 AH1, Brian Manning discovered 1997 WL23. Pursuing that find two nights later, he captured yet another asteroid, A0054, which (unfortunately) he did not notice until April 1998. Five-minute integration; 10-inch reflector; Cookbook CCD.

er before an astrometric session.

**Correct Exposures.** For stationary target objects, integrations can be as long as necessary to record a clear image, and may consist of multiple integrations that are track-and-stacked in software. However, images of the brightest reference stars must lie in the linear portion of the CCD's response curve. For moving objects, the practical upper limit on the integration time depends on how rapidly they are moving. For top-notch results, the object should not trail more than about half of the full-width half-maximum of the image object; that is, the trail should not exceed one second of arc. In a pinch, however, astrometry from trailed target images usually gives acceptable results.

**Calibration.** Astrometric images must be dark-subtracted to eliminate hot pixels that might distort star centroids, but for images that are free of vignetting, flat-fielding is usually unnecessary because the centroid algorithm operates over a small region around each star image and is not strongly affected by the sky background. Images with vignetting should not be used for high-precision astrometry because vignetting may slightly distort the stars' intensity profiles and degrade the accuracy of the centroids.

**High Elevation.** Differential refraction by Earth's atmosphere shifts the apparent places of stars, reducing the accuracy of the plate constants determined

## Chapter 9: Astrometry

from the reference stars. If the field is above 45° elevation, the effects of differential refraction become negligible. Because astrometry has built-in quality control in the form of the residuals, however, a practical approach to reporting the positions of low-sky objects is to take the image, make the measurements, and report the residuals with the derived position.

**Proper Sampling.** For accurate astrometry, you need sharply focused star images that have a full-width half-maximum of at least two pixels. Very small images do not produce high-quality centroids. Depending on the telescope optics, the atmospheric seeing, and pixel size of the CCD camera, this implies a focal length between 1 and 3 meters (40 to 120 inches). However, acceptable astrometric images may be obtained with any optical system from a 200-mm telephoto lens to a Cassegrain telescope with a 10-meter focal length, providing the star images are well sampled.

**Take Images with North at Top.** Technically speaking, the orientation of the image does not matter—but it is much easier to identify reference stars if they are oriented the same way that most finder maps and charts are. By allowing a bright star to trail, the camera can be set within a few degrees of north at the top in two or three minutes at the beginning of the observing session.

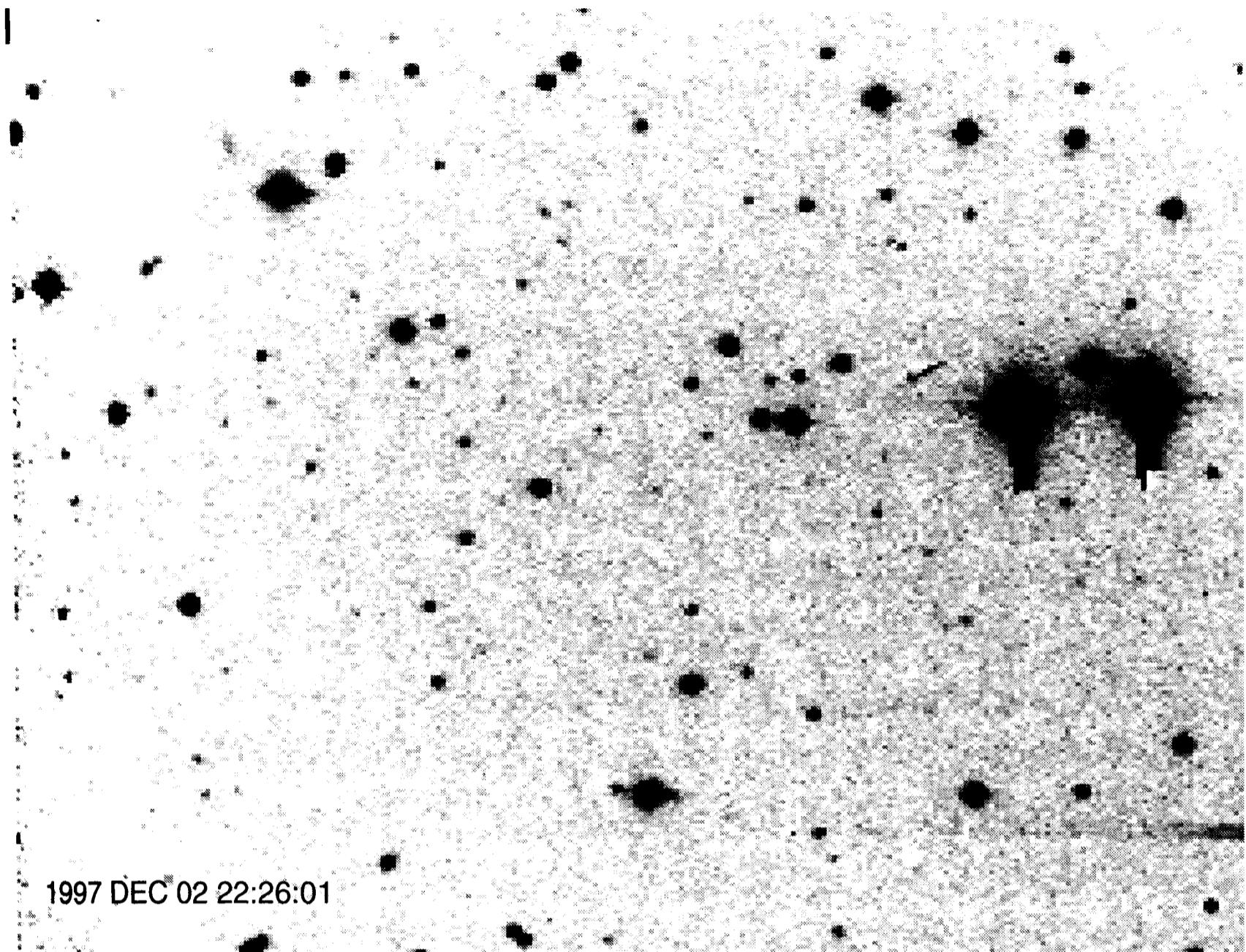
**Enough Reference Stars.** At least three reference stars are required for astrometry—between six and ten are desirable. This places significant constraints on the size of the field of view, the limiting magnitude of the image, and the reference star catalog that you use. A quick check of the field of view using star-chart software (such as *MegaStar*) will show how many potential reference stars will be available. If the field of view is very small and includes only one or two reference stars, either a shorter focal length telescope, a larger CCD camera, or an astrometric catalog with more reference stars is necessary.

**Calibration Only, Please!** The Golden Rule of CCD Astrometry is “calibrate and then analyze.” Brightness scaling, sharpening, and resampling seriously degrade the value of an image for astrometry. There is no need to process any astrometric or photometric images beyond calibration—the image is ready for analysis immediately after dark current subtraction and flat-field correction.

### 9.3.2 Scanned Photographs for Astrometry

It is possible to carry out astrometric measurements on digitally scanned photographs, although the accuracy of the positions obtained is often limited by the quality of the scan. Three problems plague scanned photographs: the initial non-linear image storage in film, the danger of saturated data in scanning, and poor scanner characteristics.

In digital images from CCDs, the pixel values are proportional to the intensity of the light that fell on the CCD. This is not true of scanned photographic images; intensity is lost at both the low end and the high end of the intensity scale. At the low end, the outer parts of the star images are lost in the “toe” portion of the photographic response curve; and at the high end, the density of the aggregated



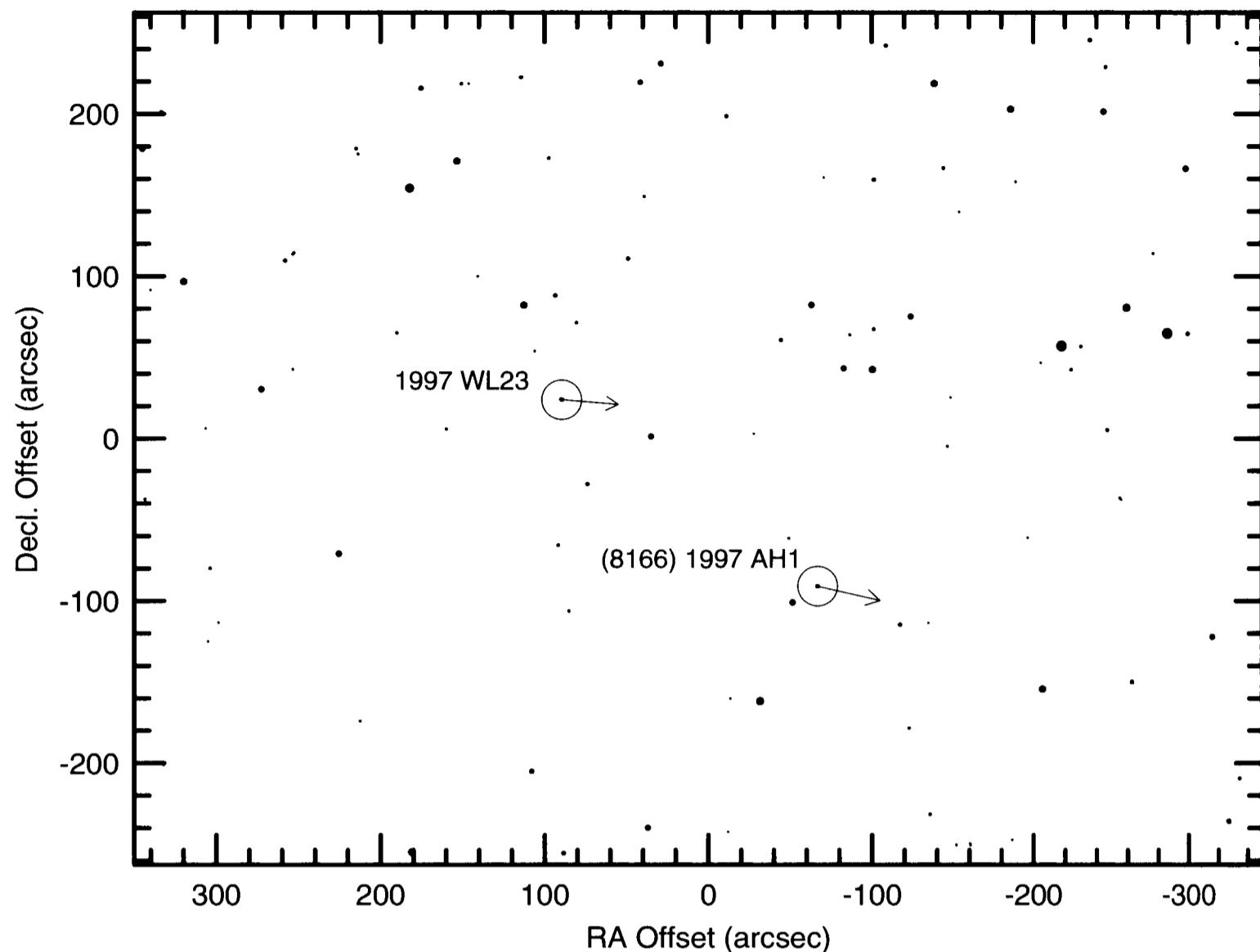
**Figure 9.4** Brian Manning made this image 22 minutes after the one in Figure 9.3. Compare carefully: you will see that all three asteroids have moved noticeably. The field is in Taurus at 4h 00m 19s; +20° 20' 20''. Five-minute integration; 10-inch reflector; Cookbook CCD.

silver grains hides the peak intensity in the center of the star image. Even with a high-quality scan, star centroids derived from photographs depend on the shoulders of the star image rather than its bright core.

The second problem is that few scanners capture the full dynamic range between the sky and well-exposed stars images. The internal algorithms that determine how the data will be saved in the scanner software allow star images to become saturated at pixel values of 255, 1023, or 4095 in order to preserve the sky background. The centroid routine must therefore operate on a star image consisting of a rough circle of saturated pixels surrounded by an apron of pixels representing only the outer edge of the star image.

Finally, the  $x$  and  $y$  axes in the scanner may depart slightly from the orthogonal (i.e., they may not be perfectly perpendicular) and not be at the same scale. In themselves, these are not problems because the plate constants compensate for non-orthogonality and differing scales in the two axes—but if the scales vary and the angle between the axes varies across the image, the astrometric solution will be degraded.

Despite these problems, astrometric measurements work reasonably well on scanned photographs. In a test, Kodak PhotoCD scans from 35-mm film yielded residuals of 5 arcseconds r.m.s., and desktop scanner scans from color enlarge-



**Figure 9.5** Lowell Observatory's web site can generate a star map showing the location of all known asteroids present in a given field at the date and time that you specify. This chart shows two of the asteroids in Figures 9.3 and 9.4; the third object has not been recovered.

ments of the same negatives yielded residuals of 10 arcseconds. For comparison, a Cookbook CCD camera used on the same telescope yielded residuals under one second of arc.

To measure scanned photographs astrometrically, it is necessary to have a fairly good idea of the effective pixel size. With scans from full-frame negatives, compute the effective pixel size from:

$$\text{pixel size} = \frac{\text{image width in millimeters}}{\text{image width in pixels}} . \quad (\text{Equ. 9.16})$$

You may need to guess at the exact width of the scanned region; for the sake of argument, assume that a PhotoCD scan covers an area 25.4 mm wide. If you use the Basex4 image for astrometry, the image width is 1024 pixels, so:

$$\text{pixel size} = \frac{25.4\text{mm}}{1024} = 24.8 \text{ microns} . \quad (\text{Equ. 9.17})$$

Similar logic applies to scanned photo enlargements, except that the enlargement factor must be measured or estimated. The formula is:

$$\text{pixel size} = \frac{\text{image width in millimeters}}{(\text{enlargement}) \times (\text{image width in pixels})} . \quad (\text{Equ. 9.18})$$

Suppose that you scan a standard 4x6-inch color enlargement from a 35-mm negative. To a reasonable approximation, the enlargement is 4 times. You measure the actual image width as 3.95 inches; and after scanning the print at 300 d.p.i. on a desktop scanner, the image is 1240 pixels wide. The effective pixel size is:

$$\text{pixel size} = \frac{3.95 \times 25.4}{4 \times 1240} = 20.2 \text{ microns.} \quad (\text{Equ. 9.19})$$

Given any reasonable value for the pixel size, the astrometric solution will compensate with a proportionate error in the focal length of the telescope or lens, and the coordinates that you measure from the image will be surprisingly good.

### 9.3.3 Making Astrometric Measurements

Although the user interface differs from one software application to the next, all astrometric software has the same basic needs. Once you have verified that the image actually contains the target object, the first step is to designate a set of reference stars. The designation process can be an on-screen pointer moved by cursor keys, a mouse-driven cross hair, a press of the enter key, or a click of a mouse button. In all cases, the user instructs the software which stars should make good reference stars. The software determines the centroid of each star image and places these in a database along with the astrometric coordinates of the star.

- **Tip:** *AIP4Win makes identifying and selecting reference stars virtually foolproof. Using the Guide Star Catalog CD-ROMs (from NASA) or MegaStar's GSC Stars function, generate a text file containing the stars in the field. Activate the Astrometry Tool, load the reference star file, and press the Overlay button. The reference stars appear as circles on the image. Use the move and rotate buttons until the reference star circles line up with the stars in the image.*

Select six to ten reference stars surrounding the target object. These should be moderately bright, single stars with clean, round images. Do not select stars that show blooming or diffraction spikes.

- **Tip:** *In AIP4Win, as soon as you have selected three reference stars, press the Update Focal Length button. This performs a preliminary astrometric solution, computes the focal length and image rotation, resulting in much more accurate positioning of the reference star overlay.*

After selecting reference stars, locate the target or targets. Unlike reference stars which you can choose, you must take target objects where you find them. If one is blended with, or too near, a star, the centroid routine may not be able to separate the target from the interfering image.

- **Tip:** *AIP4Win asks you to name each target object when you select it. As you move the cursor over the image, its right ascension and declination are displayed continuously. For “quick-and-dirty” astrometry, this coordinate readout may be all you need. To iden-*

## Chapter 9: Astrometry

*tify objects such as variable stars, compare the coordinate read-out in the Astrometry Tool with the listed position of the object.*

After you have selected the reference stars, compute the coordinates of the target objects and print a report listing the results. Examine it carefully. Each reference star is assigned a residual that indicates how well it fits into the overall solution of plate constants. If the residuals for one star are especially large, it may have a large proper motion and have changed its position since the epoch of observation. Eliminate this one as a reference star and run another solution for plate constants.

With a good CCD image and the positions from the *Guide Star Catalog*, the root-mean-square residuals are normally less than one second of arc in both axes, and quite often less than 0.5 arcseconds. However, in fields where nebulosity interfered with the scan used to create the *GSC*, errors in the reference star coordinates may lead to errors of several arcseconds. Scans from the region around the Orion Nebula seem to have been especially bad, for example, leading to residuals of two to three arcseconds.

## 9.4 Applied Astrometry

Astrometry has two primary applications: the first, determining the position of newly discovered stationary objects such as novae, supernovae, and variable stars to aid other observers in locating them; and the second, measuring the positions of moving objects such as comets and asteroids for the purpose of determining the orbit of the new-found object. In addition, astrometry is an extremely useful tool for relatively mundane tasks such as identifying which is the variable star in an image containing hundreds of star images, locating faint comets, and determining the precise scale and orientation of an image. Of course, astrometry also lends itself to educational demonstrations such as measuring the proper motion of Barnard's Star, measuring the parallax of nearby stars, and determining the orbit of an asteroid.

### 9.4.1 Astrometry of Newly-Discovered Objects

Some observers want to discover a supernova, while others search for new variable stars. However, a problem suddenly arises when the search is successful—how best to describe the location of the new object? Astrometry provides the means to place the object in a reference frame that all astronomers use.

The initial step is to obtain several images of the new object, either the discovery image or confirmation images taken as soon as the object is recognized as something unusual. The image must cover enough of the surrounding sky to include at least three, and preferably six to ten, reference stars.

Calibrate your images, but do not otherwise process them. Image enhancements have the potential to destroy the astrometric value of an image. If necessary, you can stack images to improve the signal-to-noise ratio, but do not apply any

intermediate enhancement procedures while you are stacking.

After loading the calibrated image into your favorite software, select the reference stars. Reference star data should come from a well-known astrometric catalog such as the *Guide Star Catalog* or the *USNO-SA2.0*.

- **Tip:** *AIP4Win provides an overlay to make identification of the reference stars fast and foolproof. After toggling on the overlay, shift and align it to match the star images. As you click on each star, its (x, y) position, coordinates, and magnitude are loaded into internal arrays.*

After selecting the reference stars, the plate constants take a few milliseconds to compute. Select your target object—the new supernova, variable star, or whatever—and then compute its coordinates.

- **Tip:** *The Astrometry Tool in AIP4Win includes a quick-and-dirty photometry routine that uses magnitudes from the reference stars. Unfortunately, the brightness listed in astrometric catalogs may be in error by several tenths of a magnitude; any magnitude based on astrometric reference stars must be treated as an estimate.*

When you report the position of the object, you include in the report the coordinate epoch (which is the epoch of the coordinates in the reference star catalog) and the astrometric residuals. The residuals give potential users some idea of the quality of the astrometric determination that you have made.

### 9.4.2 Astrometry of Asteroids and Comets

With the advent of CCD imaging, asteroid discovery became a growth industry in amateur astronomy. The imaging techniques are simple, and the privilege of naming the asteroid when its orbit is confirmed is a potent incentive. Since virtually all main-belt asteroids stay within  $\pm 25^\circ$  of the ecliptic, and near opposition display retrograde (east-to-west) motion of about 30 arcseconds per hour, asteroids are easy to spot on pairs of images taken an hour apart.

An effective discovery technique is to shoot slightly overlapping images covering a strip of sky near the ecliptic about an hour before the areas cross the meridian, and then to reshoot the same areas an hour later as they culminate. “Blinking” the image pairs reveals moving objects and gives a rough approximation of their rate of motion, and comparison with predicted positions for known asteroids reveals new ones.

The key to claiming an asteroid find is following the object for the next three to four months to gather enough positions to compute a preliminary orbit so that the new object can be recovered in the morning sky after conjunction. Even poor observations are crucial at this stage; and the more you gather, the better the orbit determination. When the object reappears in the morning sky, if you succeed in recapturing and following it for another six months, you should have enough data for a good orbit solution. It then becomes your privilege to name the object.

## Chapter 9: Astrometry

Asteroid astrometry is tricky because the objects are moving. If your integration times are too short, you cannot record faint asteroids (necessary because most of the bright ones have already been discovered), but if you use long integration times, the images are trailed and faint objects cannot be seen. For orbit determination, you must record the time of mid-exposure to the nearest second of time, if possible.

Comet astrometry differs from asteroid astrometry in two ways: the objects are diffuse, and they are often located low in the twilight sky. CCD cameras usually have the dynamic range needed to cope with a fuzzy object against a bright background. Images taken near the horizon may suffer from differential refraction and show large residuals.

Making measurements on asteroid and comet images is essentially the same as it is for stationary objects. Star mapping software such as *MegaStar* makes identifying and locating new reference stars fairly easy as the object moves across the sky. Comet images may pose a problem for the centroid algorithm in some astrometric software programs, but good algorithms successfully locate and solve for the center of the bright nuclear region of the comet.

### 9.4.3 Using Astrometry to Identify Objects

Deep-sky and variable star observers sometimes run into a problem: they know that their object is recorded in an image, but they're not sure just where in the image it is. They don't have a finder chart, only the report of a previously unknown flare star at such-and-such a right ascension and declination. Unfortunately, 16th magnitude variable stars don't have name tags attached—and they blend remarkably well into the background of nameless field stars.

The steps in the identification procedure are very simple: load the image, select three reference stars, solve for the plate constants, and then find the object in your image with coordinates that match the published ones.

- **Tip:** *Once you have solved for plate constants, **AIP4Win** displays the right ascension and declination of the pixel under the cursor. To identify an object, simply move the mouse around until the cursor coordinates match its coordinates.*

The same principle works for making positive identifications between stars in an image and those plotted on a finder chart.

### 9.4.4 Image Scale and Orientation

Astrometry yields several useful by-products, including the exact scale and orientation of an image, and the focal length of the telescope used to take it. These data allow an observer to measure the angular size and orientation of an object directly from its image.

## Section 9.4: Applied Astrometry

```

Target object(s): 1997 AH1
Astrometric Image: AH102DAD.fts
UT date of obs, YYYY MM DD: 1997 12 02
UT time of obs, HH MM SS: 22:04:03
Telescope focal length: 1910.62
X size of pixel (mm): 0.0255
Y size of pixel (mm): 0.0198
Ref star data source: AH102.ref

REFERENCE STARS Catalog epoch: 2000
RAS DEC Mu(RA) Mu(DC) mag X Y S
# hh mm ss.sss +dd mm ss.ss -- arcsec/y -- cat pixels pixels Y
1 04 00 39.150 +21 21 46.20 00.000 00.000 13.70 026.771 101.501 Y
2 04 00 01.800 +21 24 39.40 00.000 00.000 13.75 218.939 031.498 Y
3 04 00 21.980 +21 21 17.20 00.000 00.000 14.12 113.305 120.153 Y
4 04 00 30.540 +21 24 08.40 00.000 00.000 13.99 072.783 037.814 Y
5 04 00 04.630 +21 18 39.20 00.000 00.000 13.66 198.759 198.786 Y
6 04 00 35.740 +21 20 04.50 00.000 00.000 13.49 042.325 149.999 Y
7 04 00 10.490 +21 22 32.00 00.000 00.000 14.46 172.687 088.499 Y
8 04 00 14.910 +21 22 38.90 00.000 00.000 13.75 150.514 084.287 Y

TARGET OBJECT(S)
X Y mag RAS DEC S
-- object name -- pixels pixels CCD hh mm ss.sss +dd mm ss.ss Y
1 1997 AH1 143.643 168.991 13.47 04 00 15.694 +21 19 36.36 Y
2 1997 WL23 093.155 108.174 16.76 04 00 26.051 +21 21 40.31 Y
3 A0054 114.022 047.391 16.99 04 00 22.365 +21 23 52.73 Y

ASTROMETRIC RESIDUALS
RArms DECrms
(arcsec) (arcsec)
000.368 000.401

Plate center, RAS: 04 00 19.500
Plate center, DEC: +21 21 16.90
X of plate center: 126.00
Y of plate center: 121.00
PA, image +Y axis: 357.31

```

**Figure 9.6** This report summarizes the process used to derive the positions for the three asteroids seen on Brian Manning's images in Figures 9.3 and 9.4. In a complete report, the date, time, epoch, pixel size, and reference stars used are all shown. For this image, the residuals were only 0.4 arcseconds.

- **Tip:** If you know the orientation and focal length of an image, you can use **AIP4Win Distance Tool** to measure the precise angular separation and position angle between any two points on that image.

Few observers know the actual focal length of their telescopes, yet this parameter is an easily computed by-product of astrometry. Given a reasonably well-separated set of reference stars, a single measurement of the focal length is good to about 0.1%.

### 9.4.5 Astrometry in Education

There is virtually no limit to educational projects based on astrometry—projects that would have been either very difficult or impossible for amateur astronomers and educators before the advent of CCD imaging. What CCD images bring to astrometry is high sensitivity, star images that yield accurate centroids, and software

## Chapter 9: Astrometry

programs that make the reduction of data fast and easy.

One feature of astrometric projects that makes them especially valuable in the educational setting is the analysis of observational errors. The parallaxes of nearby stars, for example, are comparable to the errors of measurement, so that only by making careful measurements, reducing data intelligently, and understanding sources of observational error can an individual or class expect to obtain credible results.

**Proper Motion of Stars.** There are thousands of stars whose annual proper motion exceeds 0.2 of a second of arc—which is just about the limit that an observer with an 8-inch  $f/10$  telescope could detect using a given set of reference stars. The greatest known stellar proper motion is that of Barnard’s Star in Ophiuchus, a red dwarf that moves 10.34 arcseconds per year. A careful observer could (in theory) detect the motion of this object over the course of one week!

In many cases, stars that are near enough to Earth to show appreciable proper motion will also show parallax, substantially enhancing the educational value and interest in making the observations and reducing the data.

A proper motion observing program would consist of imaging a chosen star every couple weeks for several years. When its astrometric positions were plotted, they would describe a looping curve on the sky, a combination of the parallactic oval and the linear motion of the star. The *Gliese Catalog of Nearby Stars* lists some 3800 potential subjects for such study.

**Stellar Parallax.** The German astronomer Friedrich Wilhelm Bessel was the first to publish a widely accepted distance to a star (61 Cygni) in the year 1838. Bessel used trigonometric parallax to measure the apparent displacement of the nearby star against the background of more distant ones. Parallax shifts are small—only 0.7 arcsecond for  $\alpha$  Centauri, the nearest star. If we accept the limit of 0.2 arcsecond for a single measurement made with a CCD camera, then it should be possible for an observer with an 8-inch  $f/10$  telescope to measure the parallax of any star closer than about 15 light years.

The most intriguing aspect of this project would be developing effective techniques to reduce observational error. Such techniques might include using image-motion reduction to make the smallest possible star images, shooting multiple images at each observing session and stacking them to reduce noise and improve the accuracy of centroids, and working at large image scales.

The methodology of the project consists of making images of the star as Earth moves around its orbit, with the greatest number made when the Earth-Sun line is perpendicular to the Sun-star line, but with enough images between the extremes to fill in the oval path described by the parallax star against the background of distant ones.

**Determination of Orbits.** There has probably always been a small band of amateur astronomers interested in the determination of the orbits of comets and asteroids. In the past, however, good positions were hard to come by. With the advent of CCD astrometry, a single observer can track and shoot images of an aster-

## Section 9.4: Applied Astrometry

oid for three to six months, measure its positions on the images, and compute an orbit. To add spice to the project, the asteroid (or comet) could be a newly discovered object for which no previous orbit existed.

The project would consist of making images over an appreciable arc of the object's orbit. The astrometric positions would then become feedstock for an orbit determination program to make a least-squares fit to the best orbit possible, and they would be used to predict future positions for the object. As the measured orbital arc becomes longer, the quality of the solution improves, until eventually you have produced a definitive orbit.

## Chapter 9: Astrometry

---

# 10 Photometry

---

For amateurs who want to make real contributions to astronomy, few areas offer greater opportunities than photometry. Photometric measurements can pinpoint the exchange of mass between distant binary stars, reveal the tumbling of asteroids, or track the decline in the brightness of a supernova.

Photometry is the measurement of the changing brightness of celestial objects over time. The advent of CCD imaging has made photometry easier and more practical than ever before for both professional and amateur observers; not only because the CCD is both sensitive and highly linear, but also because it captures a two-dimensional “virtual sky” for careful analysis at a later date.

Once you have made a set of observations, photometric measurements of the images are fairly straightforward. Lew Cook, an amateur astronomer and photometrist, summarized his observing philosophy this way, “Nighttime is for observing. Daytime is for data processing. Cloudy nights are for sleeping, going to the movies, and taking your wife to dinner. I shoot Friday night, reduce data and email light curves Saturday afternoon, shoot Saturday evening, reduce data and email light curves Sunday. You can make discoveries that way!”

This chapter explains the methods and practices of modern CCD photometry in enough detail to get you interested in trying a few simple projects. Perhaps you will discover the great satisfaction that comes from measuring those subtle changes in starlight that tell us what is going on in the cosmos.

## 10.1 Magnitudes: How Bright Is This Star?

The ancient Greeks divided stars into six classes by *magnitude*, literally by their size. Between 141 and 127 B.C., the Greek astronomer Hipparchus compiled a catalog of about one thousand naked-eye stars, listing both positions and magnitudes. Just as we do today, this catalog listed the brightest stars as first magnitude, and the faintest visible to the naked eye as sixth magnitude.

Nearly two thousand years later, the English astronomer Norman Pogson quantified measures of star brightness, and found that those ranked as first magnitude were roughly 100 times brighter than stars of the sixth magnitude. He also recognized that each step of one magnitude represented the same ratio of brightness relative to the next. In 1856, Pogson proposed that a step of one magnitude

## Chapter 10: Photometry

should be *defined* as a factor of  $\sqrt[5]{100} = 2.512\dots$  in brightness, thereby making five magnitudes correspond *exactly* to a brightness difference of 100 times.

The Pogson scale was subsequently formalized into our current system of magnitudes, and the basis of modern photometry laid by 1900 with a mixture of visual and photographic measurements. Visual estimates seldom have an uncertainty lower than 0.2 magnitude (about 20% accuracy), and photographic photometry cannot easily be pushed to better than 0.05 magnitude (about 5% accuracy). It was not until after World War II and the invention of the photomultiplier tube that astronomers could routinely measure stellar brightness to 0.01 magnitude (about 1% accuracy), and, with the aid of the 200-inch Hale telescope, extend reliable measurements of stellar brightness to the 20th magnitude.

Today an amateur astronomer with a CCD camera on an 8-inch telescope can reach the 20th magnitude, and more importantly, perform high-quality photometry on stars of the 14th magnitude.

### 10.1.1 Magnitudes Are Comparisons

Our definition of magnitude, derived from the ancient Greeks, implies a comparison between stars. Pogson defined *differences* in magnitude. In fact, the formal definition is written as the logarithm of the ratio of flux (light) from the two stars:

$$\Delta m = m_1 - m_2 = -2.5 \log(F_1/F_2). \quad (\text{Equ. 10.1})$$

What this equation says is that the difference between the magnitude of two stars,  $\Delta m$  or  $m_1 - m_2$ , depends on the ratio of their fluxes  $F_1$  and  $F_2$ , and that's all.

The definition has nothing to say about magnitudes *per se*—it speaks only of magnitude differences. In other words, to determine the magnitude of a star, you must compare its brightness to the brightness of another star *whose magnitude you already know!* To find the magnitude of the unknown star, you compute a ratio of the instrumental responses,  $C_1/C_2$ , find the logarithm, multiply by  $-2.5$ , and add the accepted magnitude of a *standard star*:

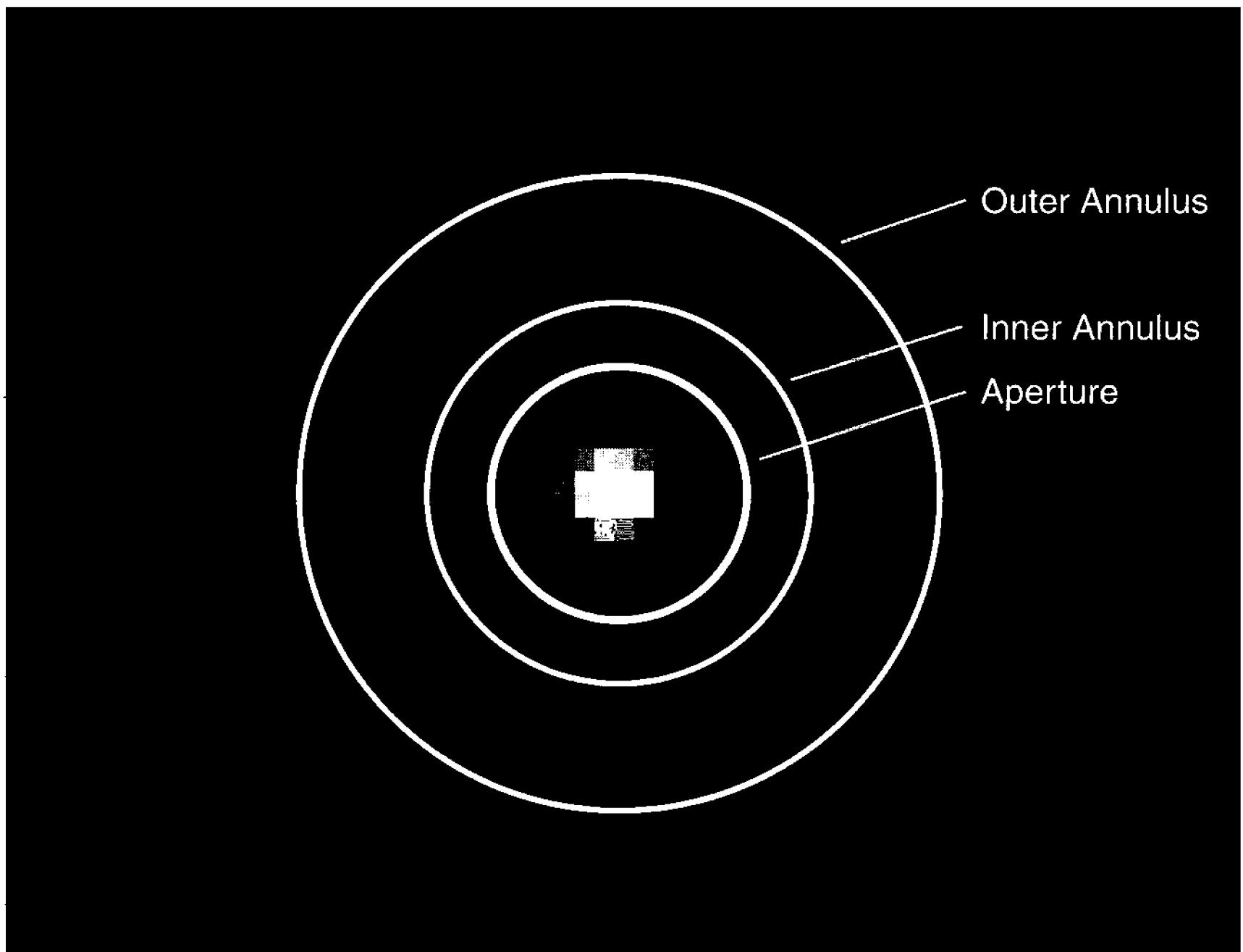
$$m_1 = -2.5 \log(C_1/C_2) + m_2. \quad (\text{Equ. 10.2})$$

The magnitude of the “known” star is thus the standard against which we define the magnitude of the star whose brightness we want to know. Once you grasp this crucial point, photometry makes a lot more sense.

### 10.1.2 Aperture Photometry

Measuring the total light in a star image is simple in principle. The image of a star is a digital copy of a small section of sky. It includes light from the star as well as background sky light. The light of the star is spread over a sizeable number of pixels, and extends to a considerably greater distance than is obvious. To extract the brightness of the star from the image, it is necessary to add up starlight from all of the pixels illuminated by it, and then to estimate the contribution from the sky background and subtract that.

## Section 10.1: Magnitudes: How Bright Is This Star?



**Figure 10.1** To determine a star’s brightness, you find the total pixel value inside the aperture (which contains skylight as well as starlight). Next, you measure the sky brightness between the inner annulus and outer annulus and subtract the skylight from the total in the aperture. The result is the star’s brightness.

In the following sections we describe how to sum the pixels in the star image, how to determine the sky contribution, and how to convert the result into a raw instrumental magnitude.

### 10.1.2.1 Summing the Star’s Light

The classic technique for summing the light from a star is called *aperture photometry*. The *aperture* is a small patch of pixels that contains a star image. Because stars don’t have sharp edges, but instead blend into the surrounding sky, to capture all of a star’s light it is necessary to make the aperture larger than the apparent size of the star image.

A convenient way to express the size of a star image is to treat it as a Gaussian blur, and to express its “radius” as the Gaussian sigma ( $\sigma$ ). Alternatively, the diameter of star images can be expressed as the *full width half maximum*, abbreviated *FWHM*. FWHM is the diameter of the star image at the point that its intensity has fallen to half its peak value. In either case, star image size is measured in pixels. For star images with a Gaussian intensity profile:

$$\text{FWHM} = 2.37\sigma. \quad (\text{Equ. 10.3})$$

## Chapter 10: Photometry

- **Tip:** Click on any star's image with the Star Image Tool in **AIP4Win**, and you'll get back the star's  $(x,y)$  coordinates, its sigma radius, its full width half maximum, and other useful characteristics.

To capture as much light as possible from a star image, the aperture should be sized considerably larger than it. As a rule of thumb, photometrists often set the radius of the aperture to five times the sigma radius of the star image. In an image with tight, well-focused stars, the sigma radius often measures between 0.9 and 1.4 pixels, so an aperture radius of 6 pixels is a good all-around size.

Totalling a star's light is quite straightforward. Given the location of the star image, the photometric software computes the centroid ("center of gravity") of the star image, then totals the value of every pixel inside the aperture radius. Note, however, that the total pixel value includes not only starlight, but also the background glow of the night sky.

In equation form, given  $n_{\text{aperture}}$  image pixels,  $p(n)$ , lying less than distance  $R_{\text{aperture}}$  from the centroid of the star image, you can compute  $C_{\text{aperture}}$ , the total pixel value inside the aperture radius:

$$C_{\text{aperture}} = \sum_{n=0}^{n_{\text{aperture}}} p(n) \text{ [ADU].} \quad (\text{Equ. 10.4})$$

When  $R_{\text{aperture}}$  is very small, the aperture will be ragged at the outside edge and starlight that should be included may be lost; but whenever this radius is reasonably large, the aperture will include all pixels containing significant amounts of starlight.

### 10.1.2.2 Subtracting Sky Background

In classic photoelectric photometry, the sky background brightness was measured by pointing the telescope at a blank patch of sky near the star. CCD photometry offers a better option: sample the sky background in an *annulus* (donut) surrounding the star image. To avoid the inclusion of starlight, the annulus should be somewhat larger than the star aperture, and should extend far enough to provide a statistically significant sample of sky pixel values.

The computation to determine the sky background level is the same one we used for the star image: determine which pixels lie outside the inner annulus radius but inside the outer annulus radius, count and sum the pixels, and compute the average pixel value of the sky. Since the annulus probably covers a sufficiently large area to include faint background stars, it is necessary to eliminate these non-sky contributions to the sky background.

A simple and computationally robust solution is to sort the pixels in the annulus into ascending order. Those that are part of another star image will be brighter than the average pixel value of the sky, so it is necessary to exclude some percentage of the high-value pixels in the sky annulus. To avoid skewing the average, it is also necessary to exclude the same percentage of the low-value pixels.

## Section 10.1: Magnitudes: How Bright Is This Star?

The corrected value for the sky brightness is the mean of the remaining pixel values. Experience shows that excluding the top and bottom 20% of pixels works well for all but the most crowded sky backgrounds.

In equation form, given  $n_{\text{annulus}}$  image pixels,  $p(n)$ , lying greater than distance  $R_{\text{inner}}$  and less than  $R_{\text{outer}}$  from the centroid of the star image, and satisfying the condition of lying between the 20% and 80% percentile in value, you can compute  $C_{\text{annulus}}$ , the total pixel value inside the annulus:

$$C_{\text{annulus}} = \sum_{n=0}^{n_{\text{annulus}}} p(n) \text{ [ADU].} \quad (\text{Equ. 10.5})$$

Total pixel value for both the aperture and the annulus are obtained exactly the same way: by computing the sum of all pixels that meet the geometric and/or pixel value criteria needed to qualify. By the way, for reasons steeped in the history of photometry, you will sometimes hear astronomers refer to the aperture and annulus totals as “counts.”

### 10.1.2.3 Raw Instrumental Magnitude

After the total pixel value of the star aperture and sky annulus have been counted, you can convert raw “counts” into magnitude. However, the resulting measure is not a “real” magnitude until it has been tied to standard stars in the sky. For this reason, the magnitude that you compute is called the *raw instrumental magnitude*. It’s *raw* because it has not been tied to the sky, and *instrumental* because it depends on the properties of your equipment; that is, it depends on your CCD camera, your filters, and your telescope.

In measuring a star image, you have determined four parameters:

- $C_{\text{aperture}}$ , the sum of pixel values in the star aperture,
- $n_{\text{aperture}}$ , the number of pixels in the star aperture,
- $C_{\text{annulus}}$ , the sum of qualified pixels in the sky annulus, and
- $n_{\text{annulus}}$ , the number of pixels in the sky annulus.

In addition to the things you have measured on the image, you also know the integration time,  $t$ , used to make the image, to convert the measured accumulation during integration into the rate at which photons arrived.

Recall now that by definition magnitudes compare stars with one another (see Equation 10.2), yet you have only one star! Since we’re not tying it to real stars, you can introduce a fictitious “star” called the *instrumental zero point*, or  $Z$ . Begin by rewriting Equation 10.2 to separate the instrument responses:

$$m_1 = -2.5 \log C_1 + 2.5 \log C_2 + m_2 \quad (\text{Equ. 10.6})$$

To use this fictitious second star, redefine the terms  $2.5 \log C_2 + m_2$  as  $Z$ . Since the choice of  $Z$  is completely arbitrary, you can choose any convenient value for it. Photometrists usually choose a value that converts raw instrumental magni-

## Chapter 10: Photometry

tudes to magnitudes that sound reasonable for the star they are measuring.

You can convert the total counts star aperture and the sky background into a *raw instrumental magnitude*,  $m$ , for the star, using:

$$m = -2.5 \log\left(\frac{C_{\text{aperture}} - n_{\text{aperture}}(C_{\text{aperture}}/n_{\text{annulus}})}{t}\right) + Z. \quad (\text{Equ. 10.7})$$

What we've done here is to pro-rate the sky total seen in the annulus from the number of pixels in the annulus to the number of pixels in the aperture, and then we've subtracted the resulting sky total. The only assumption we've made is that the sky around and behind the star has the same brightness as the sky that surrounds the star in the annulus. Dividing by the integration time means that without changing your zero point you will get the same raw instrumental magnitude for images taken with different integration times.

Measuring magnitudes from CCD images is both quick and easy. The observer, however, must remain alert to insure that numbers popping up on the computer screen are valid. Before measuring a star image, check the profile to be certain that it is well within the star aperture. If there are stars in the aperture or in the annulus, they can add to the measured star brightness or to the sky background reading.

Images used for photometry should be calibrated before they are measured, but they *must not be scaled* because doing so can change the relationship between photon flux on the detector and image pixel value, thereby destroying the linearity of the data in the image.

- **Tip:** *The Single Star Photometry Tool in AIP4Win computes raw instrumental magnitude using the above formula. You need to know the integration time for the image and supply a value for the zero point. The tool finds the star's centroid position, the total (star – sky) count, the mean sky background level, and the raw instrumental magnitude.*

### 10.1.2.4 Statistical Uncertainty

The total of detected photons that accumulates to form a star image obeys Poisson statistics. (To learn more about Poisson statistics, see Chapter 2.) This means that if a star shines with a mean brightness of 10,000 detected photons per integration, the actual number you'll see in an image will be  $10,000 \pm \sqrt{10,000}$ . Photometry, therefore, always has a built-in uncertainty—but you can determine what the uncertainty should be. Computing the statistical uncertainty “keeps you honest” when you are doing photometry.

Two measures are commonly used to express this uncertainty: the signal-to-noise ratio and standard deviation in magnitudes. Signal-to-noise ratio (SNR) is simply the size of the star signal divided by the amount of noise in the signal. An SNR of 100 is considered good; it means that the signal is 100 times greater than the noise. An uncertainty of 1 part in 100 in determining a magnitude corresponds

## Section 10.1: Magnitudes: How Bright Is This Star?

to about 0.01 magnitude error. To attain a photometric accuracy of  $\frac{1}{100}$  of a magnitude, you need a signal-to-noise ratio of 100, an easy rule to remember.

Now let's define all of the signal and noise sources.

The signal comes from one source only: detected photons (photoelectrons) from the star that you are measuring. The star-minus-sky term,

$$S_{\text{star}} = g(C_{\text{aperture}} - n_{\text{aperture}}(C_{\text{aperture}}/n_{\text{annulus}})), \quad (\text{Equ. 10.8})$$

from Equation 10.7 gives the signal in ADU units. To convert ADUs to electrons (detected photons), we must multiply the star-minus-sky count by the conversion factor,  $g$ , electrons per ADU.

Although there is only one star signal source, there are multiple sources of noise. The most obvious of these is the noise that is associated with Poisson statistics,  $N_{\text{star}} = \sqrt{S_{\text{star}}}$ . If you measured a star against a completely black sky using an ideal detector, the signal to noise ratio would be:

$$\text{SNR} = \frac{S_{\text{star}}}{N_{\text{star}}} = \frac{S_{\text{star}}}{\sqrt{S_{\text{star}}}} = \sqrt{S_{\text{star}}}. \quad (\text{Equ. 10.9})$$

However, in real photometry we must also take the noise sources in the sky and the detector into account:

- $C_{\text{sky}}$ , ADUs of sky background present in every pixel,
- $C_{\text{dark}}$ , ADUs of dark current added to every pixel,
- $\sigma_{\text{ron}}$ , readout noise in electrons r.m.s. added to every pixel,
- $\sigma_{\text{quant}}$ , ADUs of quantization noise from the digitization of the CCD's analog output. Use  $\sigma_{\text{quant}} = 0.29$ .

Because noise from each of these contributors is added to every pixel in the star aperture, their individual total is multiplied by the total number of pixels to obtain their communal total. The noise contribution from the star aperture is:

$$N_{\text{star}} = \sqrt{S_{\text{star}} + n_{\text{aperture}}(gC_{\text{sky}} + gC_{\text{dark}} + \sigma_{\text{ron}}^2 + g^2\sigma_{\text{quant}}^2)}. \quad (\text{Equ. 10.10})$$

Because Poisson and Gaussian noise sources add quadratically, the readout and quantization noise standard deviations are squared before summing.

In the determination of the sky background in Equation 10.8, the sky background count,  $C_{\text{annulus}}$ , is normalized to the same number of pixels as the aperture and subtracted from the aperture pixel value sum. The noise associated with the sky background count is:

$$N_{\text{sky}} = \sqrt{n_{\text{annulus}}(gC_{\text{sky}} + gC_{\text{dark}} + \sigma_{\text{ron}}^2 + g^2\sigma_{\text{quant}}^2)}. \quad (\text{Equ. 10.11})$$

The noise terms in this expression are identical to those in the aperture. Now consider the case in which  $n_{\text{aperture}} = n_{\text{annulus}}$ ; it is clear that the sky contribution to the total noise will double when this noise source is added to Equation 10.11. However, if the number of pixels in the annulus is greater than the number of pix-

## Chapter 10: Photometry

els in the aperture, then  $gC_{\text{sky}}$  will have a lower statistical uncertainty and  $N_{\text{sky}}$  will be proportionately reduced. We can, therefore, replace the term  $n_{\text{aperture}}$  in Equation 10.10 with the following to account for this influence:

$$n_{\text{aperture}} \left( 1 + \frac{n_{\text{aperture}}}{n_{\text{annulus}}} \right). \quad (\text{Equ. 10.12})$$

It is clearly desirable to have as many pixels as practical in the annulus. If the number of annulus pixels equals that of the aperture, the ratio is 2.0; i.e., sky noise doubles. However, by using an annulus with twice as many pixels as in the aperture, the ratio falls to 1.5. On the other hand, as the number of annulus pixels falls below those in the aperture, the determination of the sky value becomes increasingly uncertain and the ratio rises dramatically.

- **Tip:** *AIP4Win's default radii for aperture photometry are 6, 9, and 15 pixels for the aperture, inner annulus, and outer annulus, respectively. The number of annulus pixels is four times the number of aperture pixels, giving a ratio of 1.25.*

Combining the signal and noise terms yields the following for the signal-to-noise ratio in aperture photometry of a star image:

$$\text{SNR} = \frac{S_{\text{star}}}{\sqrt{S_{\text{star}} + n_{\text{aperture}} \left( 1 + \frac{n_{\text{aperture}}}{n_{\text{annulus}}} \right) (gC_{\text{sky}} + gC_{\text{dark}} + \sigma_{\text{ron}}^2 + g^2 \sigma_{\text{quant}}^2)}} \quad (\text{Equ. 10.13})$$

using the value of  $S_{\text{star}}$  computed in Equation 10.8.

It should come as no surprise that computing the statistical uncertainty in aperture photometry is more complicated than computing the raw instrumental magnitude. The magnitude is simply the excess pixel value found in the aperture over the expected sky value; whereas the noise in the signal involves factors closely associated with the CCD and its noise characteristics.

To convert SNR into magnitude error, use the following:

$$\sigma_m = \frac{1.0857}{\text{SNR}}. \quad (\text{Equ. 10.14})$$

The factor 1.0857 converts the fractional uncertainty into magnitudes. This allows you to express the measurement of a star's raw instrumental magnitude in the form  $m \pm \sigma_m$ , giving both the result and the uncertainty of that result.

## 10.2 Putting Photometry to Work

Broadly speaking, astronomers do two types of photometry: differential photometry and all-sky photometry. *Differential photometry* is focused on monitoring one “target” to observe how it changes. The target may be a variable star, an asteroid,

## Section 10.3: Photometric Systems

a star with exoplanet transits, or an exotic target like a quasar or the nucleus of a Seyfert galaxy. *All-sky photometry* is aimed at establishing accurate magnitudes for an object or objects relative to standard stars with well-established magnitudes, and generally requires careful observation and rigorous data reduction with no shortcuts allowed.

Differential photometry is easy because the target and a comparison star are usually in the same field of view, and are observed at the same time, through the same atmosphere, with the same filters—and all that matters is an accurate comparison between the target and the comparison. All-sky photometry is difficult because the target objects have been captured in different images from the standard stars, have been observed at different times, through a different atmosphere, and usually have been through multiple filters.

However, observers know that making CCD integrations at the telescope and extracting magnitudes from the resulting images is just the beginning. Raw instrumental magnitudes are strongly affected by three factors:

- Filter(s) used in making the image(s). Although some observing programs do not require filtered images, many do. A standard set of filters includes U, B, V, R, and I, although just two filters, V and R, are necessary to get started.
- Atmospheric extinction that dims stars. Although this dimming is not particularly apparent to the eye, for photometry it is significant and must be corrected.
- A unique set of instrumental magnitudes defined by the peculiarities of your particular set of filters, your CCD, your observing site and your telescope. To combine your observations with those of others, your magnitudes must be transformed to a standard photometric system.

The section following describes how astronomers measure and then compensate for these factors.

### 10.3 Photometric Systems

The reason that the magnitude scale is defined in terms of standard stars is that it is extremely difficult to measure the absolute flux of light with any precision, but it is fairly easy to compare the flux of one source with that of another. Why should this be so?

The answer lies in another question: What do we mean by “flux?” In theory, of course, we can easily define it as the arrival of some number of photons per second in some well-defined range of wavelengths—that is, in fact, the goal in measuring flux. However, what comes out of the detector, whether that detector is a photomultiplier tube or a CCD image, is an instrumental response to the starlight—a meter reading, a chart deflection, a count in electrons per second, a total pixel value—one number. We have no idea how many photons at each wavelength

## Chapter 10: Photometry

contributed to the response of the detector; we see only its integrated response to all wavelengths. When we write an equation to describe the factors that affect detector response,  $d_{\text{star}}$ , the problem becomes clear:

$$d_{\text{star}} = \int_{\lambda=0}^{\infty} F_{\text{star}}(\lambda) A_X(\lambda) T(\lambda) f(\lambda) Q(\lambda) d\lambda \quad (\text{Equ. 10.15})$$

where  $\lambda$  denotes wavelength,  $F_{\text{star}}(\lambda)$  is the flux from the star reaching the top of Earth's atmosphere as a function of wavelength,  $A_x(\lambda)$  is the transmission of Earth's atmosphere for an air-mass of  $X$  as a function of wavelength,  $T(\lambda)$  is the transmission of the telescope optics as a function of wavelength,  $f(\lambda)$  is the transmission of the filter used in the observation as a function of wavelength,  $Q(\lambda)$  is the quantum efficiency of the detector as a function of wavelength, and the notation  $d\lambda$  indicates that the integration is performed over wavelength.

This equation says that the response of your CCD depends on a large number of factors, all of which vary significantly with wavelength; and the total response that you obtain is the sum of the detector responses at each wavelength. The dilemma of photometry turns out to be figuring out how much each wavelength of light contributes to the flux.

The solution to this dilemma is not to measure all of the wavelengths at once, but to use filters to divide the spectrum into short segments, and measure each color individually. If the segments can be kept narrow enough, the wavelength dependencies of the factors disappear, and the flux equation becomes much simpler:

$$d_{\text{star}} = F_{\text{star}} A_X T f Q \Delta\lambda. \quad (\text{Equ. 10.16})$$

The dependence on wavelength has disappeared because each spectrum segment under consideration,  $\Delta\lambda$ , is made small enough that each of the parameters is constant across such a narrow wavelength span. As a matter of practical concern, however, if we make the range of wavelengths too narrow, the total signal is smaller and more difficult to measure.

By measuring two close-together stars in rapid succession, the atmospheric properties are the same, the telescope optics remain unchanged, the color filter doesn't change, and the detector quantum efficiency is constant—so the like terms in the numerator and denominator cancel:

$$\frac{d_{\text{star}}}{d_{\text{ref}}} = \frac{F_{\text{star}} A_X T f Q \Delta\lambda}{F_{\text{ref}} A_X T f Q \Delta\lambda} = \frac{F_{\text{star}}}{F_{\text{ref}}}. \quad (\text{Equ. 10.17})$$

All this equation says is that all of the factors cancel out for a star observed with the same telescope, filter, and detector through the same air mass. Thus, we can plug the two instrumental responses into Equation 10.2 and find the ratio of the flux of a star relative to that of a reference star, and then compute the difference in magnitude. To accomplish this, we have placed significant constraints on how we make the observation. We must observe stars through the same atmospheric path; and even more importantly, we must measure them through color filters that

**Table 10.1 Primary Standards for UBV System**

Name	<i>V</i>	<i>B-V</i>	<i>U-B</i>	Sp. Type
10 Lac	4.88	-0.203	-1.04	O9 V
$\eta$ Hya	4.30	-0.195	-0.74	B3 V
$\tau$ Her	3.89	-0.152	-0.56	B5 IV
$\beta$ Lib	2.61	-0.108	-0.37	B8 V
HR 875	5.17	+0.084	+0.05	A1 V
HR 8832	5.57	+1.010	+0.89	K3 V
$\alpha$ Ari	2.00	+1.151	+1.12	K2 III
$\alpha$ Ser	2.65	+1.168	+1.24	K2 III
$\varepsilon$ CrB	4.15	+1.230	+1.28	K3 III
$\beta$ Cnc	3.52	+1.480	+1.78	K4 III

pass a narrow range of wavelengths.

The oldest photometric *color* is the natural response of the human eye. The **peak** sensitivity of the dark-adapted eye occurs at a wavelength of 510 nm, or bluish-green light. Toward shorter wavelengths, sensitivity falls to half maximum at 468 nm; and toward longer wavelengths, it falls to half maximum at 550 nm. In an engineering sense, the full-width half-maximum bandwidth of the eye is only 84 nm, though it retains some sensitivity over a much wider spectral range. Magnitude observations made in the color system of the eye are called *visual magnitudes*.

Technically speaking, the term “photometry” applies to visual magnitudes **only**. As scientists began to measure light (i.e., electromagnetic radiation) outside the visible spectrum, the term “radiometry” replaced photometry in almost every field of science *except* astronomy. However, astronomers have always been a little **different!**

The second photometric color system came into existence around 1890, with the advent of astronomical photography. It didn’t take long before astronomers realized that reddish stars appeared several magnitudes fainter in photographs than they did to the eye. The sensitivity of early orthochromatic photographic plates peaked in blue light at 460 nm, fell to half at 495 nm, and plummeted to nothing by 520 nm. In those days, famous astronomers like E.E. Barnard developed their plates by inspection under a red safelight! Magnitudes derived from blue photographic plates were called *photographic magnitudes*.

Technology soon brought forth dye-sensitized panchromatic plates capable of recording “all” colors from blue to orange. By taking photographs using panchromatic plates with a yellow filter, astronomers devised a system of *photovisual*

## Chapter 10: Photometry

**Table 10.2 UBVRI Filters for CCD Photometry\***

Filter	Filter Prescription (total thickness = 5 mm)
U	1mm UG1 + 2mm S8612 + 2mm WG295
B	1mm GG385 + 2mm BG1 + 2mm BG39
V	2mm GG495 + 3mm BG40
R	3mm OG570 + 2mm KG3
I	2mm RG9 + 3mm WG295

\*From Bessell, "UBV(RI) Filters for CCD Photometry," *CCD Astronomy*, Vol. 2, No. 4, p. 21.

magnitudes—magnitudes measured with light between 495 nm and 580 nm wavelength.

The availability of photographic and photovisual magnitudes opened new areas of research. Suddenly astronomers had a way to measure the temperature of stars from their *color index*—from the difference between their photographic and photovisual magnitudes. The color index correlates well with the surface temperatures of stars and also with the spectral type; by plotting the color index against the photovisual magnitude, the main sequence and giant branch stand out clearly. Applied to star clusters, the *color-magnitude diagram* soon led to major advances in calibrating the astronomical distance scale and understanding stellar evolution.

The color index is defined as:

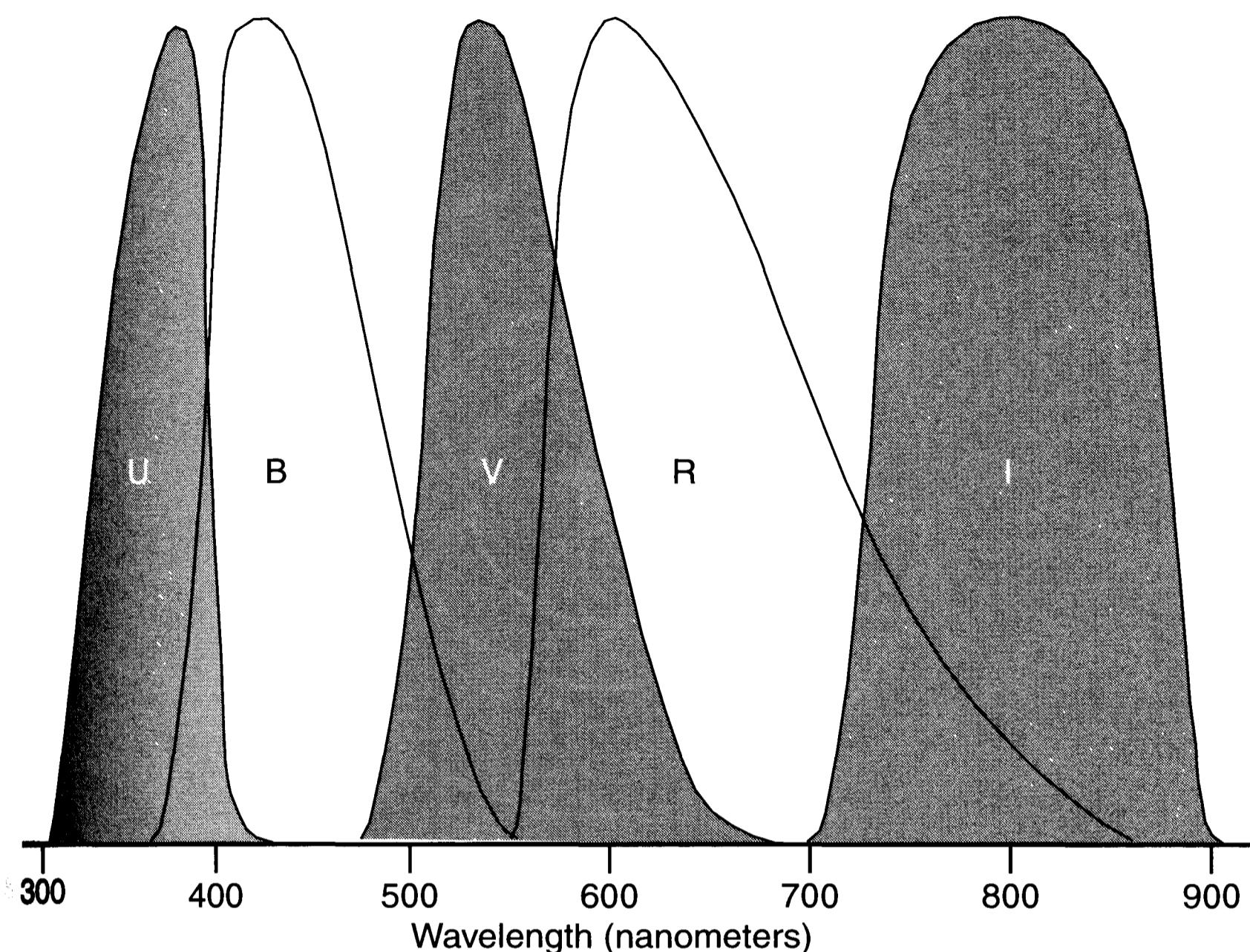
$$CI = m_{pg} - m_{pv} + \text{constant} \quad (\text{Equ. 10.18})$$

where  $m_{pg}$  is the magnitude measured on a blue plate,  $m_{pv}$  is the magnitude measured on a yellow plate, and the constant is chosen so that the color index equals zero for type A0 stars. It is negative for stars hotter and bluer than A0, and positive for stars cooler and redder than A0. The term “color” has since been generalized to mean the difference between a short-wavelength magnitude and a long-wavelength magnitude in any two color systems.

Although photoelectric photometry had been around for decades, it did not become practical until shortly after World War II, with the introduction of the 1P21 photomultiplier tube. Two astronomers, Harold Johnson and W.W. Morgan, sought a photoelectric color system that would replicate photographic and photovisual magnitudes using blue and yellow filters; and adding a third color, ultraviolet, to aid in discriminating among the different spectral types. Although many other photometric systems exist, the Johnson and Morgan UBV ( $U$  = Ultraviolet,  $B$  = Blue,  $V$  = Visual) system, with extensions by Andrew Cousins and John Menzies to include  $R$  (= Red) and  $I$  (= Infrared), has become the *de facto* standard for photometry.

The zero point of the original *UBV* system was defined by ten primary standard stars (see Table 10.1). Johnson and Morgan also established 98 secondary standard stars spread around the sky, and an additional several hundred in the Bee-

## Section 10.3: Photometric Systems



**Figure 10.2** The filters in the *UBV(RI)* system divide the spectrum into roughly equal intervals from 350 nm to 820 nm wavelength, from the deep violet of the U filter through the deep red and infrared of the I filter. The V filter mimics the sensitivity curve of the photopic (light-adapted) human eye.

hive and Pleiades star clusters and in IC4665. Because the standard stars in the system were located in the northern hemisphere and too bright for many large telescopes, astronomer Arlo Landolt rigorously established 642 new secondary standards between 10.5 and 12.5 in *V* magnitude in 24 Selected Areas within a few degrees of the celestial equator. These were published in *The Astronomical Journal*, volume 78, number 9, November 1973.

Today, the Landolt standards for *UBV* (extended to include *RI* colors of Cousins and Menzies) have supplanted those of Johnson and Morgan. Landolt *UBV(RI)* standards are especially useful to amateurs who want to do photometry with CCD cameras because they are spaced one hour of right ascension apart; so several Selected Areas are always visible. The Landolt stars are bright enough to give good signal to noise ratio, but not so bright that they quickly saturate the CCD.

Landolt standards are genuine photometric standards. The magnitudes tabulated in the *Guide Star Catalog* and USNO astrometric catalogs range from unreliable to useless for photometry and should never be used as standard stars. Observers who use *Guide Star Catalog* or USNO magnitudes can expect errors reckoned in whole magnitudes.

Photoelectric photometers brought a new level of precision to photometry. If

## Chapter 10: Photometry

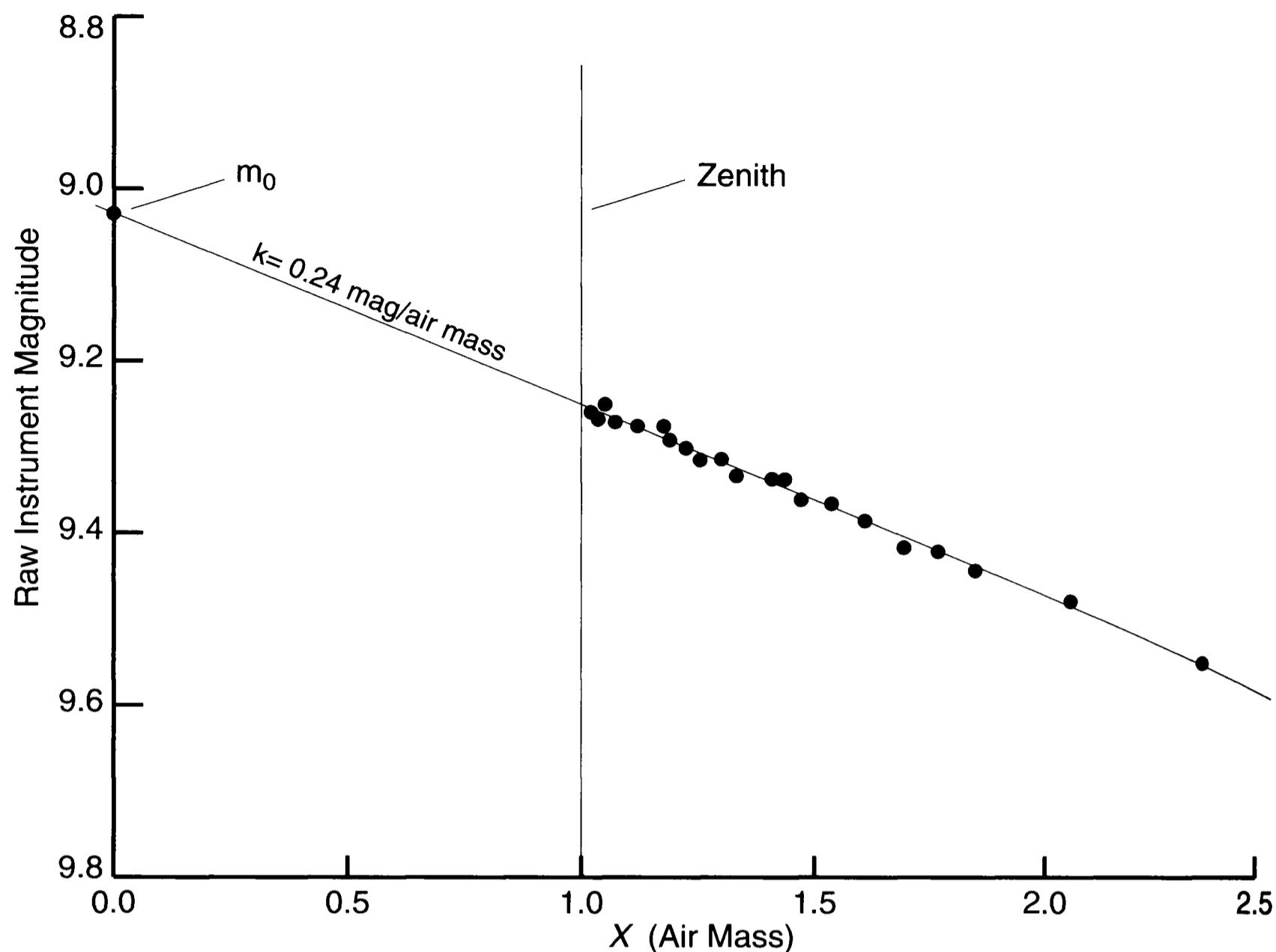
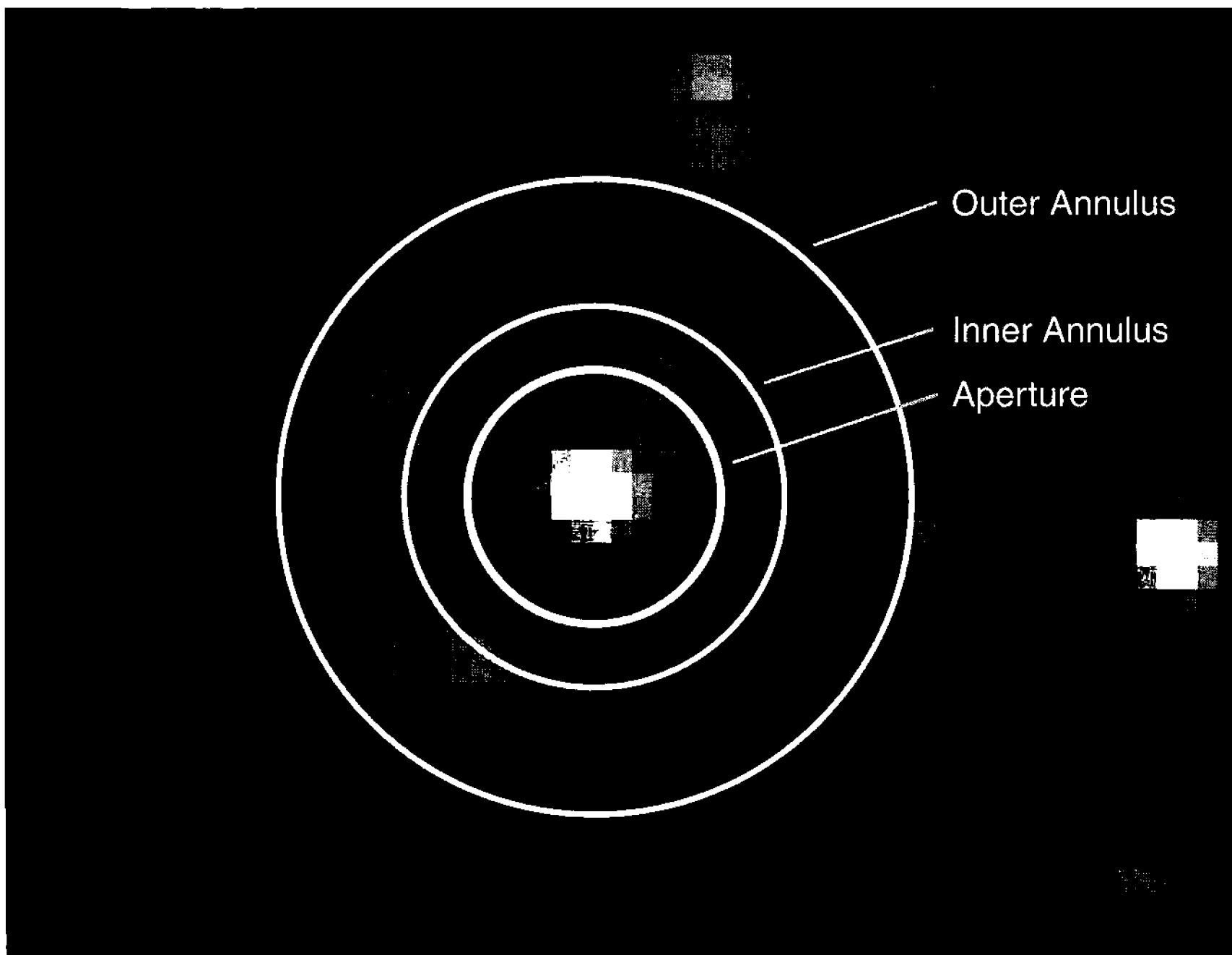


Figure 10.3 When a sky is transparent and cloud-free, atmospheric extinction depends linearly on  $X$ , the air mass. Once you determine the extinction coefficient for a series of measurements at different zenith distances, you can extrapolate to obtain  $m_0$ , the star's brightness above the atmosphere.

done with great care, visual photometry had an uncertainty of 0.20 magnitude; and photographic photometry had an uncertainty of 0.05 magnitude. Photoelectric photometry, however, was accurate to better than 0.01 magnitude. With the advent of the *UBV* system, astronomers needed some way to convert old-style photometry into the new system. Johnson and Morgan had done an excellent job matching their *V* scale to photovisual magnitudes, but the old-style color indices did not match the new  $(B - V)$  colors. However, a simple linear equation— $(B - V) = 0.16 + 0.92 CI$ —enabled astronomers to transform photographic *CI* colors into the new  $(B - V)$  color system. The disagreement in *B* occurs because the ranges of wavelengths detected by the photographic emulsion and the filter-plus-photomultiplier combination did not exactly match; so the two systems measured the star's light at slightly different effective wavelengths. In fact, the same problem occurs between any two photometers—even photometers that are identically constructed—because reflective coatings on telescope mirrors, filters used in photometers, and the spectral sensitivity of photomultiplier tubes vary slightly from one to the next. The readings that come from a photometer are called *raw instrumental magnitudes*; to be useful, they are *transformed* in the *UBV* system.

This system is defined by the properties of Johnson and Morgan's original photometer, the ten standard stars, and a few hundred secondary standard stars; in

## Section 10.3: Photometric Systems



**Figure 10.4** Photometry becomes difficult in crowded star fields. The star aperture contains an unwanted star. The sky annulus contains several star images plus irregular patches of nebulosity, making it difficult to determine how much of the light in the aperture comes from the sky around the star.

other words, raw instrumental magnitudes from Johnson and Morgan's photometer are the *UBV* system. To measure magnitudes in the *UBV* system, other astronomers build photometers that are physically similar to Johnson and Morgan's, measure the *UBV* standards, and then determine transformation coefficients that convert raw instrumental magnitudes into *UBV* magnitudes.

CCDs are obviously quite different from photoelectric photometers; but with appropriate filters, CCD images yield data that can be transformed into accurate magnitudes in the *UBV* system. The key is to make or purchase filters that, when used with a CCD, match the passbands of the Johnson and Morgan filters with a photomultiplier tube. The peak wavelength of the standard *U* color is 367 nm with a passband 66 nm wide. The standard *B* color peaks at 436 nm with a passband of 94 nm, and the standard *V* peaks at 545 nm with a passband of 88 nm. The *R* and *I* magnitudes (added by Cousins a decade after the original *UBV* system was defined) peak at 638 nm and 800 nm respectively, with passbands of 138 nm and 149 nm.

Although the sensitivity curves of typical photomultiplier tubes and typical CCDs are very different, judicious combinations of filters reproduce the standard Johnson-Cousins *B*, *V*, *R*, and *I* colors quite well with CCD cameras. Because many CCDs have low blue sensitivity, making it difficult to obtain good images

## Chapter 10: Photometry

in  $U$  and  $B$ , few CCD observers bother with  $U$  and some must also ignore  $B$ . Ideally, photometric filters should be “tuned” to match the spectral sensitivity curve of a specific CCD; but a standard filter set from Optec, Inc., Murnahan Instruments, Custom Scientific, or Schuler Astro-Imaging, and reasonable care in determining transformation coefficients, should satisfy the needs of most observers.

### 10.3.1 From CCD Images to the Standard System

As you have seen in the foregoing section, subtle differences between apparently identical detectors, filters, and sky conditions can produce different raw instrumental magnitudes. For many observing programs, it is necessary to eliminate these differences, so that regardless of the telescope, sky, detector, and filters used, all observers report their magnitudes in the same standard system.

Transformation from the raw instrumental magnitudes measured from CCD images to the standard system is a two-step process. The first step is correcting for atmospheric extinction; the second step is transforming extinction-corrected instrumental magnitudes to the standard system. Sections 10.3.2 and 10.3.3 describe these two steps.

To distinguish between the names of the standard filters, standard magnitudes (*i.e.*, magnitudes in the  $UBV(RI)$  system), and instrumental magnitudes (*i.e.*, the magnitudes measured from your images), we will use the following notation:

- filters are designated by upper-case roman: U, B, V, R, I.
- standard magnitudes by upper-case italics:  $U$ ,  $B$ ,  $V$ ,  $R$ ,  $I$ .
- instrumental magnitudes by lower-case italics:  $u$ ,  $v$ ,  $b$ ,  $r$ ,  $i$ .

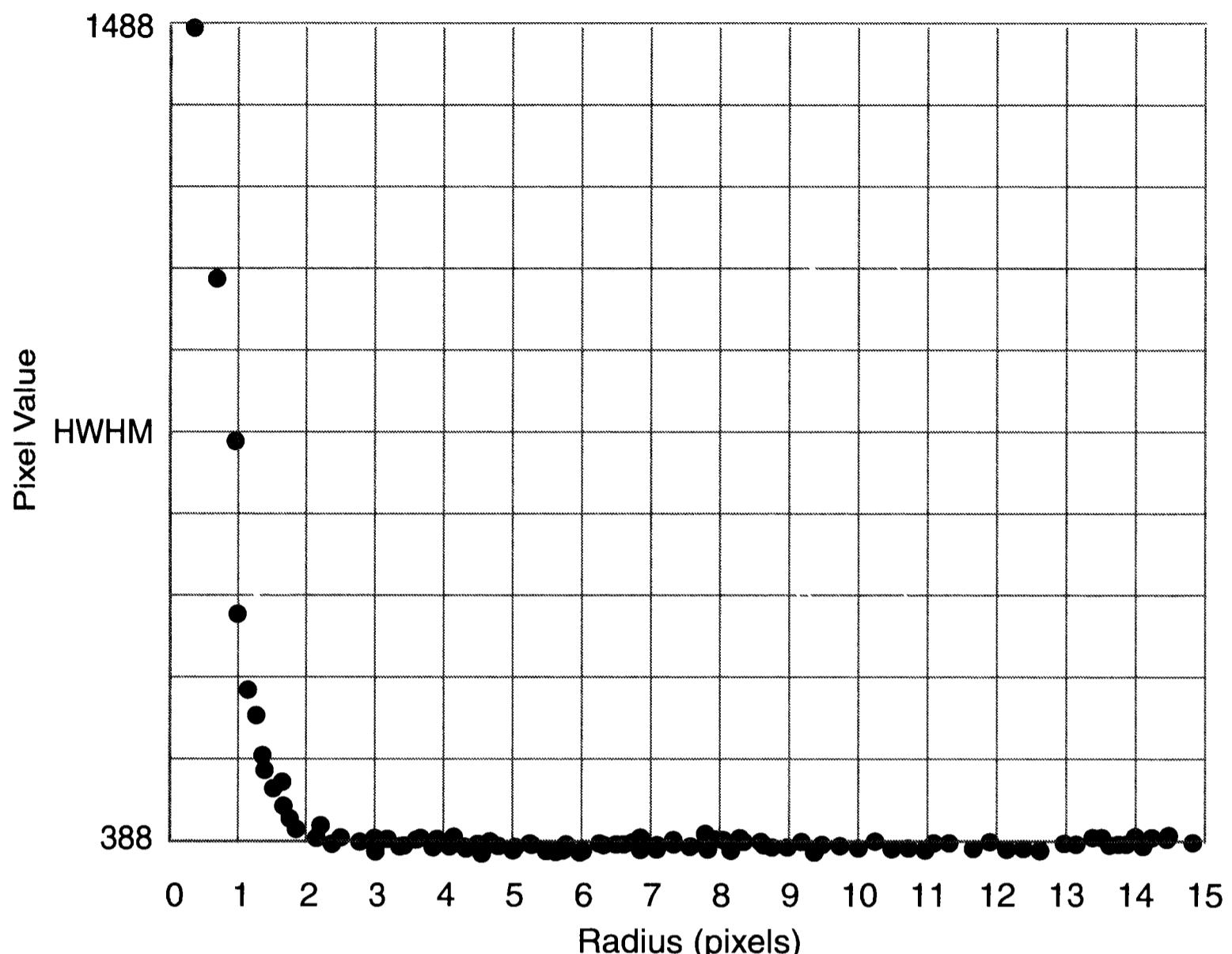
For example, suppose that you make a set of photometric images using V, R, and I filters. When you measure the raw instrumental magnitudes, you would obtain  $v$ ,  $r$ , and  $i$  magnitudes. After correcting them for extinction and converting to the standard  $UBV(RI)$  system, you would report your measurements as  $V$ ,  $R$ , and  $I$  magnitudes.

### 10.3.2 Atmospheric Extinction

“Red sky at night, sailors delight” runs the folk rhyme. As astronomers, we know that the Sun reddens as it nears the horizon because the atmosphere scatters and attenuates blue light more than red. To the photometrist, equipped to measure small changes in the intensity and color of starlight, the atmosphere is a variable strength filter that must be measured and corrected for.

Imagine carrying out the following experiment: you know that a star of spectral type A0 is going to pass exactly through the zenith, and you set up to perform CCD photometry on it. The night is beautifully clear. You install a standard V filter in front of the CCD. As the star passes through the zenith, you take an image, and every ten minutes thereafter you take another until the star finally sets. From your collection of images, you measure the raw instrumental magnitude of the star. As you expected, it becomes dimmer as it sets. After measuring the images,

## Section 10.3: Photometric Systems



**Figure 10.5** The profile of an isolated star image shows a smooth decline from a peak value and a uniform sky background. From the graph, you can see that the peak value in the star is 1488, well below saturation; the sky background is 388, and the HWHM, at 1.0 pixels, is somewhat smaller than desirable.

the data from the experiment consist of a list of times (starting when the star passed through the zenith) and an instrumental magnitude corresponding to each time.

Look at your data. When you graph the time versus the magnitude, it plots out rather nicely as a curving line. You are pleased because it looks as if there is an underlying law at work; if only you can figure out what the curve is, or better yet, figure out how to make the curve into a straight line.

Since the star is obviously at its brightest when it is straight overhead, you decide to plot the distance from the zenith versus magnitude. A little spherical trigonometry gives you  $\zeta$ , the angle between the zenith and the star:

$$\cos \zeta = \cos \phi \cos H \cos \delta + \sin \phi \sin \delta \quad (\text{Eq. 10.19})$$

where  $\phi$  is your latitude,  $\delta$  is the declination of the star, and  $H$  is the hour angle of the star; i.e., the length of time since the star crossed the meridian. When you graph  $\zeta$  versus magnitude, it is only to experience the disappointment of seeing another wild curve.

You are now forced to think a bit. The attenuation of the star's light clearly depends on how much air it passes through. Each time the light passes through some length of air, it loses a fixed proportion of its light, which means that the star's brightness declines exponentially with the length of the air path (called the

## Chapter 10: Photometry

*air mass*). However, you are measuring the star's brightness as a magnitude, which is a logarithmic function. Since the log of an exponential function is linear, you've found the answer: the star loses a constant amount in magnitudes for each unit of air mass it traverses!

Another couple minutes' rummaging around in your math books from college reveals the answer: the length of the air path is proportional to the secant of  $\zeta$ , the inverse of  $\cos\zeta$ , which you computed for your last graph. So at last you can write out the equation for the air mass,  $X$ :

$$X = \sec\zeta = 1/(\cos\phi\cos H\cos\delta + \sin\phi\sin\delta). \quad (\text{Equ. 10.20})$$

A few more minutes of plotting reveals a beautiful linear relationship between the length of the air path (which equals the secant of the zenith angle) and the dimming that the star has suffered, in magnitudes.

You quickly realize several things. The length of the air path is exactly 1 when a star is straight overhead, at the zenith. However, as a star moves away from that point and the angle from the zenith increases, the air mass increases. The increase comes on slowly at first, and only reaches 2.0 at a zenith distance of  $60^\circ$  ( $30^\circ$  up from the horizon). The effect is hardly noticeable to the eye until a star gets within 20 to 25 degrees of the horizon. Lower in the sky, the air mass increases rapidly.

On your plot you see that the slope of the line tells you how many magnitudes dimmer the star becomes for unit of air mass that it traverses—a quick check of the numbers shows an increase of 0.24 magnitude per unit air mass (i.e., the star becomes less bright). Finally the grand revelation: even though you *cannot* measure a star outside the atmosphere, you *can* extend your line all the way to zero air mass; in other words, from a series of measurements made *within* Earth's atmosphere, you can compute how bright the star would appear if measured *outside* it.

This experiment is well worth doing if you contemplate trying your hand at photometry. Seeing is believing: you can indeed compensate for atmospheric extinction. If you try the same experiment with different filters, you will discover that extinction is greater at short wavelengths and less at long wavelengths, which you already knew because sunsets are red.

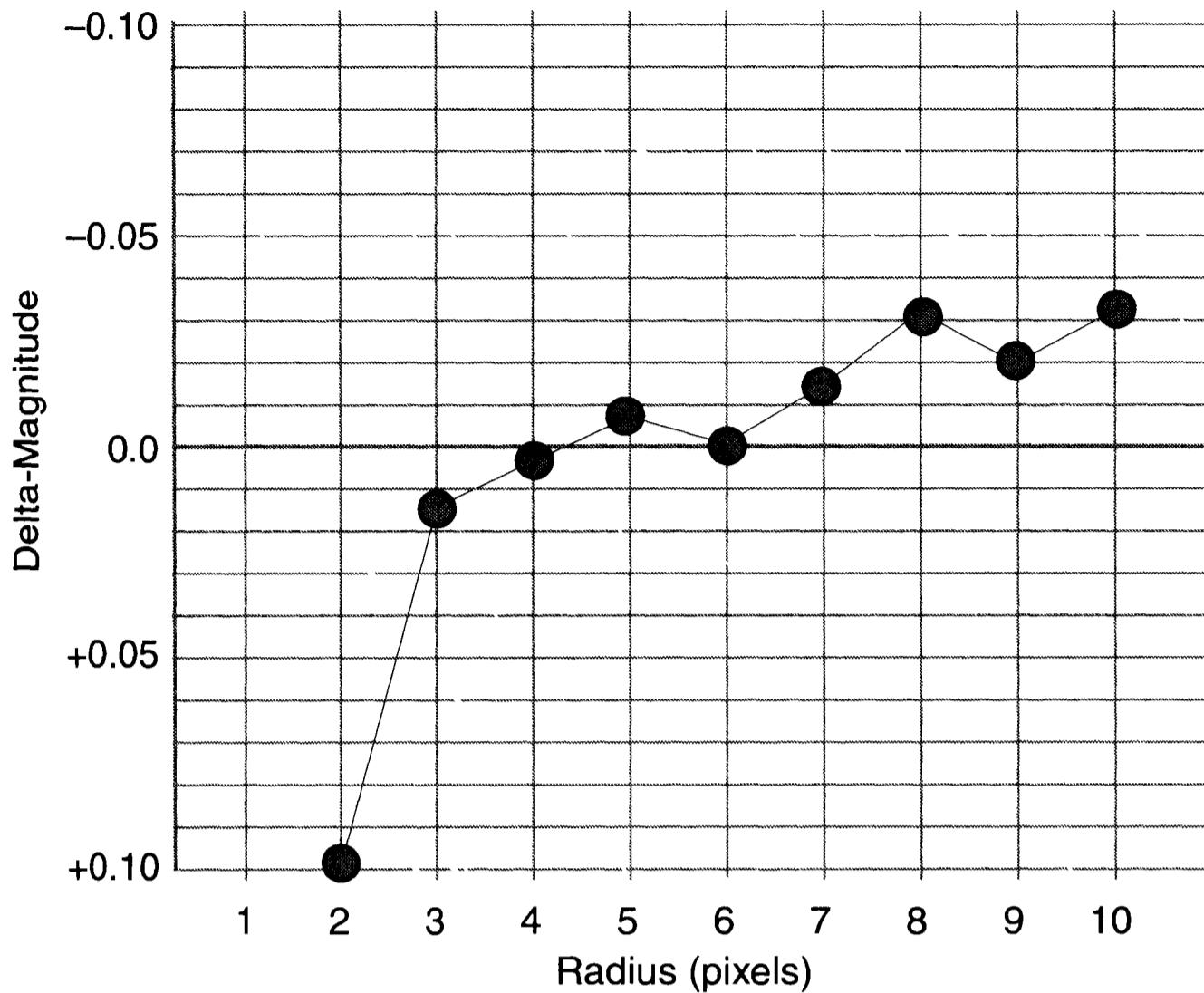
On a technical level, extinction is slightly more complicated. The simple equation for the air mass given above must be modified to correct for the curvature of Earth with additional terms:

$$\begin{aligned} X = \sec\zeta &- 0.0018167(\sec\zeta - 1) \\ &- 0.002875(\sec\zeta - 1)^2 \\ &- 0.000808(\sec\zeta - 1)^3. \end{aligned} \quad (\text{Equ. 10.21})$$

The secant is valid from the zenith down to 30 degrees elevation; this equation is good down to an air mass of ten, or about 6 degrees altitude.

In your simple experiment, you found that extinction measured through a V-band filter obeys the following simple relationship:

## Section 10.3: Photometric Systems



**Figure 10.6** The curve of growth graphs raw instrumental magnitude versus the radius of the aperture. Although the curve of growth does not plateau for this star, the difference between radii of 4, 5, and 6 pixels is only about 0.01 magnitude, indicating that radii of 4, 5, and 6 pixels should work equally well.

$$v_0 = v_X - k'_v X \quad (\text{Equ. 10.22})$$

where  $v_X$  is the raw instrumental magnitude measured through the air mass,  $X$ ,  $k'_v$  is the extinction coefficient for your V filter, and  $v_0$  is the extinction-corrected instrumental magnitude of the star above Earth's atmosphere (with a zero subscript to remind you of zero air masses). At sea level,  $k'_v$  is typically about 0.24 magnitude per air mass, but it falls to 0.15 magnitude per air mass at dry, high-altitude sites. If this sounds a bit complicated, remember that it's really nothing but a straight-line graph that relates raw instrumental magnitudes to air mass.

However, because extinction is stronger at short wavelengths, as a star sets the blue-end wavelengths in the passband are absorbed more strongly than the red; so the center of the passband shifts toward longer wavelengths. In order to compensate accurately for extinction, you have to measure the star's color also, and then correct for it. The simple equation becomes:

$$v_0 = v_X - k'_v X - k''_v X(b - v)_X. \quad (\text{Equ. 10.23})$$

The  $k''_v$  term is called the second-order extinction coefficient. It is really nothing but a small correction that depends on the  $(b - v)$  color of the star. Luckily, this term is so small that astronomers simply ignore it when measuring magnitudes through a standard V filter.

Although it might seem logical to set up analogous equations for each of the other colors you are measuring (*i.e.*, U, B, R, and I), astronomers measure V and

## Chapter 10: Photometry

then use the other filters to derive the colors  $(b - v)$ ,  $(u - b)$ ,  $(v - r)$ , and  $(v - i)$ . Although photometry seems perverse at times, the internal logic is consistent: color is a differential measurement. For  $(b - v)$  and  $(u - b)$ , the extinction equations look like this:

$$\begin{aligned}(b - v)_0 &= (b - v)_X - k'_{(b-v)} X - k''_{(b-v)} X(b - v) \\ (u - b)_0 &= (u - b)_X - k'_{(u-b)} X - k''_{(u-b)} X(u - b)\end{aligned}\quad (\text{Equ. 10.24})$$

Nothing new happens here: you simply take a raw instrumental color, correct extinction for that color, and make a small second-order correction that depends on the star's color. Second-order corrections are small and hard to measure. In fact, rather than measure  $k''_{(b-v)}$ , many photometrists simply assume a value of 0.03; and in the *UBV* system,  $k''_{(u-b)}$  is set to zero by definition. Thus, the extinction corrections reduce to:

$$\begin{aligned}(b - v)_0 &= (b - v)_X - k'_{(b-v)} X - 0.03 X(b - v) \\ (u - b)_0 &= (u - b)_X - k'_{(u-b)} X\end{aligned}\quad (\text{Equ. 10.25})$$

If these equations didn't *look* so hideous, their simplicity would be obvious!

There are three basic ways to deal with extinction: the all-sky approach, the shotgun-scatter approach, and the differential photometry approach.

The first approach is used in all-sky photometry; that is, photometry done on stars over the entire sky. In addition to observing program stars, you observe several extinction stars over a range of zenith distances during the night. This enables you to determine extinction coefficients that you can apply to the program stars. To find the best values for the extinction coefficients, you use the method of least squares (or more likely, the software that you are using to reduce the data uses the method of least squares). With any luck, you can piggyback observations of extinction stars with those of standard stars (more about these in the next section).

The shotgun-scatter approach is best for observers in sites with poor skies, because it is fast and uses stars spread over the whole sky. The idea is to measure standard stars low in the east and west, and then several more near the zenith—all in quick succession. These data sample the extinction curve at widely separated air masses, allowing the observer to determine the slope easily. As a follow-up to guard against changing atmospheric conditions, you should observe a few more standards at intermediate elevations during the course of the night. At good sites, the photometrist simply assumes that well-determined standard values are valid, and does not measure extinction at all.

The differential photometry method, which is not valid for the all-sky method, is to ignore extinction and make differential measurements between your program star and a comparison star. The idea behind this method is that extinction is virtually the same for two stars in the same field of view; therefore to an excellent approximation, extinction cancels out. Of course, if the stars have very different colors, extinction will not be the same for the two—so you need to be careful in your choice of comparison stars. Differential photometry is a powerful technique

## Section 10.3: Photometric Systems

with CCD cameras because you can “sit” on the object (with the comparison star in the same field of view) and take image after image, following the behavior of the program star in time. Differential photometry even works through light clouds, as long as you can still obtain reasonably good images.

### 10.3.3 Transformation to the *UBV(RI)* System

To measure the time of minimum light of an eclipsing binary star, or to monitor outbursts from a dwarf nova, it is usually not necessary to convert instrumental magnitudes to the standard system. However, to calibrate the magnitudes of comparison stars on AAVSO charts for visual observers, it is necessary to insure that the magnitudes fit into the standard *UBV* system. Once you have established a set of reliable transformation coefficients for your telescope, filter, and CCD combination, you can correct your raw instrumental magnitudes for extinction and then transform them to the standard *UBV* system.

Observationally, you need to take images of a dozen or so standard stars through each of your filters, and you also need images of several extinction stars over a wide range of zenith distance to establish extinction coefficients for the night. After measuring raw instrumental magnitudes for all of the standard stars and extinction stars from your images, you determine the extinction coefficients and use them to correct the raw instrumental magnitudes of the standard stars.

Consider your data at this point: for each standard star, you have its accepted *V* magnitude, and your instrumental magnitude, *v*. If you observed with other filters, you also have the accepted (*B* – *V*) color and the “natural” color from your system, (*b* – *v*). The transformation looks like this:

$$V = v - \epsilon(B - V) - Z_V \quad (\text{Equ. 10.26})$$

where  $Z_V$  corrects the zero point of your instrumental magnitudes to the standard system, and  $\epsilon$  is a color-dependent term that corrects for minor differences in the effective wavelengths of the filters you are using. Since you know *V*, *v*, and  $(B - V)$  for a dozen stars, you merely perform a simple least-squares fit to obtain  $Z_V$  and  $\epsilon$ .

You follow a similar procedure for the colors. Since you have the standard colors, (*B* – *V*) and (*U* – *B*), and the instrumental colors, (*b* – *v*) and (*u* – *b*), for each of the stars, you look for transforms in the form:

$$\begin{aligned} (B - V) &= \mu(b - v) - Z_{(B-V)} \\ (U - B) &= \psi(u - b) - Z_{(U-B)}, \end{aligned} \quad (\text{Equ. 10.27})$$

where  $\mu$  and  $\psi$  compensate for differences in passbands and effective wavelength, and the two constants  $Z_{(B-V)}$  and  $Z_{(U-B)}$  adjust the zero points.

Note that because corrections for extinction and transformation in the standard system are linear transforms, they can be reduced to a single linear transform.

In summary, stellar magnitudes must be measured over a well-defined range of wavelength. To define and limit the wavelength range reaching the CCD, images intended for photometry should be taken through a filter that has been de-

## Chapter 10: Photometry

signed to match the passband of a standard photometric color system. The *UBV(RI)* color system is the *de facto* standard in astronomy. Magnitudes measured from a properly filtered CCD image are called raw instrumental magnitudes. For all-sky photometry, after the stars are measured, the raw instrumental magnitudes must be corrected for atmospheric extinction and transformed to the standard system. For differential photometry, you simply compare the measured magnitude of the star you are observing with a comparison star in the same image.

### 10.4 Photometric Observing

This section sketches the practical side of the three basic types of CCD photometry: all-sky, “do-what-you-can,” and differential. Use the method that is most appropriate for your circumstances.

The goal of differential photometry is to produce accurate magnitude differences between (supposedly) steady comparison stars and program stars. Differential photometry asks, “How has this star changed?” Not surprisingly, this technique is much easier than all-sky photometry because, to a first approximation, extinction coefficients don’t matter (both stars suffer the same extinction); and so long as the comparison star has a similar color to the variable, transformation to the standard system is not necessary (because you are interested only in how much the star has changed).

The goal of all-sky photometry is to be able to point your telescope at any sky location, shoot images, and produce magnitudes in the standard system. In a session of all-sky work, you must determine extinction coefficients so that you can correct raw magnitudes for that factor; and you must determine transform coefficients in order to convert instrumental magnitudes into the standard system—which imposes a burden on the observer to record extinction stars and standard stars as well as the program stars that are the point of the observing program. All-sky photometry seeks an exact answer to the question, “How bright is that star?”

“Do-what-you-can” photometry is a simplified method of all-sky photometry advocated by photometrist Brian Skiff of the Lowell Observatory. It is designed to allow amateurs to make meaningful measurements with short observing sessions under poor skies. Briefly, you determine the instrumental transforms and measure extinction values carefully just once, and thereafter sandwich photometric images of variables between images of standard fields. Reducing the data is equally simplified by adjusting the “fit” of the standards and then assuming that variables can be adjusted the same way. “Do-what-you-can” is a practical way for amateurs to produce good results.

#### 10.4.1 Preparing to Observe

Photometry demands forethought, careful record keeping, and meticulous procedures in extracting the data contained in the images. Although an observatory is not absolutely necessary, nothing makes for better observations than a controlled environment with everything at hand and ready to go.

## Section 10.4: Photometric Observing

An observing session begins well ahead of time, with the preparation of a list of targets. Some observers track several dozen stars on a nightly, weekly, or monthly basis; while others observe eclipsing binaries drawn from an ever-changing list of stars that need “work.” Some observers run intensive programs on targets of opportunity such as newly-discovered variables, novae, or supernovae; while others work closely with professionals to monitor objects like X-ray binaries for sudden (and unpredictable) activity—which, if detected, makes it the target of an orbiting X-ray observatory. Whatever your objective, you must organize and have at hand the necessary finding charts for program, extinction, and standard stars. With your observing plan in hand, you are ready to begin.

- Turn on the CCD camera early. Allow it an hour to reach thermal equilibrium before you start observing. Open the telescope so that it too reaches air temperature.
- Set the clock in your image-logging computer. The best way is to log onto a time service web site that will synchronize the clock in your computer to the correct time. Although computer clocks are not terribly accurate, most maintain time well enough for a night’s observations. Another solution is to install a GPS card in your PC, and you’ll have precise time all the time.
- Before you begin work, start a new page in your observing notebook. Note the date in civil time (i.e., November 5/6, 2007) as well as the Julian date. Write down the equipment in use (it’s amazing how quickly you can forget important details if you don’t record them), your planned observing program, and notes on the sky condition.

Throughout the evening, continue making notes. Many high-tech astronomers have discovered the hard way that notes written on paper are easier to access and last longer than text files on a hard disk. Without supporting documentation your data can lose their value.

The following sections present brief scenarios that illustrate strategies and methods typical of all-sky, do-what-you-can, and differential photometry.

### 10.4.2 Differential Photometry

In CCD photometry, all of the stars in an image have been observed at the same time and through very nearly the same atmospheric path. As a result, atmospheric extinction is the same for all of them, and variations in atmospheric transmission due to haze or light clouds very nearly cancel out. These simplifications are the basis for differential photometry.

Differential photometry varies considerably depending on the observing program, but the underlying principles remain constant: obtaining a series of images showing an object of interest (referred to as “V”), a comparison star (a normal star that hopefully does not vary, called “C1”), and a second comparison star

## Chapter 10: Photometry

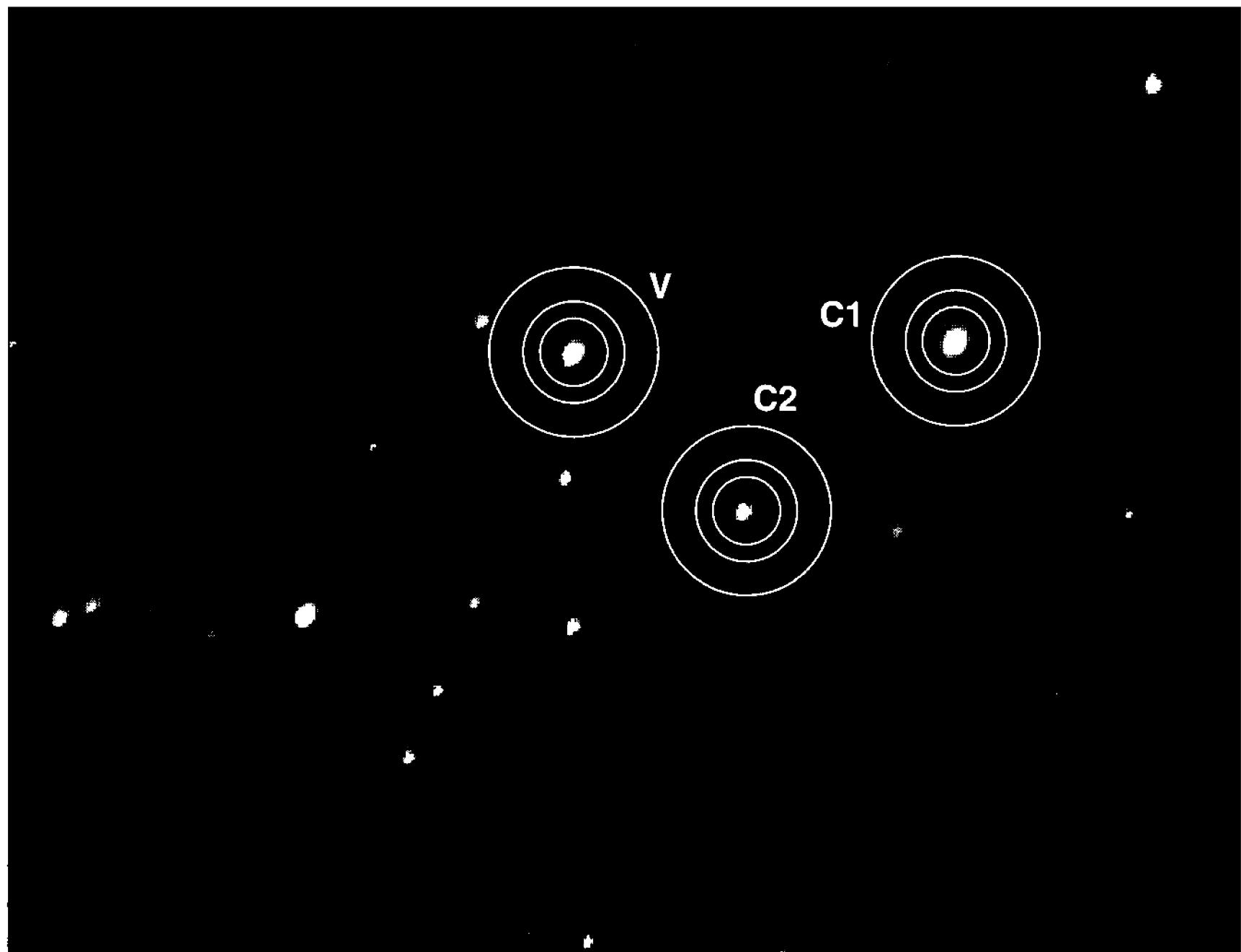


Figure 10.7 Phil Kuebler obtained differential photometry on BR Cygni with 40 images taken at two-minute intervals with a Cookbook camera, a 10-inch SCT, and 60-second integrations with a V filter. Despite soft images and poor tracking, he got excellent photometric results. The light curve appears in Figure 10.8.

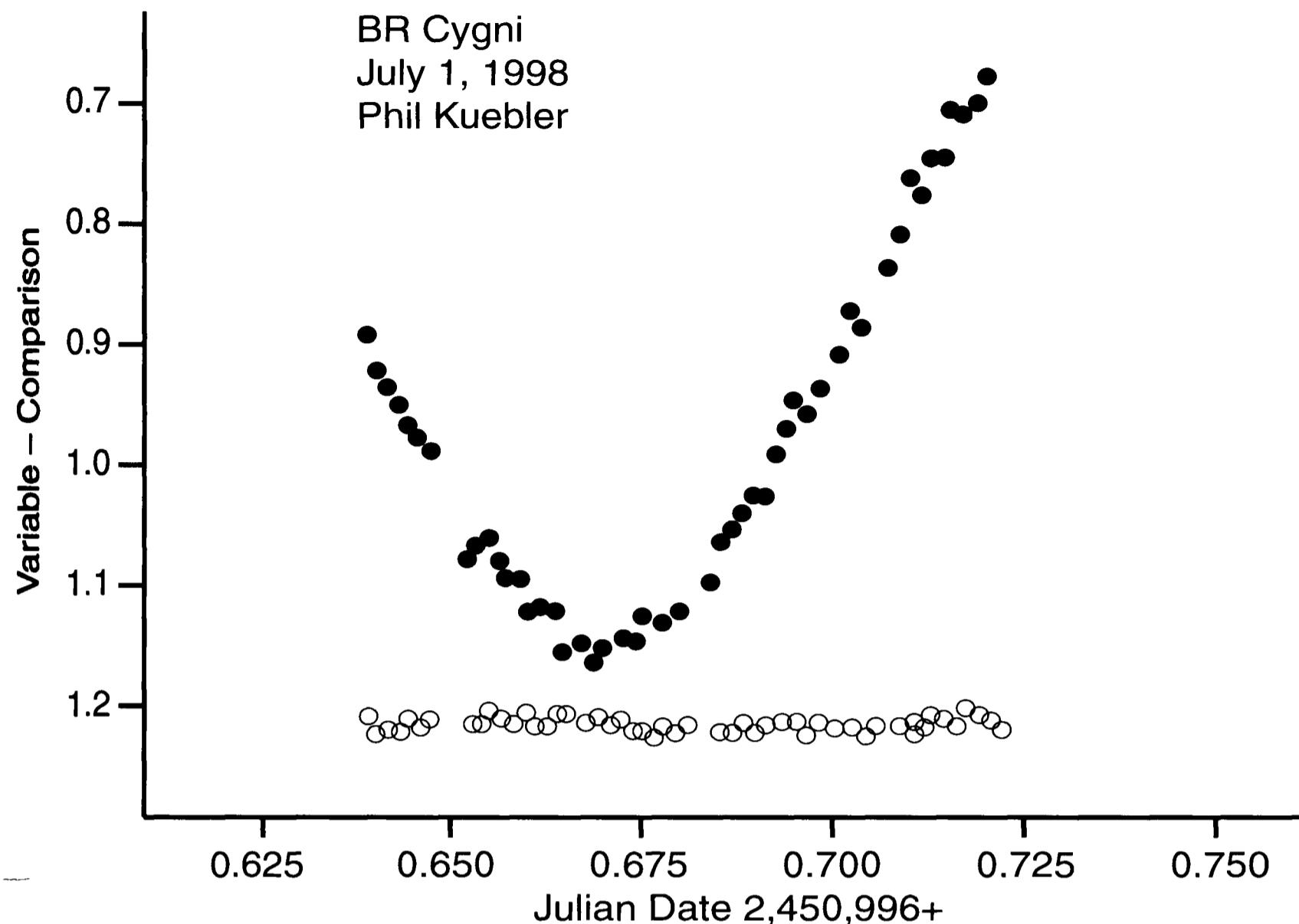
called “C2,” or the “check” star. The purpose of the check star is to verify that the C1 comparison star does not vary. A time series of images can run for many hours and contain hundreds of images.

When reduced, the observation is the magnitude difference between the variable and comparison star, usually written as  $V - C1$  (variable minus comparison). To monitor that nothing has gone awry, the difference between C1 (the comparison star) and C2 (the check star) is also extracted as  $C2 - C12$ .

For greater precision, observers sometime employ more than two comparison stars. By summing multiple comparison stars, they create an aggregate comparison star with a large photon count and reduced statistical error.

Below are profiles of a few typical observing programs.

**Eclipsing Binary Stars.** When the angle between the orbital plane of a binary star and our line of sight is small, we observe eclipses with each orbital revolution. Periods of close binaries range from a few hours to several days. If the period is constant, we can predict the time of eclipse years in advance; but if the stars are interacting and gas flows from one to the other, the period can change. Measuring the time of mid-eclipse provides a very sensitive tool for probing the physical nature of the stars; hence, professional astronomers have a continuing need for measured times of mid-eclipse. Requests for observations of particular



**Figure 10.8** This light curve shows the eclipsing binary BR Cygni at mid-eclipse, as measured from 40 images like the one in Figure 10.7. Filled circles are  $V - C_1$  and open circles are  $C_1 - C_2$ . Phil measured the time of minimum as  $JD, 2,450,996.6709 \pm 0.0003$ . Image and data courtesy of Phil Kuebler.

stars are most often channeled through organizations such as the AAVSO and the Center for Backyard Astrophysics.

The observing protocol is simple: as a clear evening comes up, the observer scans a list of program stars and computes their expected times of minimum. If a program star is expected to eclipse that night, the observing run is scheduled to begin an hour or two before the anticipated time of minimum. Before and after the observing run itself, the observer allows time to make dark frames and flat fields.

The observing run consists of making images at regular intervals (usually one or two minutes) through the time of eclipse and for an hour or two afterward. If the camera software has an “autograb” or “multiple image” feature and the telescope tracks well, the observer has little to do but oversee that everything moves along smoothly. A second set of dark frames and flat fields is then taken as a hedge against changes in the camera or telescope.

To reduce the data, the images are calibrated and the  $V$ ,  $C_1$ , and  $C_2$  stars are measured on each one. If the time-of-minimum prediction was good, the eclipse is obvious in the plot of  $V - C_1$  against time. If the equipment functioned properly, the corresponding plot of  $C_2 - C_1$  is flat and straight. To extract as much information as possible from the light curve, the data can be analyzed statistically to obtain the best-fit time of minimum, often within less than one minute. See Figure 10.8.

## Chapter 10: Photometry

The observed time of minimum is then reported to the AAVSO or to the astronomer who requested observations of the star.

**Exoplanet Transits.** Planets orbiting stars other than the Sun can be discovered and their properties determined by observing the decrease in the star's light when the planet transits its disk. Although a signal-to-noise ratio of 100 or better is required, this is well within the ability of many CCD observers.

Preparation for an exoplanet transit observation is much like that for an eclipsing variable star. Check the ephemeris for future transits, and begin a time series several hours in advance. Continue observing through the predicted transit time, and continue as long as practical after the transit. The "baseline" magnitude established before and after the transit enables you to distinguish the slight decrease in magnitude from variations caused by statistical uncertainty and other sources of noise.

If you are serious about exoplanet work, join a group that regularly observes and reports on newly discovered planets. Amateur observations help to establish the reality of the transit, aid in pinning down its period and amplitude, and ultimately, in documenting the characteristics of the planet.

**Other Variable Stars.** To obtain good light curves for variables with periods of a few days to a year or more, regular observations are needed. These may be Cepheid variables (which sometimes do strange things—like Polaris, a Cepheid that nearly quit varying), RR Lyrae stars, RS Canum Venaticorum stars (which vary because of enormous starspots on their surfaces), or many other types. The AAVSO maintains lists of interesting variables that need to be observed.

With a clear evening in the offing, the observer makes up a list of program stars. For extremely slow variables, only one observation per month may be needed, but for more rapid ones, an hourly check may be called for. The run itself depends on what's on the schedule. For slowly-changing stars, an observation might consist of three 60-second integration images through each of three filters (such as B, V, and I) and take a total of 15 minutes. If the program consists of a dozen such stars, the observer could begin at dusk and complete precise three-color differential photometry on 12 objects by midnight.

When time permits, the images are calibrated and  $V - C_1$  and  $C_2 - C_1$  are measured on each one. Since everyone working on this particular variable uses the same comparison star, the  $V - C_1$  measures from different observers mean the same thing. Of course, you check the  $C_2 - C_1$  values for consistency from one observing session to the next.

At the end of the month, observer sends a report to AAVSO headquarters listing high-quality three-color differential measurements on 16 stars made during five observing runs.

### 10.4.3 All-Sky Photometry

A night of all-sky photometry requires orchestrating three simultaneous observing programs: finding extinction coefficients, finding transformation coefficients, and

## Section 10.4: Photometric Observing

observing program stars. This makes it challenging, to say the least, but also an skill well worth pursuing.

To illustrate what's involved in this method, suppose you have been asked to assist with two new comparison-star charts for use by the observers who contribute visual estimates to a well-known variable-star organization. Each chart is 15 minutes of arc square and centered on a variable. Suitable comparison stars are marked with their visual magnitudes. Observers estimate the magnitude of the variable by comparing its brightness with the comparison stars. If the magnitudes of the comparison stars are not accurate, the estimates will not be correct. Your photometry must be as accurate as possible.

To make sure that the comparison stars will serve their purpose, you need to observe each candidate in B, V, and I to insure that no red stars (which tend not only to be variable, but also to fool the eyes of visual observers) are included among them.

You begin planning for the observation by figuring out when the chart fields will be highest in the sky, and you identify two Landolt Selected Areas—one that will cross the meridian an hour before and the other an hour after the chart fields. Fortunately, you have three hours of darkness before the chart fields transit, so there is ample time to obtain extinction and standard-star images. This adds to your program a third Landolt field that will transit shortly after full darkness falls.

After rejecting several nights for streaks of high cirrus after sunset, you rejoice when a big front clears the dust and haze from the lower air and leaves the whole sky crystal clear. At last you have your “photometric” night! An hour before darkness, you have set the computer’s clock, fired up the CCD, and opened the telescope to cool.

During twilight, shoot your bias, dark, and flat-field frames. Then, as twilight ends, you’ll be ready to identify two Landolt areas that are rising. Make two 120-second integrations of each field through each of the three filters. Between each exposure move the telescope slightly so the star field falls on slightly different groups of pixels. As the evening progresses, you must come back to observe these fields as they rise and are viewed through progressively less air mass. Also, shoot some Landolt areas close to the meridian. In this way you can piggyback the exposures needed for extinction data while simultaneously making standard star fields.

With the high-air-mass exposures taken, you locate the Landolt area that is near the meridian, carefully focus, and repeat the exposure sequence. As the night progresses, you will make more images of this region as it sets into progressively greater air mass. It doesn’t hurt to check that extinction coefficients for the eastern and western sky are the same.

At this point, you cap the telescope and take 16 bias frames and 16 dark frames; then rig the light box on the telescope and take 16 flats through each of the three filters, and take 16 flat darks. Good calibration is vital to accurate photometry, but it certainly helps to fill up the hard disk on the computer! The bias and flat-frame exposures are short, but a set of dark frames can take half an hour.

## Chapter 10: Photometry

If your telescope is light-tight, make dark frames during twilight.

With calibration behind you, you concentrate on imaging the two rising Landolt fields through the three filters. You want plenty of data points to extract extinction coefficients and transform the magnitudes to the standard system.

As the chart fields approach the meridian, and you locate and make three exposures with each filter for each chart field, shoot the Landolt areas, and then shoot the chart fields again. Finally, shoot another set of images of the third Landolt field, which has sunk low in the western sky.

In theory, you now have all the data that you need—but as a conscientious observer, you image the two Landolt fields again and make a backup set of calibration images. By the time you close up for the night, you have gathered a very complete set of data. Of course, if you have time, you'll make another set of observations on another night as proof against zero-point errors, and as a check that none of the new standards is itself a variable!

Extracting magnitudes is a time-consuming but satisfying activity. You begin by computing the air mass for every exposure made during the night. Next, to be sure that you can identify the correct stars in each of the Landolt fields, print a negative hard-copy of each field and verify the identification of each standard star astrometrically. You have selected eight stars in each Landolt field as your standards, but it only takes a few extra minutes to make sure that no errors will creep into your work.

Now examine each image and perform aperture photometry on each standard star. As you measure, transfer the data—the raw instrumental magnitude, the air mass, and Landolt's standard magnitude—to a spreadsheet program. Since you obtained five images in each of three colors for the rising Landolt fields, and four images in each of three colors for the setting Landolt field, you have 42 images to measure. When you are done, the spreadsheet solves for extinction coefficients in each color and produces transform coefficients that convert your raw instrumental magnitudes and air masses into standard *B*, *V*, and *I* magnitudes. Be glad you don't have to compute these values by hand!

Now comes the fun part: examine the images of the chart fields, two sets of three exposures in three colors for each of the charts—18 images in all. Again, measure raw instrumental magnitudes for each star; then enter those and the air mass for the image into spreadsheet software. This time, the output is a standard *B*, *V*, and *I* magnitude for each star in the chart field. To your pleasure, the results from the separate images agree to better than 0.02 magnitude in every case. The result of your work is standard *B*, *V*, and *I* magnitudes for 46 candidate comparison stars in the two charts. As you submit your report, you know you have done a good job.

### 10.4.4 “Do-What-You-Can” Photometry

Most amateur astronomers live and observe at humid, low altitude sites. They can rarely observe more than a few hours simply because it usually doesn't stay clear any longer than that, or because they have to go to work the next morning. Con-

## Section 10.4: Photometric Observing

ventional all-sky photometry appears too daunting, quite apart from the complexity of performing data reduction, so no photometry gets done.

A realistic solution to this dilemma is to adopt a mode of observing that makes some approximations; that is, observing in “do-what-you-can” or “what-you-can-get-away-with” mode. As long as you don’t feel compelled to press for high accuracy, you can rest assured that your results are pretty good. Very importantly, observations can be made quickly in this mode, so you *will* make the observations.

To begin with, you need only two filters: *V* and *I*. This allows you to measure and correct for the color terms. The *V* filter ties into the *V* and visual magnitudes. Also, the  $V - I$  color is an excellent temperature indicator for all stars, and it even works for the reddest stars. This field is challenging enough in its own right that trying to do photometry in four colors is bound to be discouraging. Instead, do what you can do, which is accurate two-color photometry.

At some point early on in the process of getting data from a CCD, would-be photometrists should spend several stable photometric nights—the kind of nights that occur only once or twice a month—observing nothing but bona-fide standard stars. The idea is to establish a good set of instrumental transformations and to get a feeling for the extinction values at your locale. Once you have established the transformations, check them once or twice a year to detect changes or problems.

In an ideal world, every observer would measure the extinction coefficients **every night**. However, if you observe from a site below 1,500 feet elevation, you can simply assume a *V* coefficient of 0.35, except on winter nights when the air is particularly dry and clear (when you can assume a *V* coefficient of 0.25). Extinctions for colors other than *V* are offsets; for *B*, the extinction coefficient is 0.13 higher than *V*; for *R*, it is 0.04 lower than *V*, and for *I*, it is 0.08 lower than *V*.

With the instrumental transformations established, the main thing needed for new observations is the zero point. This varies mainly because of night-to-night changes in extinction and, to a lesser degree, instrumental variations. A simple procedure to set the zero point is to shoot one or two standard fields, then shoot the target, and reshoot the standard field. That’s it! Your standard field can contain a single star if need be, but it is usually not hard to find a good sequence near the target.

To reduce your observations, apply the mean extinction value and the previously determined color terms to the data. This will produce nearly-correct standard magnitudes for the reference stars. Then compare these magnitudes with “real” magnitudes of the standards, and apply any difference to the magnitude and color of the target. As long as you do not observe targets at high air mass, using approximate extinction coefficients will not compromise your data, because the standard stars are close to the target, and errors in extinction show up only when there is a difference in air mass between the standard fields and your targets.

“Do-what-you-can” photometry has the merit of being quick, which is a big advantage at a poor site. A set of observations in two colors shouldn’t take more than 10 to 20 minutes, depending on how faint the target is and how well the tele-

## Chapter 10: Photometry

scope points. The speed of working minimizes transparency variations, and the method does away with making tedious extinction observations.

### 10.5 *Desiderata* for Photometry

The requirements for photometric images are really not much different than for any high-quality CCD image: expose correctly, calibrate properly, shoot high in the sky, and do everything you can to insure a high signal-to-noise ratio. The major differences are that you need to make your images in a well-defined part of the spectrum using colored glass filters, and that star images need to be large enough to insure proper sampling.

**Correct Exposure.** The integration time must be long enough to obtain a good signal-to-noise ratio for objects of interest, but cannot be outside the linear portion of the CCD's response curve. For most of the sensors used by amateur astronomers, this means that the peak value in star images of interest should not exceed one-half the full-well capacity of the CCD. Blooming is *absolutely* not allowed in any star of interest. In addition, any camera options that lead to non-linearity (such as an anti-blooming gate) should be shut off or disabled. Taking multiple images and combining them to obtain a better signal-to-noise ratio is acceptable.

**Proper Calibration.** The images must be calibrated properly, preferably using the advanced calibration protocol (bias removal, dark current subtraction, and flat-fielding; see Section 6.3.3), with a separate set of flat-fields for each filter in the photometric set. The flat-fields should be exposed to one-half of the full-well capacity; and the observer should combine multiple flats, so the signal-to-noise ratio of the master flat for each color is considerably higher than the signal-to-noise ratio expected in the image. For photometry to 0.01 magnitude (1%), the signal-to-noise ratio of the flat-fields should be 500:1 or better.

**Minimize Air Mass.** Shoot photometric images as high in the sky as possible to reduce the effects of extinction. For all-sky photometry, this means working higher than two air masses ( $30^\circ$  elevation) on nights when the sky is free of any trace of clouds. For differential photometry, thin passing clouds are usually acceptable high in the sky; but near the horizon, where cloud motion is slower and extinction is greater, any cloud is unacceptable.

**Insure Proper Sampling.** Front-surface CCDs (i.e., the inexpensive kind used by amateur astronomers) have a polysilicon gate structure that starlight must pass through to reach the light-sensitive bulk silicon of the chip. The gate structure is not present everywhere on the face of the CCD, but is laid down in strips. If a star image is only one or two pixels across, and it happens to fall on a strip, a significant fraction of the incidental starlight can be lost. If the star images are four or more pixels in diameter, the loss is averaged over multiple pixels and becomes unimportant. With back-illuminated CCDs (i.e., no gate structure), star images can be as small as two pixels in diameter.

As a rule of thumb, the optimum image scale is about two seconds of arc per

## Section 10.6: Don't Be Afraid to Try!

pixel, but amateurs have done excellent photometry with scales between one and six seconds of arc per pixel. The primary concern is that the star image must cover at least two pixels at full-width half-maximum. If the focal length of your telescope is so short that your star images are only two or three pixels across, set the CCD very slightly out of focus. Enlarged star images don't look nice, but they produce reliable photometric results. An additional benefit from defocusing is that the peak pixel values in "mushy" star images are lower than they are in sharply focused ones; therefore, the camera is more likely to be working in the linear portion of the CCD's response curve.

**Shoot Multiple Images.** For accurate photometry, why rely on a single image when you can shoot two or even three exposures? Offset the images slightly from one another so the star images fall on different groups of pixels, reduce the images individually, and then check that the magnitudes in the different frames agree with one another. If your darks and flats are good, they will. In sequences of images taken for differential photometry of eclipsing binary stars or rotating asteroids, defective images stand out when the difference between the comparison and check stars falls well outside the normal range of statistical variation.

**Shoot through the Right Filters.** Unfiltered CCD images have limited value in photometry because too many factors influence the response of the CCD. However, a full UBV(RI) filter set costs several hundred dollars. To keep costs down, order a V filter so that your instrumental measures correspond to visual magnitudes. You can always add other filters later. Few amateurs bother with a U filter because front-illuminated CCDs have so little sensitivity in the U passband that photometry is all but impossible; and with many CCDs, B filters are hardly practical. For many CCD observing programs, a V filter and an I filter may be all you'll ever need. In any case, consult with the organization that you observe with, and use the filters that they recommend.

Some observing programs do not require filters. An example is monitoring cataclysmic variable stars for the Center for Backyard Astrophysics, where the goal is to get good time coverage of these rapidly changing variables.

## 10.6 Don't Be Afraid to Try!

By their very nature, photometrists strive to push their art to the limit, right into the millimagnitude range ( $\frac{1}{1000}$  of a magnitude) if they possibly can. As a result, and without intending to, they often make photometry sound both scary and boring. It is not. Even if you shoot unfiltered images and don't make flat frames, you can do useful differential photometry of short-period variable stars or eclipsing binaries that will probably be good to 0.05 magnitudes—about four times better than a skilled observer can estimate by eye.

With a little care in shooting flat-fields, you can improve your accuracy to 0.015 magnitude, which is good enough to make light curves of stars with star-spots, observe the 6-day rotational light curve of Pluto and any number of other interesting observations. Add a single standard photometric filter to the mix and

## Chapter 10: Photometry

join a group like the CBA, AAVSO, BAAVSS, or an ad hoc team from VSNET, and you'll tap into a lifetime of fascinating (and real) observing challenges.

# 11 Spectroscopy

In this chapter you will learn the basics of spectroscopy, in which light is separated into its constituent wavelengths, enabling astronomers to deduce the physical conditions of the stars, nebulae, and galaxies that emit the light. Although spectroscopy has long been one of the primary tools of professional astronomy, amateurs have been slow to follow. With the emergence of electronic cameras, spectroscopy—and astrophysics—are now open to the amateur.

## 11.1 What is Spectroscopy?

The application of the spectroscope to celestial bodies marked the birth of modern astronomy and astrophysics. Because of its great brilliance, the Sun was the first astronomical source observed spectroscopically. In 1802, William Wollaston observed the solar spectrum using a narrow slit and discovered that the continuous flow of spectral colors was broken by a few narrow dark lines. However, it fell to Joseph Fraunhofer, a Bavarian physicist and lensmaker, to construct the first spectroscope with enough resolving power to reveal about 600 of the thousands of narrow dark lines (now known as Fraunhofer lines) that cross the spectrum.

The significance of the lines was not fully revealed until 1859, when Robert Bunsen and Gustav Kirchhoff announced their discovery that the dark lines in the solar spectrum could be duplicated in the laboratory. The most dramatic example was the bright yellow line of sodium—easily produced by placing a pinch of ordinary table salt (sodium chloride) in the flame of a Bunsen burner—that matched the strong dark line in the solar spectrum. Kirchhoff found that a glowing vapor not only gave off lines characteristic of the elements in the vapor, but would absorb the same wavelengths from the spectrum of a white-hot body passing through the vapor. The spectral lines of other elements—cesium, strontium, rubidium, magnesium, hydrogen—also matched those in the Sun.

The Sun's dark lines were, in fact, atomic “fingerprints” of the elements in its atmosphere. During the solar eclipse of 1868, astronomers were mystified by two strong emissions that did not match any known terrestrial elements—but they gave the name “helium” to this element that appeared to be present only on the Sun. Not until 1895 was it found on Earth.

An important key to extending the scope of element identification was pre-

## Chapter 11: Spectroscopy

cise measurement of the wavelengths of the lines. In 1868, a Swedish professor of physics, Anders Ångström, published a catalog listing accurate wavelengths of 1200 Fraunhofer lines, of which 800 coincided with lines of terrestrial elements. Ångström used a diffraction grating rather than prisms to create the high dispersion that made his wavelength determination so precise. By the end of the 1880s, over 50 elements would be identified in the solar spectrum.

Pioneering researchers such as William Huggins were not long in applying spectroscopy to the stars. Beginning in 1860, Huggins set about a systematic examination of the spectra of planets, bright stars, and nebulae. In 1864, he found that the spectrum of NGC 6543, a bright planetary nebula in Draco, consisted of spectral lines. Huggins wrote, “The riddle of the nebulae was solved. The answer, which had come to us in the light itself, read: Not an aggregation of stars, but a luminous gas.”

By the end of the 1870s, astronomers had investigated the spectra of the planets, stars, comets, novae, galaxies, and nebulae; and they were well settled on the course that led to the “new astronomy”—astrophysics—at the turn of the century. Spectroscopy remained the province of the professional astronomer throughout the twentieth century.

The situation has changed with the advent of electronic cameras. Spectroscopy is now practical for everyone. Pioneering British amateur Maurice Gavin made excellent use of an objective prism spectrograph (i.e., placing a 9-inch prism in front of a 10-inch  $f/4$  Newtonian equipped with a Starlight Xpress CCD camera), and Texas amateur Don Davis constructed a compact spectrograph for his 17.5-inch  $f/4.5$  Newtonian and Cookbook CCD camera. With the introduction of a telescope-mounted imaging spectrograph by SBIG, and the compact optical-fiber fed model by Sivo Scientific, this important but neglected branch of astronomy will almost certainly become more popular among non-professionals.

## 11.2 Spectra and Spectrographs

Spectroscopy requires separating light of different wavelengths; yet in a telescope, (theoretically at least) light of all wavelengths comes to focus at the same point. Spectroscopy, therefore, requires an auxiliary optical element or instrument to disperse the light so that different wavelengths fall on different places on the detector. Such instruments are called *spectroscopes* when used visually and *spectrographs* when the spectrum is recorded on photographic film or with an electronic camera.

### 11.2.1 Prisms and Gratings

The key part of a spectroscope or spectrograph is its dispersing element, an optical part that splits light into its component wavelengths. The classic dispersing element is a glass prism which works because different wavelengths are refracted differently and exit the prism at slightly different angles. Although prisms are quite efficient in the sense that virtually all the light that enters a prism exits it, their dispersion is *irrational*; that is, the refracted angle is a strongly nonlinear function of

## Section 11.2: Spectra and Spectrographs

wavelength. Light at the blue end of the spectrum is dispersed about three times more than that at the red end.

The other common dispersing element is the diffraction grating. A grating consists of fine grooves ruled very close together on a glass or metal substrate—often thousands per inch. In the wave model of light, at any given wavelength new peaks appear by constructive interference at well-defined angles from the original beam. Because at longer wavelengths the angles are larger, the diffracted light is arranged in an orderly spectrum. By carefully shaping and spacing the grooves, about 60% of the light falling on a grating can be focused into a spectrum. An important property of gratings is that the angle of dispersion is almost linear with wavelength; but a disadvantage is that they generate different spectral orders, so that if the range of wavelengths is large, gratings require auxiliary filters or prisms.

They come in two types: reflection and transmission. Reflection gratings appear mirror-like, and like a mirror, reflect the diffracted light in the direction it came from. Because of the fine grooves on their surfaces, music CDs act like diffraction gratings. Although reflection gratings are often flat, they can also be formed on a concave surface, so that a single optical element can both focus and disperse light to form a spectrum. Transmission gratings allow the bulk of the light to pass through and form a spectrum behind the grating.

The dispersing element of a spectrograph spreads a single ray of light into a spectrum. Classically, the spectrum refers to the varicolored smear of light with blue at one end and red at the other. Today, the term has evolved to mean any image, graph, or plot of flux versus wavelength. The primary task of spectroscopic image processing is to extract spectrum data from the image. Further analysis is usually required to create the spectrum from the extracted data.

In the visible part of the spectrum, the accepted unit of wavelength is the nanometer ( $10^{-9}$  meters). Although it's no longer officially recognized, many spectroscopists continue to employ the angstrom ( $\text{\AA}$ ) of  $10^{-10}$  meters as a unit of wavelength. The conversion is  $10 \text{ \AA} = 1 \text{ nanometer (nm)}$ .

### 11.2.2 Practical Spectrographs

To be practical for an amateur astronomer, a spectrograph must be reasonably simple to construct, reasonably robust in operation, and reasonably inexpensive. Of course, “reasonable” is wide open to interpretation. It depends on the skills, finances, and available time of the individual observer. Of the enormous range of spectrographs, however, four designs stand out: objective prism, grating-prism, imaging slit, and fiber-fed.

### 11.2.3 The Objective Prism Spectrograph

The easiest way to create a spectrum is to place a prism in front of a telescope. The light is dispersed even before it enters the telescope, which then brings the dispersed light to focus as a spectrum. Professional astronomers use objective prisms

## Chapter 11: Spectroscopy

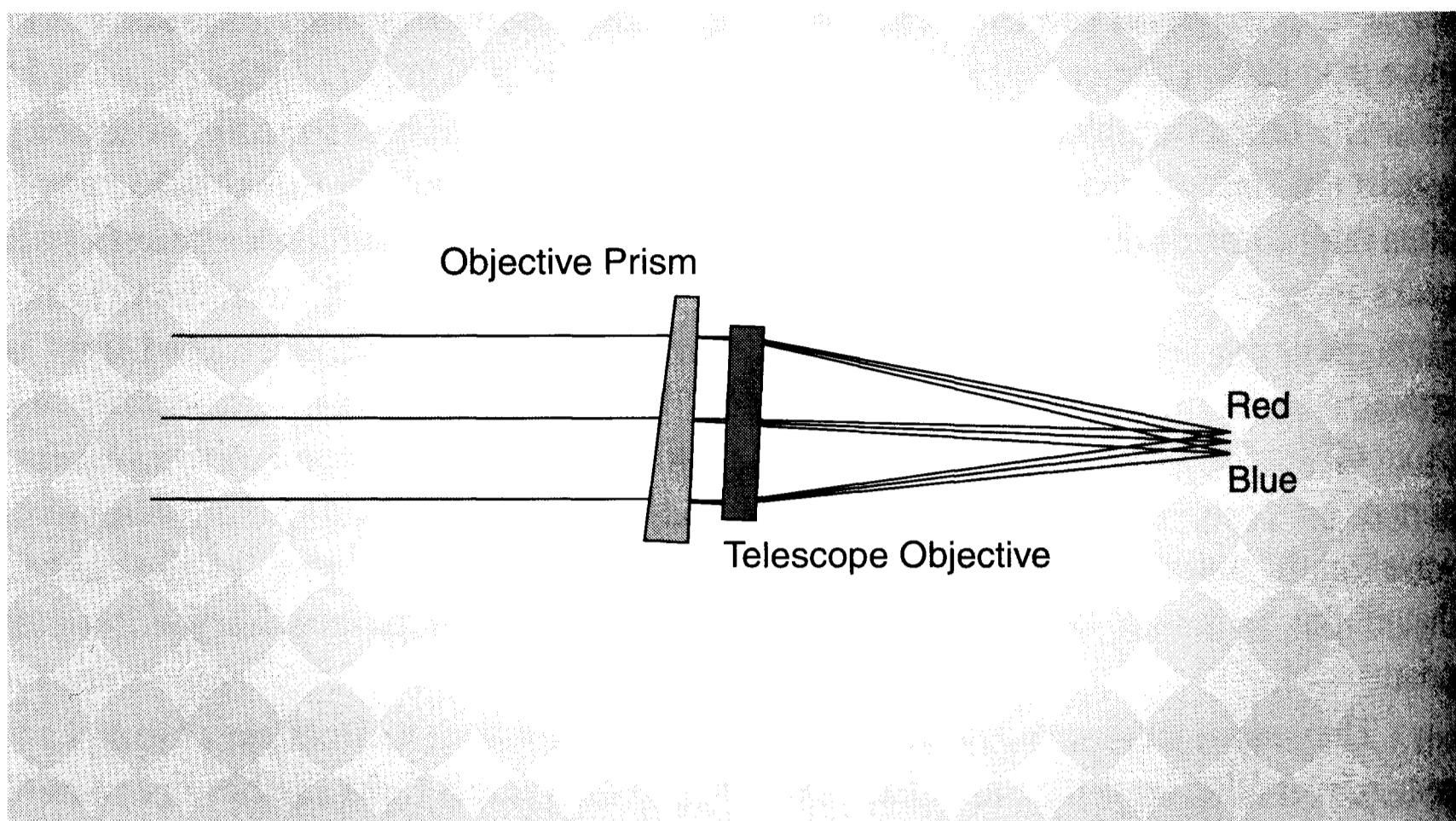


Figure 11.1 A prism with a small apex angle spreads the light of every object in the field of view of a telescope into a spectrum. Although the resulting spectra are contaminated with the background light of the night sky, objective prisms are the easiest way to make low-dispersion spectra of stars.

to survey and catalog large numbers of stars rapidly, since with an objective prism, every star in the image appears as a spectrum.

Unfortunately, the prism must be approximately as large as the telescope objective, made of high-quality glass, and polished accurately flat on both faces. Even the glass for such a prism is expensive, and finished prisms are extremely so. However, making a prism with a 4-inch to 8-inch aperture and a  $5^\circ$  vertex angle is within the ability of a determined amateur optician.

Objective prisms work because the refractive index of glass varies with wavelength. Parallel rays of light of different wavelengths entering the prism will exit at different angles. To calculate the angles, use Snell's Law:

$$\frac{\sin \epsilon}{\sin \epsilon'} = \frac{n'}{n} \quad (\text{Eq. 11.1})$$

where  $n$  and  $n'$  are the respective refractive indices of the glass and the surrounding air, and the angles  $\epsilon$  and  $\epsilon'$  are measured relative to a line perpendicular to the glass surface. Consider the following example of a spectrograph that an amateur could construct: an objective prism with a vertex angle of  $5^\circ$  made of high-quality but inexpensive BK 7 optical glass. For a prism 200 mm in diameter, the glass would be 10 mm thick at one side and 27 mm thick at the other. This prism would deviate a ray of green light by  $2^\circ 36'$ , meaning that the telescope would have to be pointed this amount away from an object for it to appear in the center of the field. Rays at other wavelengths would undergo slightly different deviations (see Table 11.1), so that the angular spread between violet and deep red would be 5 minutes of arc. Although this seems like a very small difference, at the focus of a telescope

**Table 11.1 Ray Paths in an Objective Prism**

Wavelength (nm)	Refractive Index (BK7 glass)	Deviation Angle (degrees)
707	1.51289	2° 34' 36"
656	1.51432	2° 35' 03"
588	1.51680	2° 35' 48"
546	1.51872	2° 36' 23"
486	1.52238	2° 37' 29"
436	1.52669	2° 38' 47"
405	1.53024	2° 39' 52"

with a focal length of 2000 mm, the 5-arcminute spectrum is roughly 3 mm long and spans 340 pixels on a CCD with 9-micron pixels. Thus, the very weak objective prism combined with a telescope creates an efficient and practical low-resolution instrument for stellar spectroscopy.

Images taken with an objective prism contain the spectrum of every star in the field of view. Some of the spectra run off the edges of the field, and some will undoubtedly overlap each other. The advantage of the objective prism is that no modification to the telescope is necessary except to mount the prism over the front. The telescope can be any type—refractor, reflector, or catadioptric.

The main disadvantage of this form of spectrograph is that the starlight is superimposed on the full intensity of the night sky (see Figure 11.5). An additional disadvantage is that you cannot measure wavelengths directly from the spectrum because its location, and hence the position of the spectral features in it, depends on the location of the star in the field.

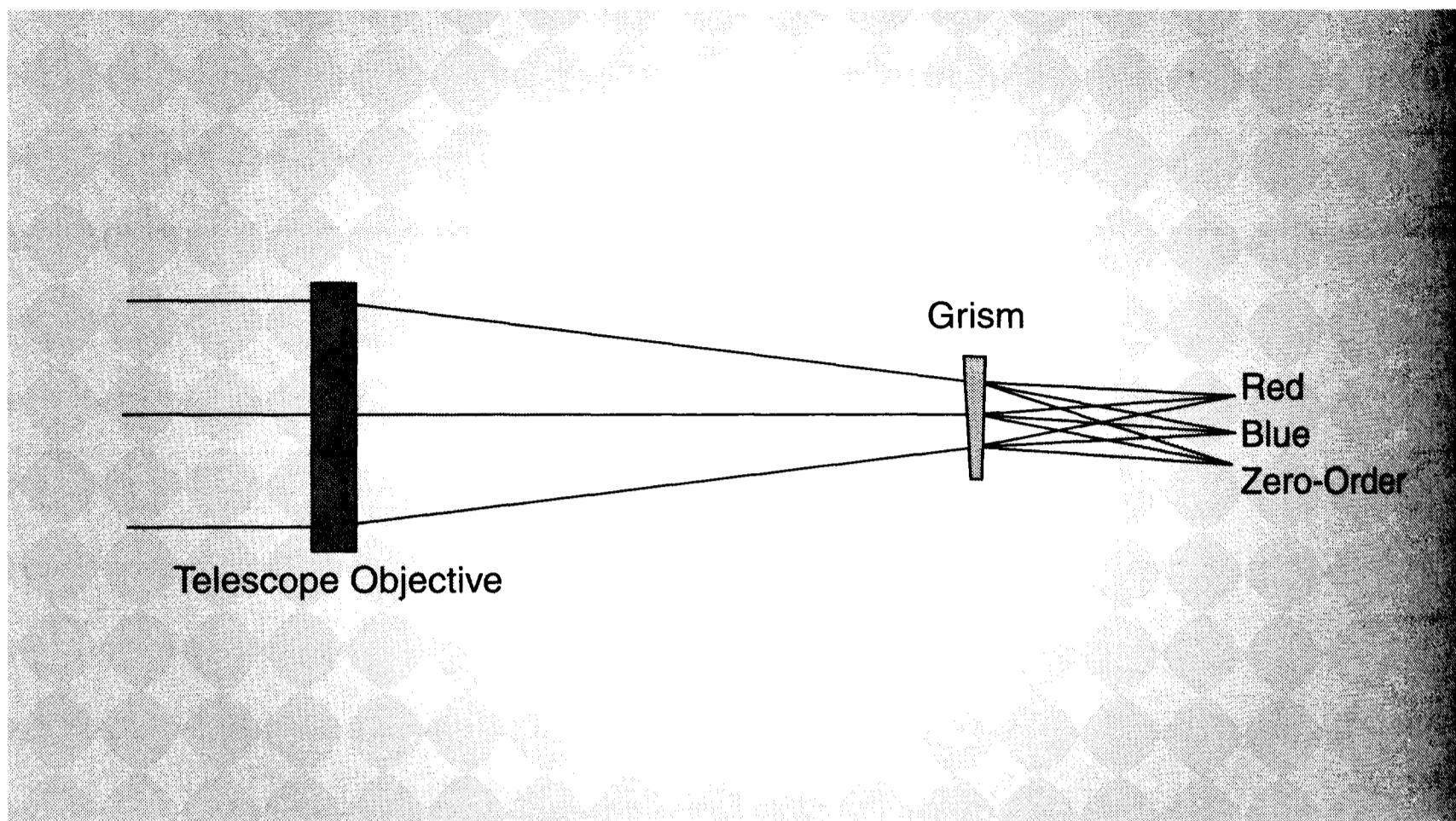
#### 11.2.4 The Grating-Prism Spectrograph

A grating or grating-prism spectrograph is perhaps the most practical way for an amateur to get into spectroscopy. A grating-prism, or *grism*, is a transmission grating combined with a prism. The grating or grism is placed in the beam of light converging toward focus, usually a few inches ahead of focus. The grating disperses the light into a spectrum. In a grism, the prism refracts the beam so that the spectrum is formed directly behind the grism, and in so doing, corrects some of the optical aberrations introduced by the grating.

Grating and grism spectrographs are well suited to the needs of amateur astronomers because the grism optical element is both small and relatively inexpensive (several hundred dollars). Grisms lend themselves equally well to reflectors, refractors, and catadioptric telescopes. Furthermore, it is possible to shift a grism in and out of the converging beam without making other changes to the optical configuration.

Transmission diffraction gratings consist of a thin glass substrate that supports a transparent film ruled with several hundred evenly spaced parallel grooves

## Chapter 11: Spectroscopy



**Figure 11.2** Grism spectrographs disperse light by means of a small optical element, a grating or grism (a combination of a grating and a prism), just ahead of focus. Because the dispersing element is small and close to focus, spectroscopy with gratings and grisms is relatively easy and inexpensive.

per millimeter. Light passing through the grating is dispersed into a spectrum, with the angle of deviation dependant on its wavelength. The grating equation is:

$$\sin i + \sin \delta = mn\lambda \quad (\text{Eq. 11.2})$$

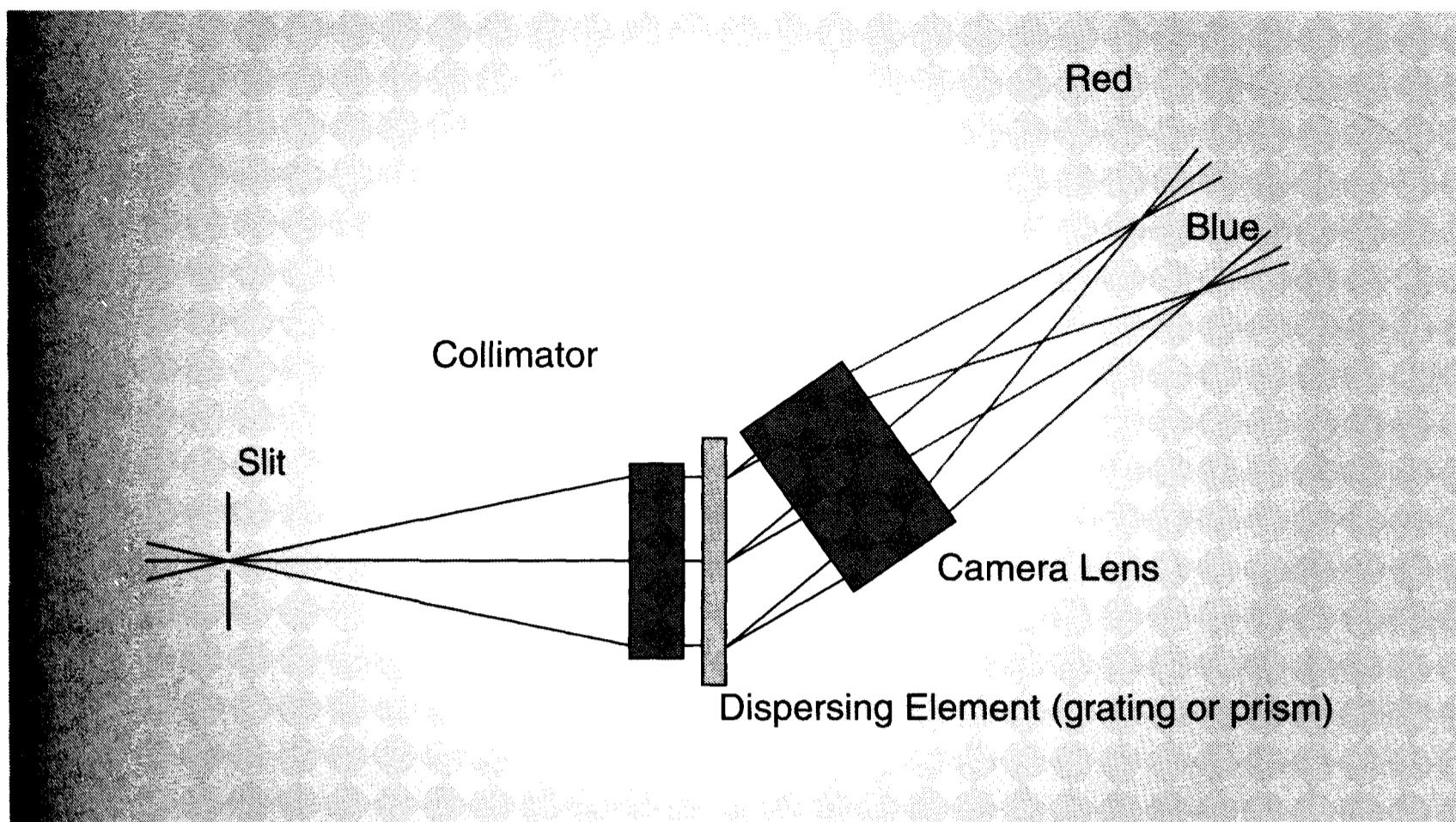
where  $i$  is the angle of the incident light,  $\delta$  is the angle of the diffracted light,  $m$  is the spectral order (which can be 0, 1, 2, or more, but usually equals 1),  $n$  is the number of grooves per millimeter, and  $\lambda$  is the wavelength in millimeters. (The angles are measured relative to a line normal to the grating surface.) If the incident light strikes the grating perpendicularly so that  $i$  is zero, the diffracted rays are deviated by the angle  $\delta$  relative to their original paths.

The grooves in modern gratings are shaped, or “blazed,” so that they diffract approximately 60% of the incident light into the first spectral order; that is, at an angle corresponding to  $m = 1$  in the grating equation. However, some light—the zero-order spectrum—passes through the grating without being deviated. The zero-order image provides a fiducial marker for measuring wavelengths.

The distance between the grating and the detector depends on the angular dispersion of the grating, the size of the CCD, and the type of spectra you wish to obtain. From the grating equation, you can calculate the deviation angle,  $\delta$ , using a grating with 200 grooves per millimeter, for deep blue light with a wavelength of 400 nm ( $400 \times 10^{-6}$  millimeters):

$$\sin \delta = 1 \times 200 \times 400 \times 10^{-6}.$$

For deep blue, the first-order deviation angle is 4.6 degrees. Repeating the calculation for near-infrared light at 900 nm, the deviation is 10.3 degrees.



**Figure 11.3** Throughout the twentieth century, professional astronomers have obtained spectra with slit spectrographs mounted directly on their telescopes. These instruments are ideal for making medium- to high-dispersion spectra with precisely calibrated wavelengths.

In a grism spectrograph, a prism is placed immediately in front of the grating. The vertex angle of the prism is chosen to refract light so that a wavelength in the middle of the spectrum exits the grating parallel to its original path. A prism chosen to pass light of 650 nm wavelength on a straight path gives an angular dispersion of 5.7 degrees between 400 nm and 900 nm.

We can find the length of the spectrum on the detector by multiplying the distance between the grism and the CCD chip by  $\tan(5.7^\circ) = 0.099$ . If the distance between the grism and the CCD is 50 mm, the spectrum will be 5 mm long. The separation should be chosen so that the zero-order image and the first-order spectrum fit comfortably into the image.

Because of aberrations introduced by the grating, the zero-order image and the spectrum do not lie at exactly the same focus. If you make grating or grism spectra, be sure to focus on the spectrum rather than on the zero-order image.

### 11.2.5 The Slit Spectrograph

The slit spectrograph is an optical instrument designed to isolate a narrow strip of light from the focus of a telescope by means of a slit, pass it through a dispersing element, and reimagine a wavelength-dispersed image of the slit onto a detector. The two-dimensional spectrum recorded by the detector consists of a thin slice of sky in one axis and a sequence of images of the slit at different wavelengths spread along the other axis.

Textbook illustrations necessarily present a simple picture of spectrograph optics; the reality is that they require sophisticated design. In addition to the basic function of dispersing light, the spectrograph must form sharp images of the slit at

## Chapter 11: Spectroscopy

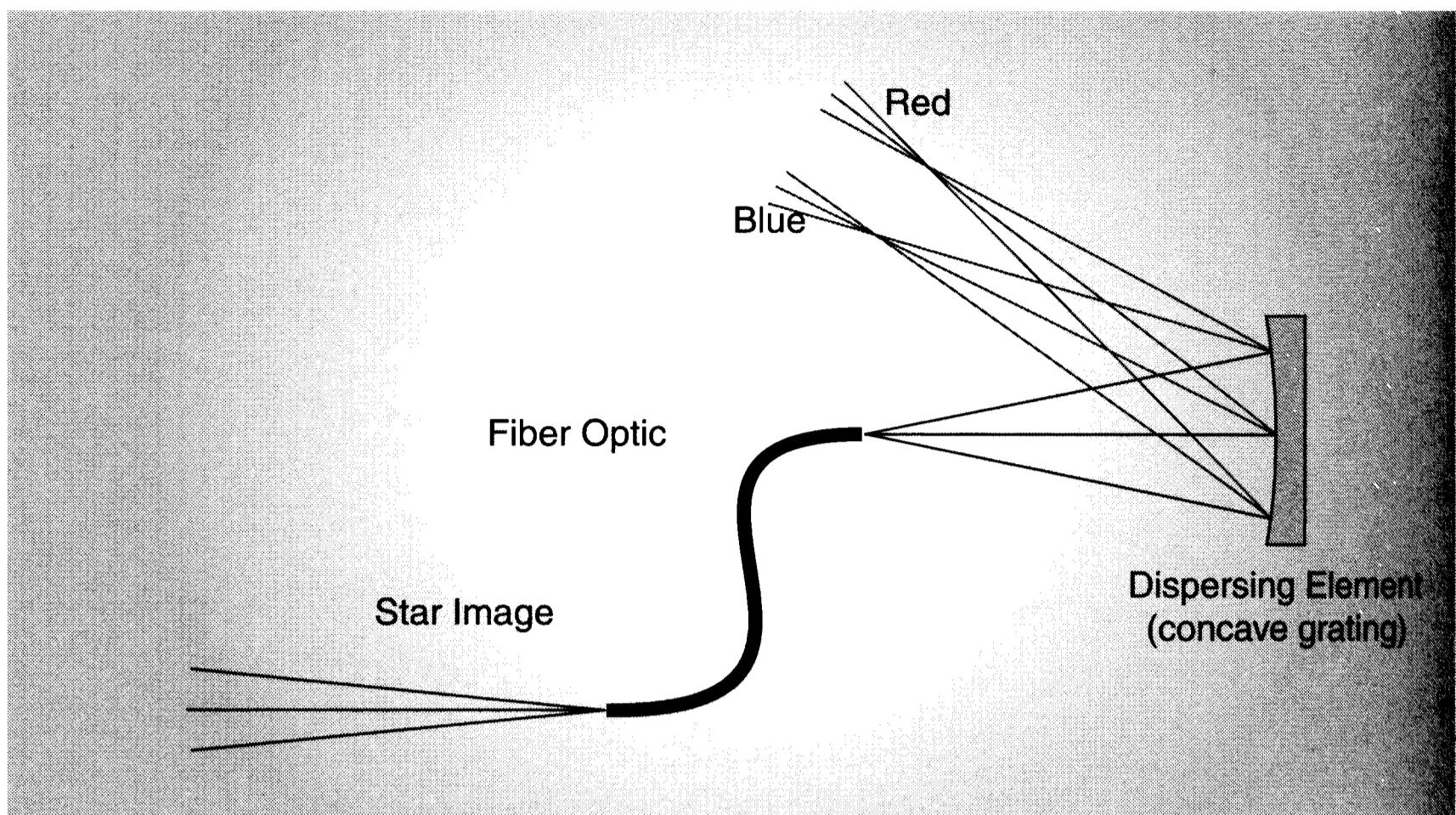


Figure 11.4 The fiber-fed spectrograph makes spectroscopy practical for amateur astronomers because starlight is carried from the focus to the spectrograph through a fiber-optic bundle, so the telescope does not have to support the weight of a spectrograph.

different wavelengths over a fairly wide range of angles with little or no vignetting. To avoid incurring significant aberration by dispersing a converging or diverging beam, rays passing through the dispersing element should be parallel. The basic elements are therefore a slit to isolate a thin strip of the light from the focal plane, a collimator to render the beam parallel, the dispersing element, and camera optics to reimaging the slit in monochromatic light.

To avoid vignetting, it may be necessary to place a field lens behind the slit to reimaging the telescope objective on the collimator, which must be able to collimate light over the full height of the slit. The dispersing element—whether a prism, grating, or grism—must be sized and placed to intercept all of the diverging parallel bundles from the collimator. The reimaging camera optics must be large enough to intercept ray bundles diverging along the height of the slit as well as along the axis of dispersion, and capable of forming good images over those two dimensions. The CCD is located at the focal plane of the camera optics to record the image that the spectrograph optics produce.

Finally, the assembled spectrograph optics must be mounted solidly at the focus of a telescope, with adequate provision for focusing, viewing the location of the slit, and guiding. Despite these challenges, amateur astronomers have designed and constructed slit spectrographs.

### 11.2.6 The Fiber-Fed Spectrograph

Because high-resolution slit spectrographs are bulky to mount at the focus of even a large telescope, in the late 1980s and early 1990s professional astronomers began experimenting with thin optical glass fibers to carry light from the focus of the

## Section 11.3: Properties of Spectrum Images

telescope to the slit of the spectrograph. Moving the heavy instrument off the telescope so improved the accuracy and stability of spectrographs that a wave of exoplanet discoveries (using radial velocity measurements to detect the effect of the orbiting planets on a parent star) soon followed. In addition, by placing hundreds of fibers at the focus of a telescope, observational cosmologists could record the spectra of a hundred distant galaxies per integration, making massive surveys of galaxy redshifts feasible.

For amateurs, fiber-fed designs move the spectrograph and CCD camera system to a convenient position beside the telescope. At the focus of the telescope is a “light pipe” to pick up starlight. The light pipe consists of a bundle of tiny optical fibers that convey the starlight several meters from the telescope to the spectrograph. At the pickup end, the fibers are bundled into a circular region to capture the whole star image; but at the output end, they are lined up side by side to form a narrow slit pattern.

The optical arrangement of this design parallels that of a slit spectrograph, but the properties of the fiber bundle dictate the design. Although starlight may enter a fiber bundle in an  $f/10$  cone, after transversing and exiting it, the light spreads into a conical distribution requiring a much faster lens to collimate. As an alternative to using a collimator, grating, and camera lens, the spectrograph can employ a concave reflection grating to collimate, disperse, and focus the spectrum with just one optical element. The CCD is placed to receive the spectrum at the focus of such a grating.

Although fiber-fed spectrographs are compact and convenient, the light pipe scrambles all of the light that enters it, mixing starlight and background sky light into an image that is effectively a single spectrum. Unless the source is so bright that the sky background can be ignored—fairly often the case with stars—it is necessary to point the telescope at a nearby patch of sky and make a separate exposure to capture the spectrum of the night sky. Since the spectrum is captured digitally, the background sky light can be subtracted later.

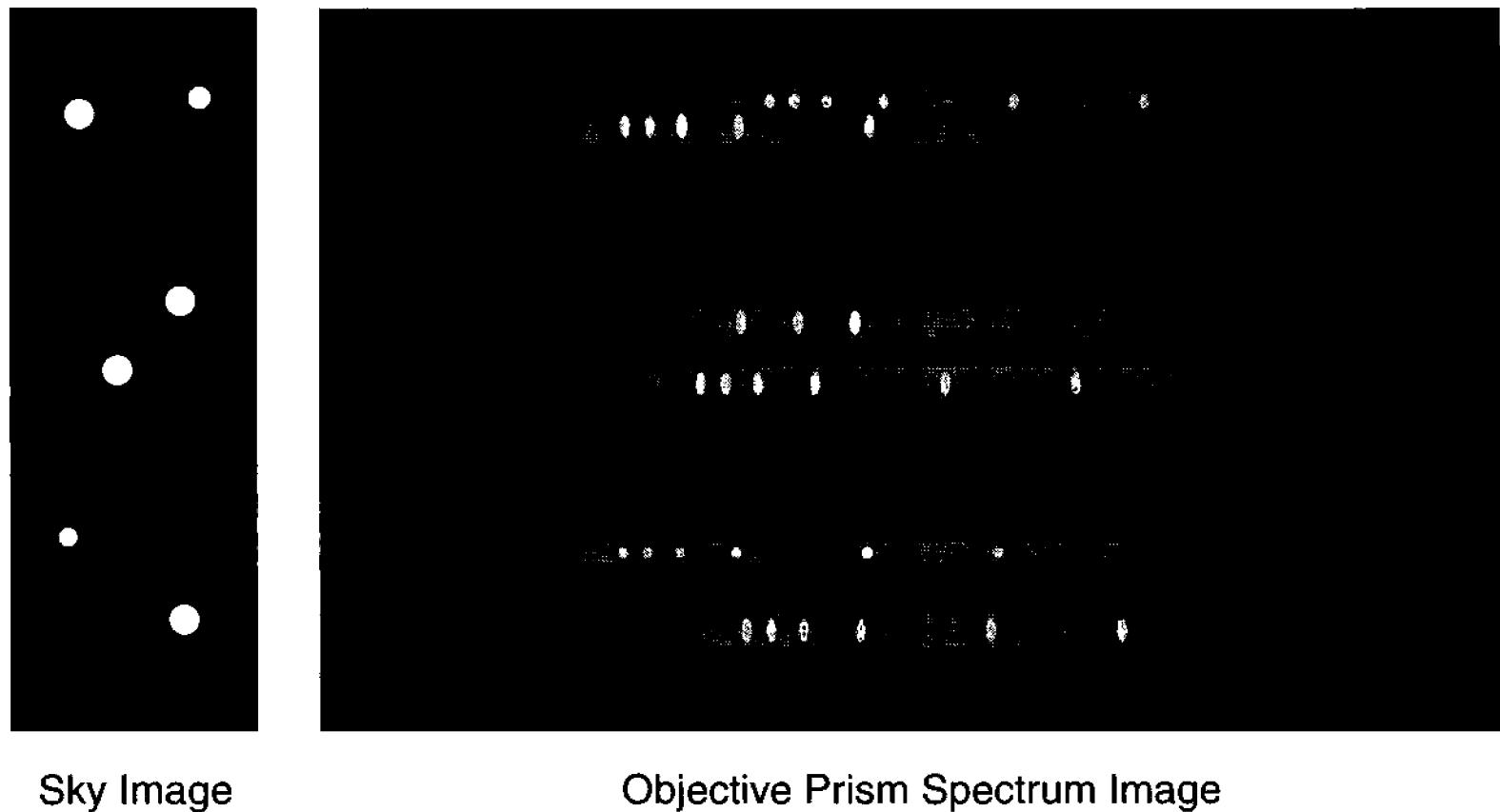
### 11.3 Properties of Spectrum Images

Spectra come in a great variety of lengths, widths, and resolutions. The splash of color thrown on the classroom wall from a prism is an extremely low-resolution spectrum, spreading the light only enough to show broad bands of color. To carry out spectroscopy, you must understand the range of wavelengths you need to examine, how the wavelengths should be spread, and how clearly separated the wavelengths need to be. In the jargon of spectroscopy, the key properties of spectra are:

- the free spectral range,
- the dispersion, and
- the spectral resolution.

These properties depend on the dispersing element, the optical design of the spec-

## Chapter 11: Spectroscopy



**Figure 11.5** The image seen with a telescope equipped with an objective prism consists of every object in the field of view converted into a spectrum. Because objective prism spectra are superimposed on the night sky background, obtaining spectra of faint objects may be difficult.

trograph, and the width of the slit, star image, or fiber tip.

The free spectral range is simply the range of wavelengths in the spectrum. In an objective prism spectrograph, the spectrum may run from the short-wavelength cutoff of the atmosphere at  $\sim 360$  nm in the near ultraviolet to the CCD's long-wavelength cutoff around 1100 nm in the near infrared, yielding a free spectral range of 740 nm. More typically, however, stellar spectra are studied with higher dispersion over a free spectral range of 150 nm.

Dispersion is the rate of change of wavelength with unit distance in the spectrum. In the days of photography, the dispersion was expressed in angstroms per millimeter; but in CCD spectroscopy, dispersion is usually given in nanometers per pixel. For a grating spectrograph, the dispersion is easy to calculate because it is nearly constant along the spectrum:

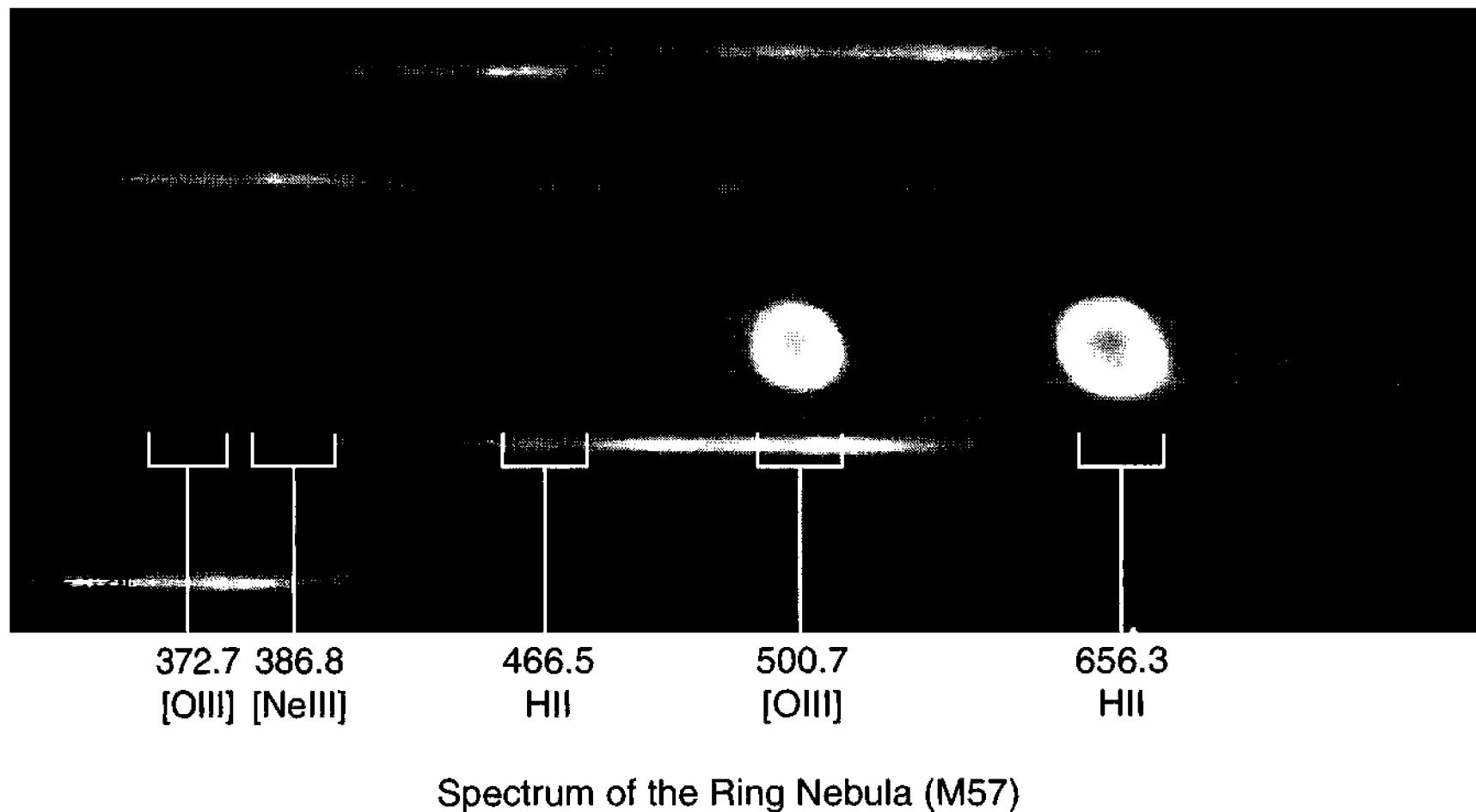
$$\text{Dispersion} = \frac{\text{Free spectral range}}{\text{Number of pixels}}.$$

For example, consider a spectrum image with a free spectral range of 150 nm on a CCD image 512 pixels wide; the dispersion would be 0.293 nm per pixel.

Spectral resolution is the instrument's ability to separate lines of nearly the same wavelength. The resolution achieved by a spectrograph is a rather complex mix of factors, including:

- the reimaged width of the slit,
- the size of the star image,
- the tip of the fiber tip,
- the dispersion of the prism or grating, and

## Section 11.4: Extracting a Spectrum from an Image



Spectrum of the Ring Nebula (M57)

**Figure 11.6** An objective prism spectrum of M57 reveals three strong spectral lines and more than a dozen weak ones. Planetary nebulae and HII regions emit light at discrete wavelengths, so their spectra show multiple images of the object. The strongest lines are due to hydrogen, oxygen, and neon.

- the optical quality of the spectrograph optics.

Assuming good optics and a reimaged slit smaller than a pixel width, the resolution in nanometers is roughly twice the dispersion. The resolving power of a spectrograph,  $R$ , is expressed as the ratio between the wavelength and the resolution at that wavelength:

$$R = \frac{\text{wavelength}}{\text{resolution}}$$

A spectrograph that can just resolve the two lines of the sodium doublet (at wavelengths of 589.0 and 589.6 nm) has a resolution of 0.6 nm and a resolving power of about 1000. Classification of star types can be done with resolving powers between 150 and 1000. The spectrographs that professional astronomers use to study the profiles of lines in stellar spectra have resolving powers in excess of 100,000.

## 11.4 Extracting a Spectrum from an Image

Raw spectrum images must be dark-subtracted and, with objective-prism and grating-prism spectrographs, flat-fielded before further processing. For the best signal-to-noise ratio, the master dark frame should be averaged or medianed from multiple darks. To further increase the signal-to-noise ratio, multiple spectrum images can be stacked, providing that the spectrum falls in exactly the same location in every image.

With slit- and fiber-fed spectrographs, you should obtain wavelength and flux calibration spectra. Wavelength calibrations are often done using neon and ar-

## Chapter 11: Spectroscopy

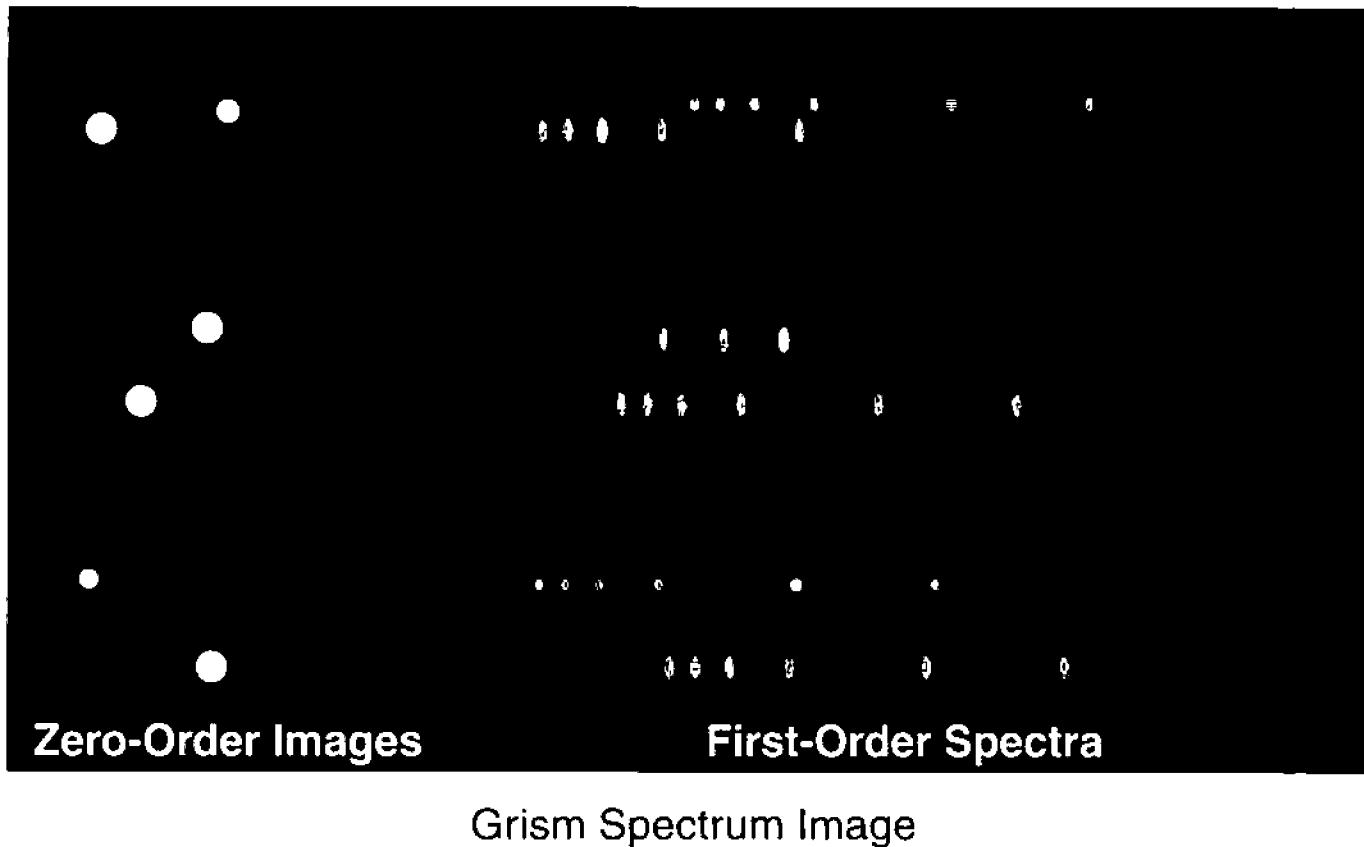


Figure 11.7 Spectra formed by gratings and grisms have a zero-order image of every celestial object in the field as well as a spectrum of each object. However, the distance between the zero-order image and features in the spectrum is constant, making it easy to calibrate the wavelength scale.

gon glow lamps. Flux calibrations are done by taking spectra of a source, such as a tungsten filament, with a known spectrum.

Recall that the spectrum is a graph, plot, or image of the flux versus wavelength. In the raw image from a spectrograph, the spectrum is widened by several pixels in height and contaminated by the background sky light. To extract the spectrum from an image, the background light must be removed and the spectrum summed over its height. The extracted spectrum consists of a plot of the extracted signal versus a pixel coordinate.

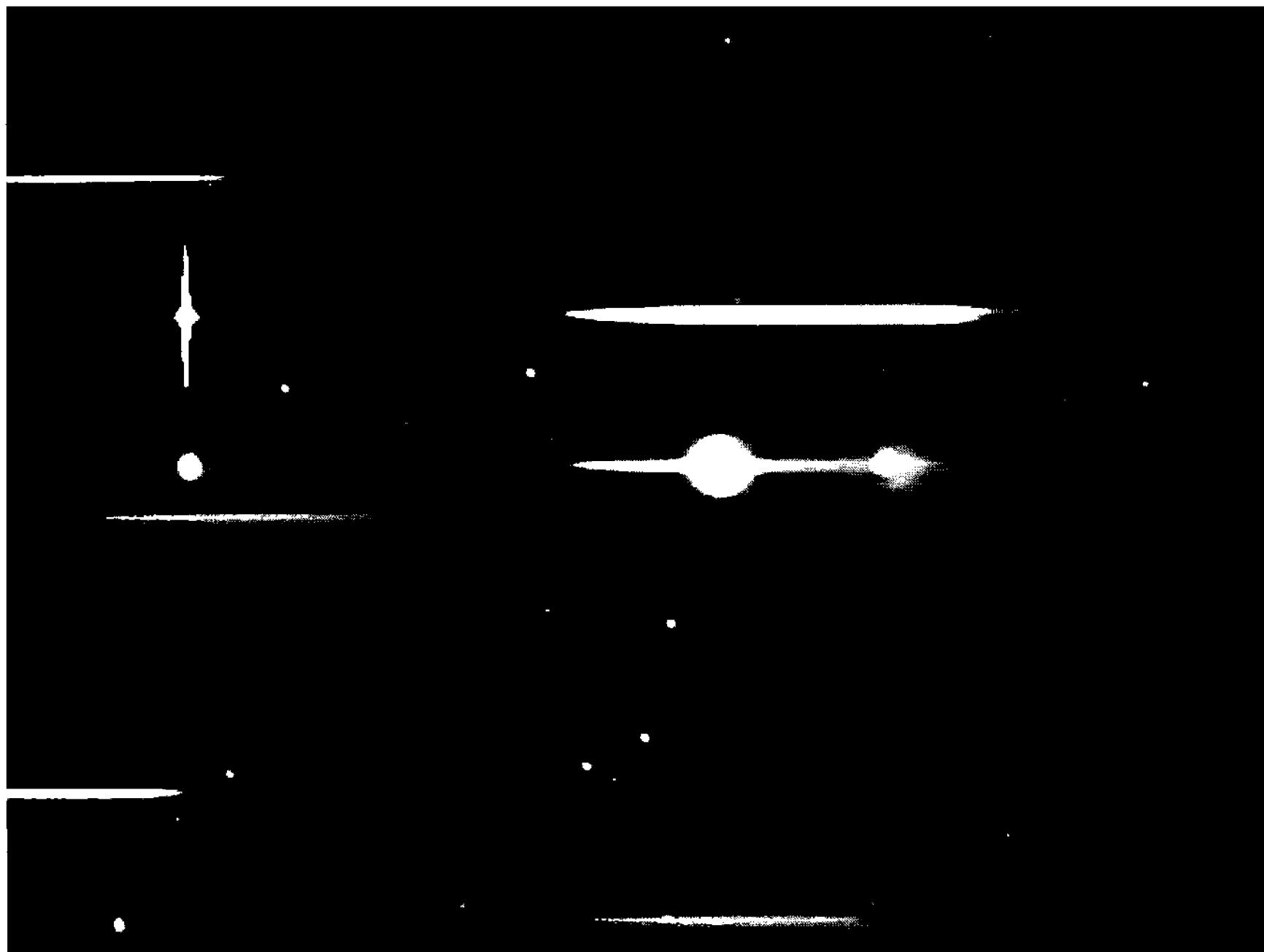
- **Tip:** *AIP4Win provides a Spectroscopy tool for extracting spectrum data from CCD images. To create wavelength—and intensity—calibrated spectra, transfer the extracted data to spreadsheet software to perform the analysis that your observations require.*

### 11.4.1 Spectra from Objective Prism Images

Spectrum images from telescopes equipped with an objective prism consist of spectra superimposed on the light of the night sky. Because the locations of the spectra depend on the locations of the stars in the sky and the orientation of the objective prism, partial spectra enter and exit the images; and complete spectra in the center of the image may overlap one another.

Assuming that the spectrum you wish to extract does not overlap other spectra, and is oriented with the direction of dispersion along the sample axis (i.e., the  $x$ -axis) of the CCD chip, extracting the spectrum is straightforward. (If the spectrum is not properly aligned, the image should be rotated and enlarged before extracting it.) Extracting the data consists of identifying the range of lines contained

## Section 11.4: Extracting a Spectrum from an Image



**Figure 11.8** This spectrum of the Eskimo Nebula, NGC2392, was made with a diffraction grating placed ahead of focus. Zero-order spectra appear to the left of the spectrum for each object. The zero-order images simplify calibrating the wavelength scale of the spectra. Image by Tim Puckett.

in the spectrum, and two ranges of lines containing regions of blank sky above and below it. The formula for the spectrum is:

$$S(x) = \sum_{y_{\text{star+sky}}} P_{x,y} - \alpha \left( \sum_{y_{\text{sky1}}} P_{x,y} + \sum_{y_{\text{sky2}}} P_{x,y} \right) \quad (\text{Equ. 11.3})$$

where  $S(x)$  is the spectrum at pixel  $x$ ,  $P_{x,y}$  is the pixel value of pixel  $(x, y)$ , and  $\alpha$  is a coefficient that matches the measured brightness of the background sky to the sky brightness contaminating the spectrum. Assuming a uniform sky background, you would compute  $\alpha$  from:

$$\alpha = \frac{\text{Lines of sky background}}{\text{Lines of star spectrum}}. \quad (\text{Equ. 11.4})$$

For example, if the range of lines containing the spectrum included six lines, and you included four lines of sky above the spectrum and five lines of sky below it—a total of nine sky lines—then  $\alpha$  would be 0.6667.

## Chapter 11: Spectroscopy

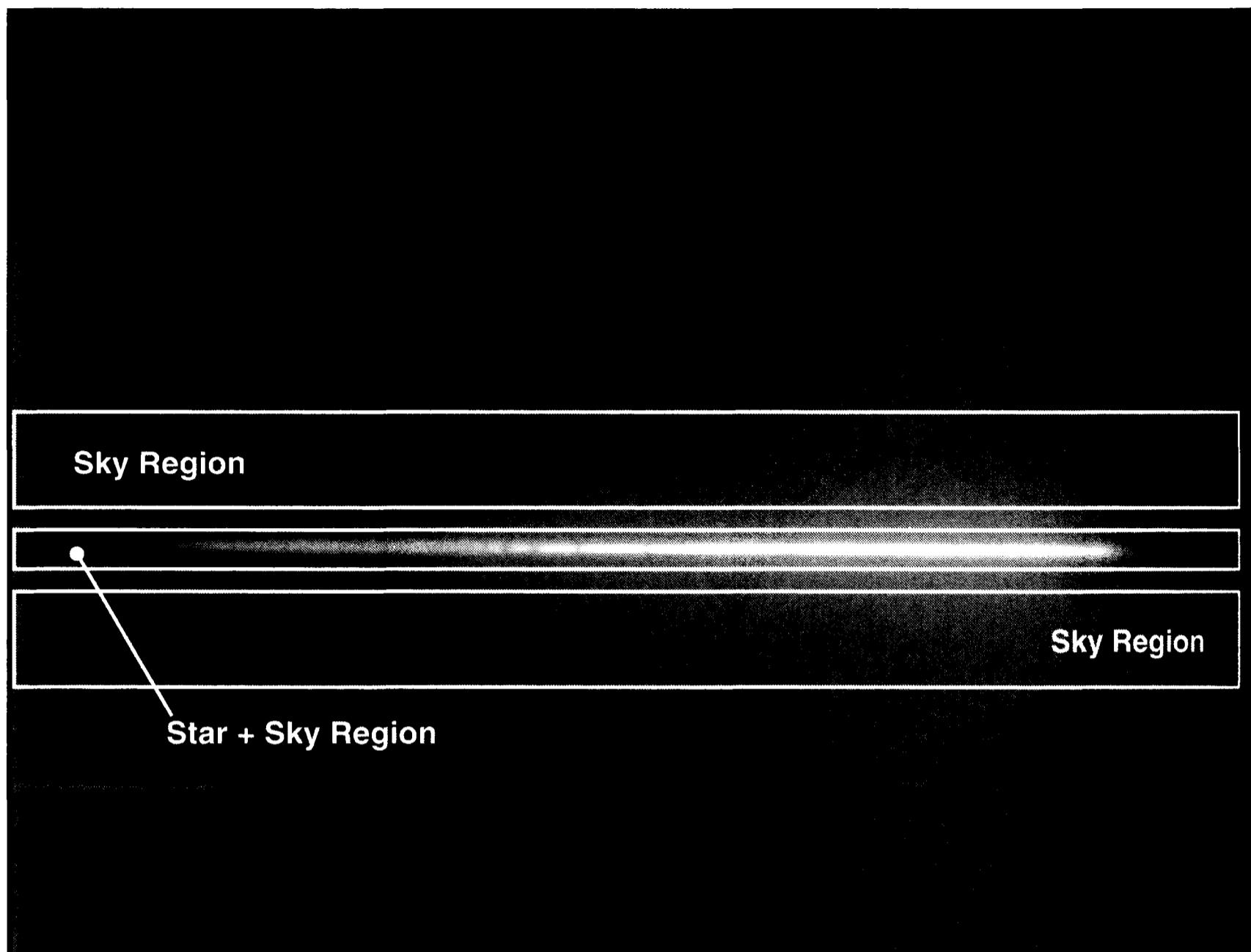


Figure 11.9 In this objective prism spectrum, the region containing the star image—superimposed on the light of the background sky—lies between two sky background regions. Measuring the sky background and removing it from the spectrum curve yielded the spectrum in Figure 11.10.

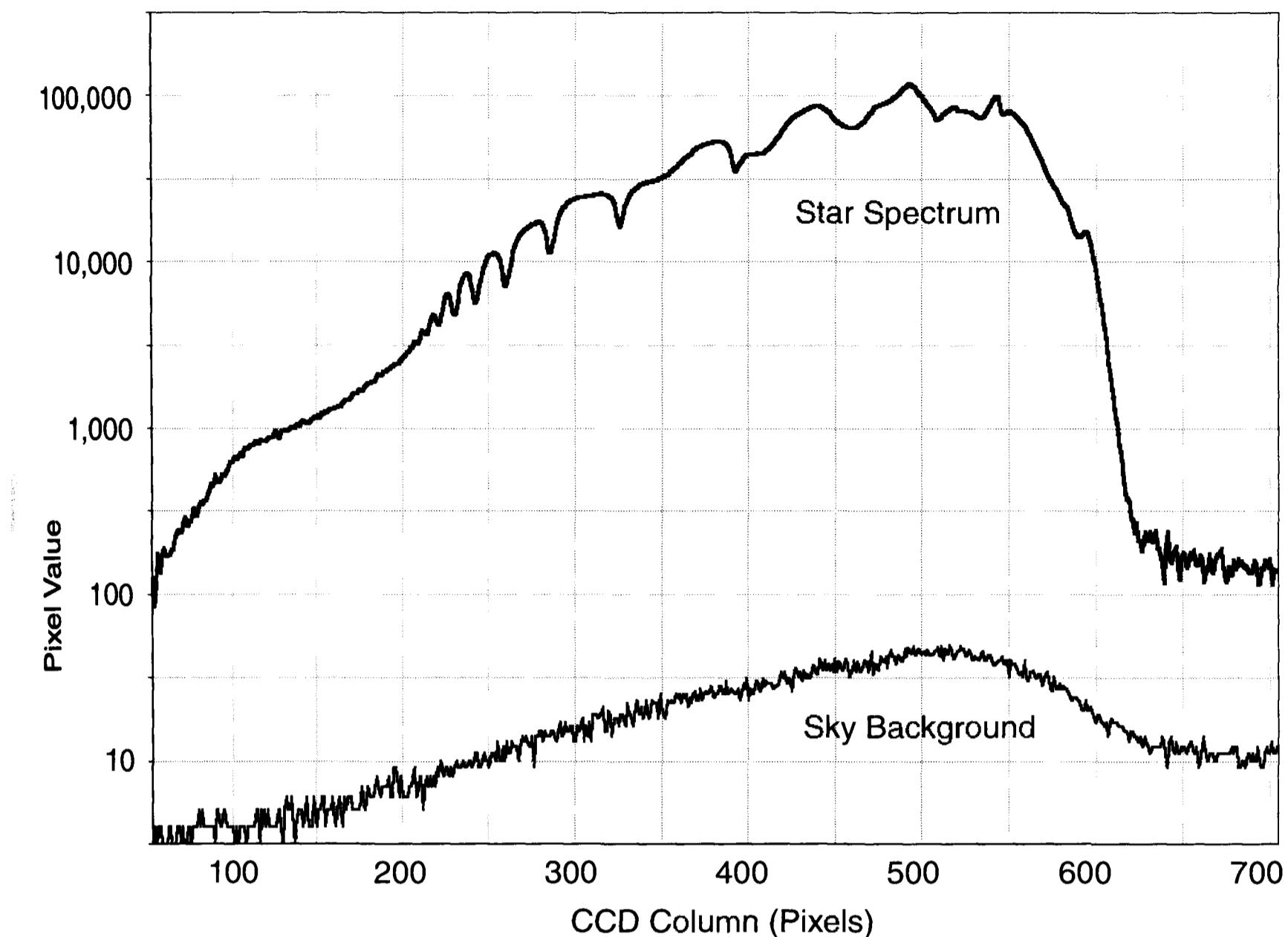
- **Tip:** *AIP4Win does not assume that the sky background is uniform. Instead, it saves a table of  $S_{\text{star+sky}}(x)$  and  $S_{\text{sky}}(x)$ , leaving the determination of the optimum sky subtraction to the observer.*

### 11.4.2 Spectra from Slit Spectrographs

Images from slit spectrographs consist of a slice of sky spread into a spectrum. If the CCD has been aligned correctly, its long axis coincides with the long axis of the spectrum. In the days of photographic spectroscopy, the normal practice was to broaden the stellar spectra by trailing the star back and forth along the slit—a practice made necessary by the limited storage capacity of the photographic emulsion and made desirable by the greater ease of “reading” widened spectra. With CCDs, widening the spectrum is necessary only in the case of extremely bright stars to prevent saturation.

The spectrum of the object of interest thus appears as a streak against a broad spectrum of the night sky. Its width can be as little as three or four pixels for a well-guided spectrum of a dim star, or as much as half the height of the slit for an extended source such as a nebula or galaxy. For extended objects that fill the slit height, it is necessary to make additional sky spectrum exposures well away from the object under study. In spectrographs designed for measuring precise wave-

## Section 11.4: Extracting a Spectrum from an Image



**Figure 11.10** This graph, plotted with Excel spreadsheet software, shows the spectrum of an A0 main-sequence star. The data, plotted on a logarithmic scale, covers the CCD's entire range of sensitivity, from 350 nm at the left to 950 nm at the right. The hydrogen Balmer series starts around column 210.

lengths, comparison spectra produced by glow lamps filled with argon and neon may overlay or flank the object and sky spectra.

The process of extracting a spectrum from a slit spectrograph image is essentially identical to extracting an objective spectrum even though what you are actually doing is quite different. With the objective prism spectrum, you are removing a sky background of all colors; whereas in the slit spectrum, you are subtracting the spectrum of the night sky from the combined spectrum of the object and the night sky.

To perform the extraction, identify those lines that contain the star spectrum, and identify the lines above and below the spectrum of the night sky. The spectrum of the object alone results from subtracting the appropriately weighted sky spectrum from the star spectrum:

$$S(x) = \sum_{y_{\text{star+sky}}} P_{x,y} - \alpha \left( \sum_{y_{\text{sky1}}} P_{x,y} + \sum_{y_{\text{sky2}}} P_{x,y} \right). \quad (\text{Equ. 11.5})$$

Summations are carried out in the  $y$  axis over the lines containing the star and the background sky.  $S(x)$  is the spectrum at pixel  $x$ ,  $P_{x,y}$  is the pixel value of pixel

## Chapter 11: Spectroscopy

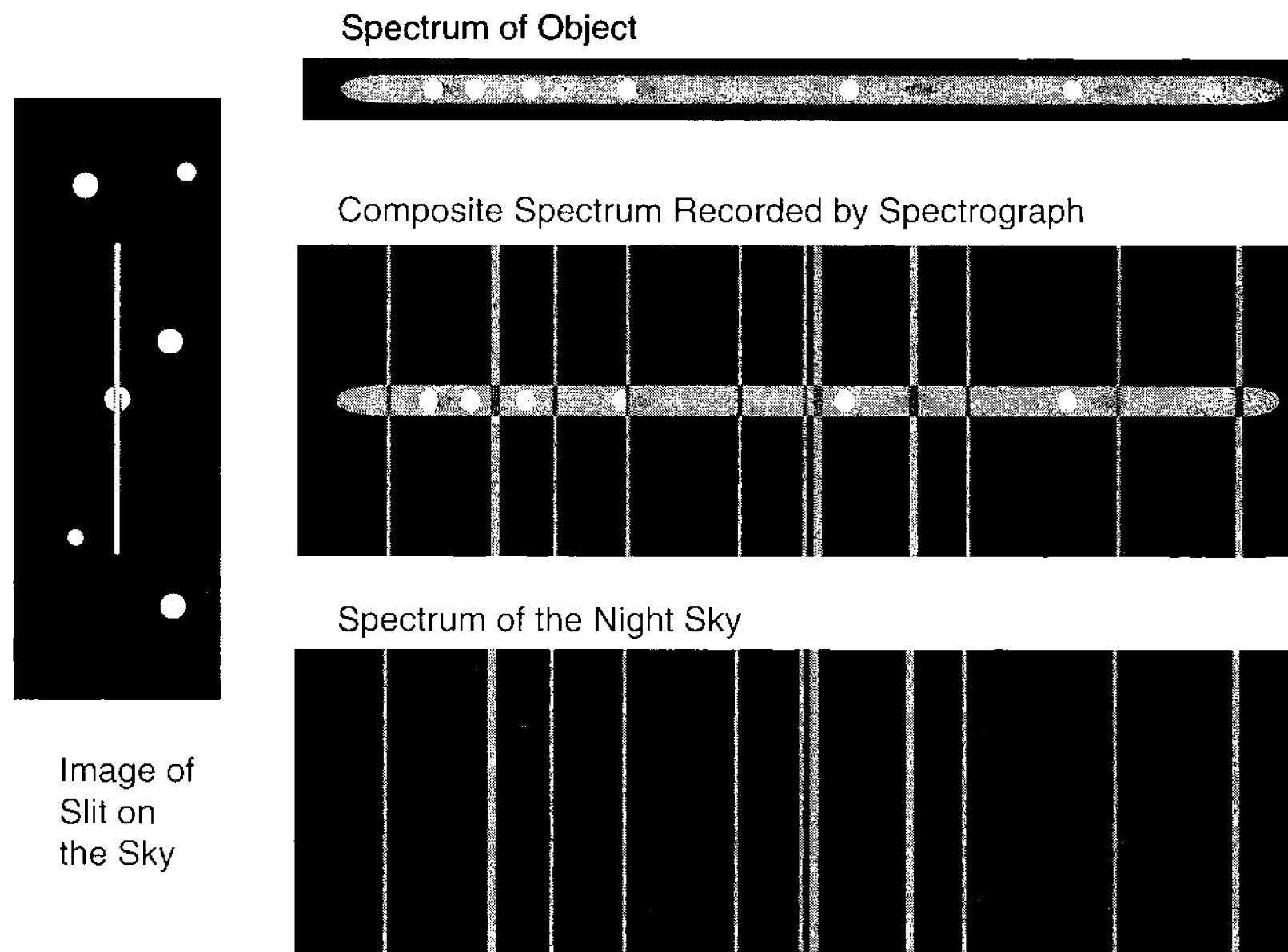


Figure 11.11 The image formed by a slit spectrograph is the spectrum of the narrow slice of sky admitted by the spectrograph slit. The spectrum is a combination of the spectrum of the night sky plus that of the celestial object. In CCD spectra, an uncontaminated spectrum of the object can be recovered.

( $x, y$ ), and  $\alpha$  is a coefficient chosen to remove the contaminating sky spectrum accurately. In the interest of a good signal-to-noise ratio, it is desirable to have more lines of sky than of star spectrum.

- **Tip:** *AIP4Win saves the extracted spectrum data as a text file containing the x-coordinate pixel, the sum of the star lines, and the sum of the sky lines. Because the sky illumination may not be uniform, the observer should carefully adjust the scaling coefficient to obtain the best correction of sky contamination.*

The effective wavelength of slit spectra can be calibrated using either the comparison neon/argon spectra (if they are present) or by making separate spectra of a neon/argon calibration lamp, and using the data to derive the relationship between known wavelengths in the calibration spectrum and their  $x$ -axis position in the spectrum image. In a pinch, spectral features in the sky background can be used to anchor a wavelength scale previously established more rigorously.

### 11.4.3 Spectra from Fiber-Fed Spectrographs

The spectrum image in simple fiber-fed spectrographs consists of the light of the star and the background sky scrambled together. To remove the sky background,

## Section 11.5: Spectrum Calibration and Analysis

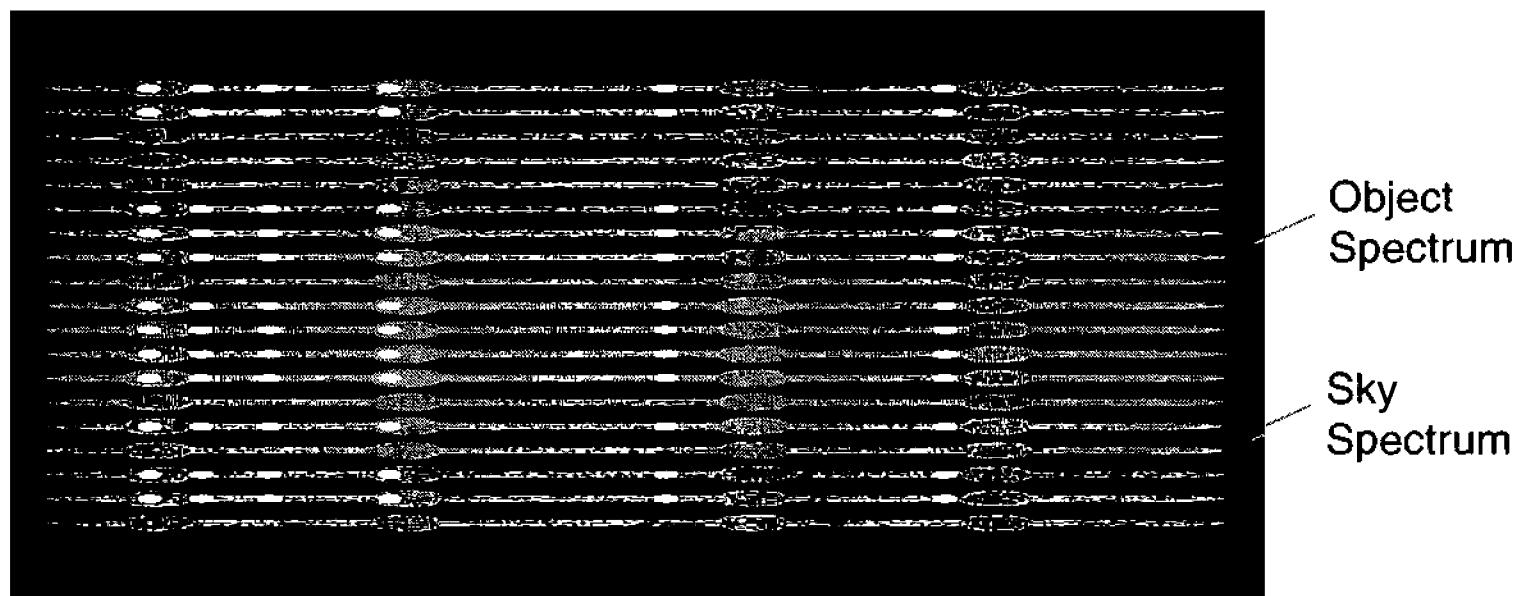


Figure 11.12 The spectrum from a fiber-fed spectrograph consists of narrow spectra from each fiber in the bundle of fibers that conveys light from the telescope to the spectrograph. By taking a separate spectrum, the sky background spectrum can be removed from that of the object.

it is necessary to make a separate exposure of the sky, extract that spectrum, and subtract it from the extracted spectrum of the star plus the sky. If the star spectrum is very much brighter than that of the sky, you may decide to ignore the contribution of the night sky. To derive a wavelength scale, obtain spectra of a neon/argon calibration lamp or other calibration spectrum, and extract the calibration spectrum.

To extract a spectrum, for each  $x$ -axis position sum the pixel values over the  $y$ -axis height of the spectrum:

$$S(x) = \sum_{y_{\text{star}}} P_{x,y}. \quad (\text{Equ. 11.6})$$

Here,  $S(x)$  is the spectrum at pixel  $x$ , and  $P_{x,y}$  is the pixel value of pixel  $(x, y)$ . Repeat the extraction for the sky spectrum, then for each  $S_{\text{star+sky}}(x)$ , subtract the corresponding background  $S_{\text{sky}}(x)$ . To improve the signal-to-noise ratio, you can sum multiple spectra of the star; and to improve the sky background correction for faint objects, you can take and sum multiple sky background spectra.

## 11.5 Spectrum Calibration and Analysis

When spectral data are extracted from an image, the result is a list of pixel values at a series of pixel locations. The list may contain pixel values from the spectrum only, values from the spectrum and background, and for serious observations, values from wavelength and intensity calibration lamps.

Removing the background, or sky spectrum correction, is the usual first step. This is accomplished by scaling the background or sky values and then subtracting them from the spectrum:

## Chapter 11: Spectroscopy

$$S(x) = S_{\text{star+sky}}(x) - \alpha S_{\text{sky}}(x) \quad (\text{Equ. 11.7})$$

where  $S_{\text{star+sky}}(x)$  is the spectrum pixel value at pixel  $x$ ,  $S_{\text{sky}}(x)$  is the sky pixel value at pixel  $x$ , and  $\alpha$  is a scaling factor chosen to remove the background or sky spectrum from the object spectrum. There are some useful constraints that aid in the choice of a value for  $\alpha$ : the pixel value of the corrected spectrum cannot be negative, and atmospheric features such as the 557.7 nm airglow line in the sky spectrum should be absent from stellar spectra.

After correction for background, slit and fiber spectra can be calibrated in wavelength if you have taken spectra of a calibration source. Wavelength calibration is accomplished by measuring  $x$  pixel position for spectra lines of known wavelength. This should produce a dataset consisting of  $x$ -axis positions,  $x_i$ , and corresponding wavelengths,  $\lambda_i$ . From these, use the method of least squares to find the best fitting polynomial:

$$\lambda_i = a + bx_i + cx_i^2 + dx_i^3 + \dots \quad (\text{Equ. 11.8})$$

where the polynomial coefficients will allow you to express the pixel values as calibrated wavelengths.

If you have performed a radiometric calibration of the spectrograph by taking spectra of an object with a known energy distribution, you can remove the instrumental transmission, grating efficiency curve, and sensitivity dependences on wavelength, and convert the spectral data into energy units.

The analysis and use of spectra extend far beyond image processing and the extraction of spectra from images. Spectra supply much of the basic information that astronomers use in examining celestial bodies:

- star classification by temperature and composition,
- identification of stellar populations of clusters and galaxies,
- finding temperature and composition of gaseous nebulae,
- radial velocities (i.e., redshifts) of galaxies and QSOs,
- chemical and physical properties of newly discovered objects.

Although spectroscopy is more involved than traditional areas of amateur astronomy, the availability of compact, efficient spectrographs equipped with CCD detectors to record faint spectra, and computers to assist in processing and displaying the large amount of data generated, have opened this area to amateurs as never before.

# 12 Geometric Transforms

This chapter explores a class of image operations in which the content of the image is not “changed” as such, but its context—the array that contains it—is changed. In it, you will learn what the basic geometric image transforms are and see how they are implemented as computer algorithms. Geometric transforms include:

- translation, or shifting the image;
- rotation, or turning the image;
- scaling, or resizing the image;
- flipping and flopping, or mirroring the image;
- cropping and floating, or cutting off or adding to the image;
- resampling, or enlarging and shrinking the image.

The results of geometric transforms appear obvious—the image is shifted, turned, enlarged, or shrunk. Not so obvious is the manipulation of the discrete pixel structure that can result in the loss of image information. The better you understand why and how information is lost, the better you can manipulate and assemble images into track-and-stack images, movies, and great color pictures.

## 12.1 Translation

Translation means shifting an image up, down, or from side to side. The image itself is not changed, but simply moved to a new location. Each pixel  $P(x, y)$  becomes a new pixel  $P(x', y')$ . The equations for translating a point are:

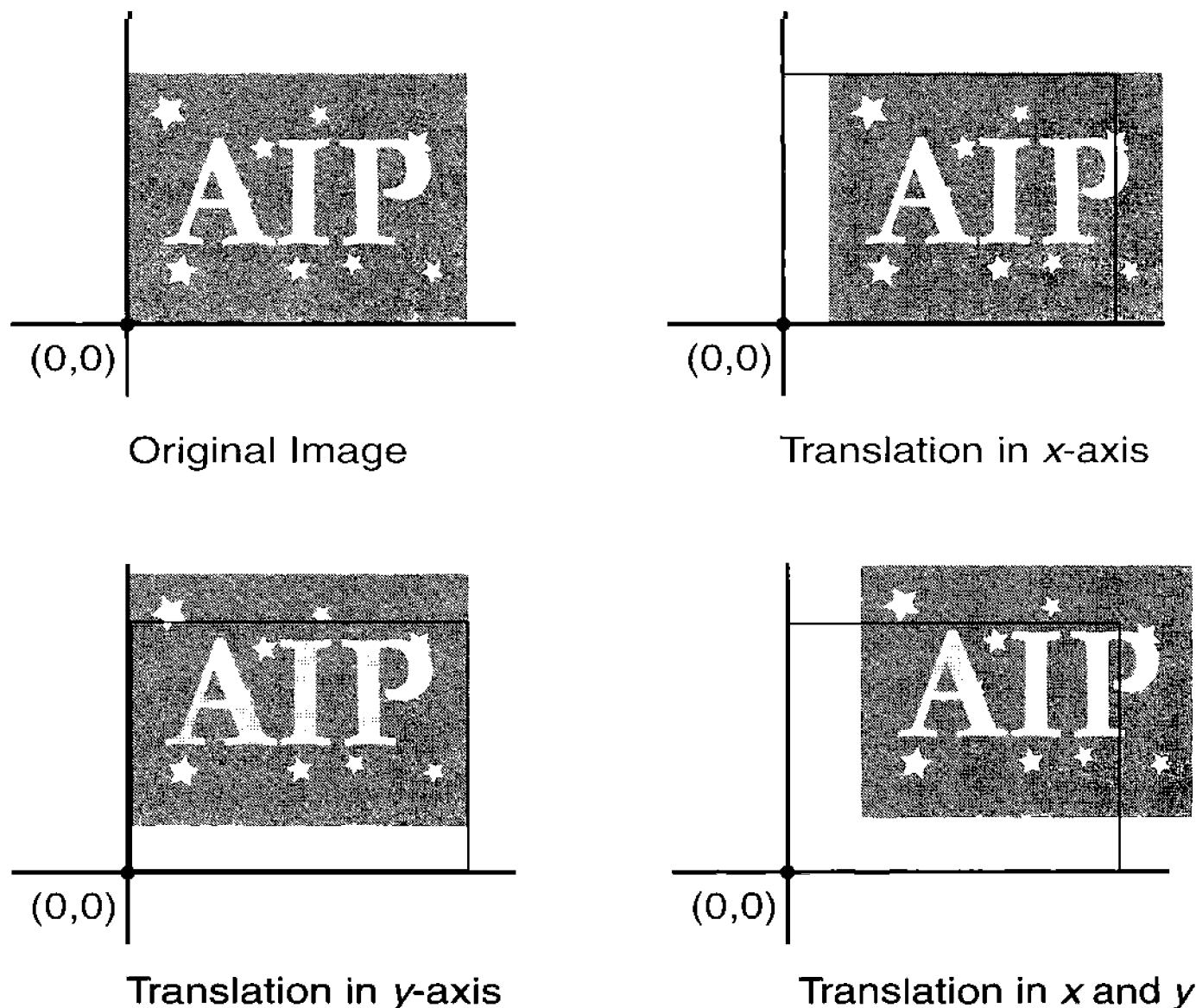
$$\begin{aligned}x' &= x + x_T \\y' &= y + y_T.\end{aligned}\tag{Equ. 12.1}$$

where  $x'$  and  $y'$  are the coordinates of  $P$  in the new image,  $x$  and  $y$  are the coordinates in the old image, and  $x_T$  and  $y_T$  are the shifts in the two axes.

There are many reasons for wanting to employ translation, but the most common application is to register one image precisely with another. Once they are registered, the images can be blinked or combined.

Although the mathematics of translation are straightforward, in this section

## Chapter 12: Geometric Transforms



**Figure 12.1** Translation appears to be the simplest possible image transform—the image is simply moved left, right, up, or down. However, to be really useful, translation must work when you need to shift the image by a non-integer number of pixels, and that means finding pixel values between pixels.

we look at programming translation to illustrate the difficulties that beset programming geometric transforms.

Assuming that you have an original image array, `old(0 TO xmax)`, and a second array, `new(0 TO ymax)`, to receive the translated image, you might initially try to implement translation like this:

```

FOR y = 0 to ymax
    FOR x = 0 TO xmax
        xp = x + xt
        yp = y + yt
        new(xp,yp) = old(x,y)
    NEXT x
NEXT y

```

where `x` and `y` are the coordinates in the original image, `xp` and `yp` are the coordinates of the new location, `xt` and `yt` are integer translations in the `x` and `y` axes, and the expressions `(0 TO xmax)` and `(0 TO ymax)` define the array boundaries in the `x` and `y` axes, respectively.

Unfortunately, this will not work because `xp` and `yp` will be assigned values outside the bounds of the `new()` array. Furthermore, the routine never writes to parts of the `new()` array, leaving some parts of the new image either zero or not defined.

To make sure that every point in `new()` is evaluated, you must write the loops to generate a new image rather than scan the old image. This necessitates recasting the translation equations to yield  $x$  instead of  $x'$ :

$$\begin{aligned}x &= x' - xt \\y &= y' - yt.\end{aligned}\tag{Equ. 12.2}$$

Finally, to insure that the program does not attempt to access the `old()` array outside its bounds, you must check that both  $x$  and  $y$  lie within the bounds of `old()`, and if they do not, you must assign a value to `new()`, in this case zero. Recall that the array dimensions run from 0 to `xmax` and 0 to `ymax`.

```

FOR yp = 0 to ymax
    FOR xp = 0 TO xmax
        x = xp - xt
        y = yp - yt
        IF x<0 OR x>xmax OR y<0 OR y>ymax THEN
            new(xp, yp) = 0
        ELSE
            new(xp, yp) = old(x, y)
        END IF
    NEXT xp
NEXT yp

```

Thus far we have assumed that you can get away with translating the image by an integer number of pixels—but what if you need to shift the image 5.539 pixels to the left and 9.745 pixels down for a perfect match? Since there is no pixel waiting at that location to be copied from the `old()` array, you need to calculate what the pixel value would be—if there were a pixel at the right location.

To assign an appropriate pixel value for the location, we can interpolate; that is, we can compute the weighted average of the four pixels surrounding a point between pixels in the old image and assign this computed value to the new pixel. Here's how it works: in executing the translation algorithm we have just found that we need a pixel value for the location  $x, y$  in the old image, but  $x$  and  $y$  are not integers. We begin by designating  $x$  and  $y$  as floating point numbers. Next, we copy the values of the four surrounding pixels in the `old()` array into four new variables, `a`, `b`, `c`, and `d`:

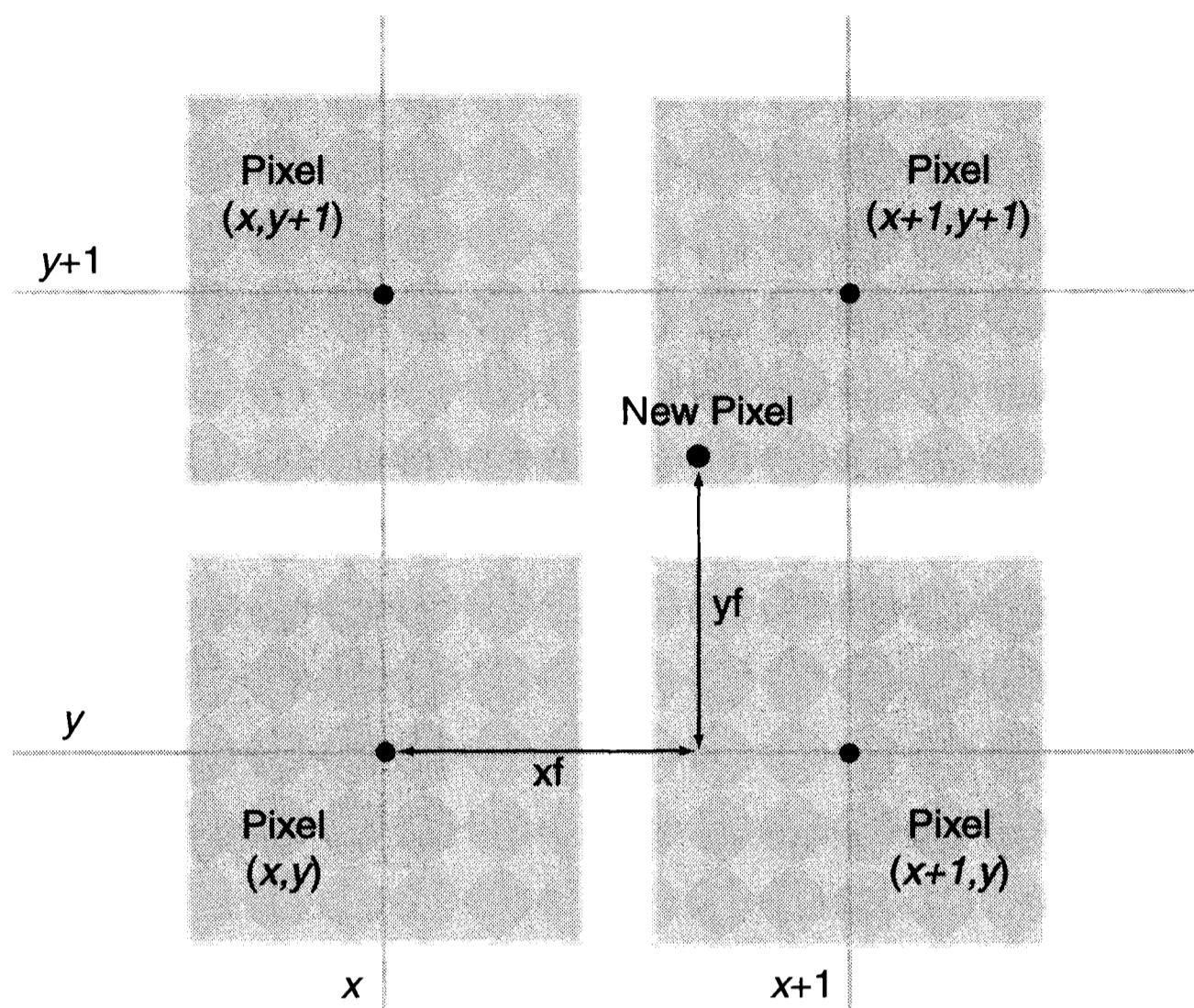
```

a = old(INT(x), INT(y))
b = old(INT(x), INT(y)+1)
c = old(INT(x)+1, INT(y))
d = old(INT(x)+1, INT(y)+1)

```

The `INT` function returns an integer, so that given an  $(x, y)$  position of (23.4, 56.7), the first line above would assign the pixel value of `old(23, 56)` to the variable `a`. The value of the new interpolated pixel must fall somewhere between the values of these pixels. It cannot be smaller than the smallest of `a`, `b`, `c`, or `d`, nor larger

## Chapter 12: Geometric Transforms



**Figure 12.2** When a pixel in the new image you are creating,  $(xp, yp)$  falls on a non-integer location in the old image,  $(x+xf, y+yf)$ , you must compute the value that the pixel in that location would have had, if there had been a pixel at that location. The process is called interpolation.

than the largest of  $a$ ,  $b$ ,  $c$ , or  $d$ .

To determine what fraction of the way the new pixel lies between  $\text{INT}(x)$  and  $\text{INT}(x) + 1$ , assign the result to two more new variables,  $xf$  and  $yf$ , to hold the fractional part of the surrounding `old()` pixels:

$$\begin{aligned} xf &= x - \text{INT}(x) \\ yf &= y - \text{INT}(y) \end{aligned}$$

The variables  $xf$  and  $yf$  necessarily have values between 0 and 1. To find the value of the new pixel, we add the fractional contribution from each of the four surrounding pixels:

$$\begin{aligned} \text{new}(xp, yp) = & a * (1-xf) * (1-yf) _ \\ & + b * (1-xf) * yf _ \\ & + c * xf * (1-yf) _ \\ & + d * xf * yf \end{aligned}$$

Note that the trailing underscore character “`_`” means that we have broken the line to fit on the page; the line actually continues without a break.

You can now assemble these elements into a translation procedure that takes non-integer arguments for translation and returns a bilinearly interpolated image:

```

PROCEDURE TRANSLATE (xp, yp)
FOR yp = 0 to ymax
    FOR xp = 0 TO xmax
        x = xp - xt
        y = yp - yt
        IF INT(x)<0 OR INT(x)+1>xmax OR_
            INT(y)<0 OR INT(y)+1>ymax THEN
            new(xp,yp) = 0
        ELSE
            xf = x - INT(x)
            yf = y - INT(y)
            a = old(INT(x), INT(y))
            b = old(INT(x), INT(y)+1)
            c = old(INT(x)+1, INT(y))
            d = old(INT(x)+1, INT(y)+1)
            new(xp,yp) = a * (1-xf) * (1-yf) -
                + b * (1-xf) * yf -
                + c * xf * (1-yf) -
                + d * xf * yf
        END IF
    NEXT xp
NEXT yp
END PROCEDURE TRANSLATE

```

With few exceptions, geometric transforms must be programmed to work “backward.” Rather than converting original image coordinates into new image coordinates, you must set up the equations to find old-image coordinates for each new image pixel. The computed coordinates must be checked to insure that they lie inside the boundaries of the old image; and finally, the pixel value of the new image must be interpolated from the four pixels surrounding the non-integer old-image coordinates.

## 12.2 Rotation

It is often necessary to rotate an image to match the orientation of another one, whether for blinking, track-and-stack compositing, or registering filtered images to make a color image. The basic equations for rotating a point  $(x, y)$  around the origin of the coordinate system are:

$$\begin{aligned} x' &= x \cos \vartheta + y \sin \vartheta \\ y' &= -x \sin \vartheta + y \cos \vartheta \end{aligned} \tag{Equ. 12.3}$$

where  $\vartheta$  is the angle of rotation measured counterclockwise. In image processing, this is often not very useful because the coordinate origin is in the lower left-hand corner of the image. More often, we will need to rotate the image about its center

## Chapter 12: Geometric Transforms

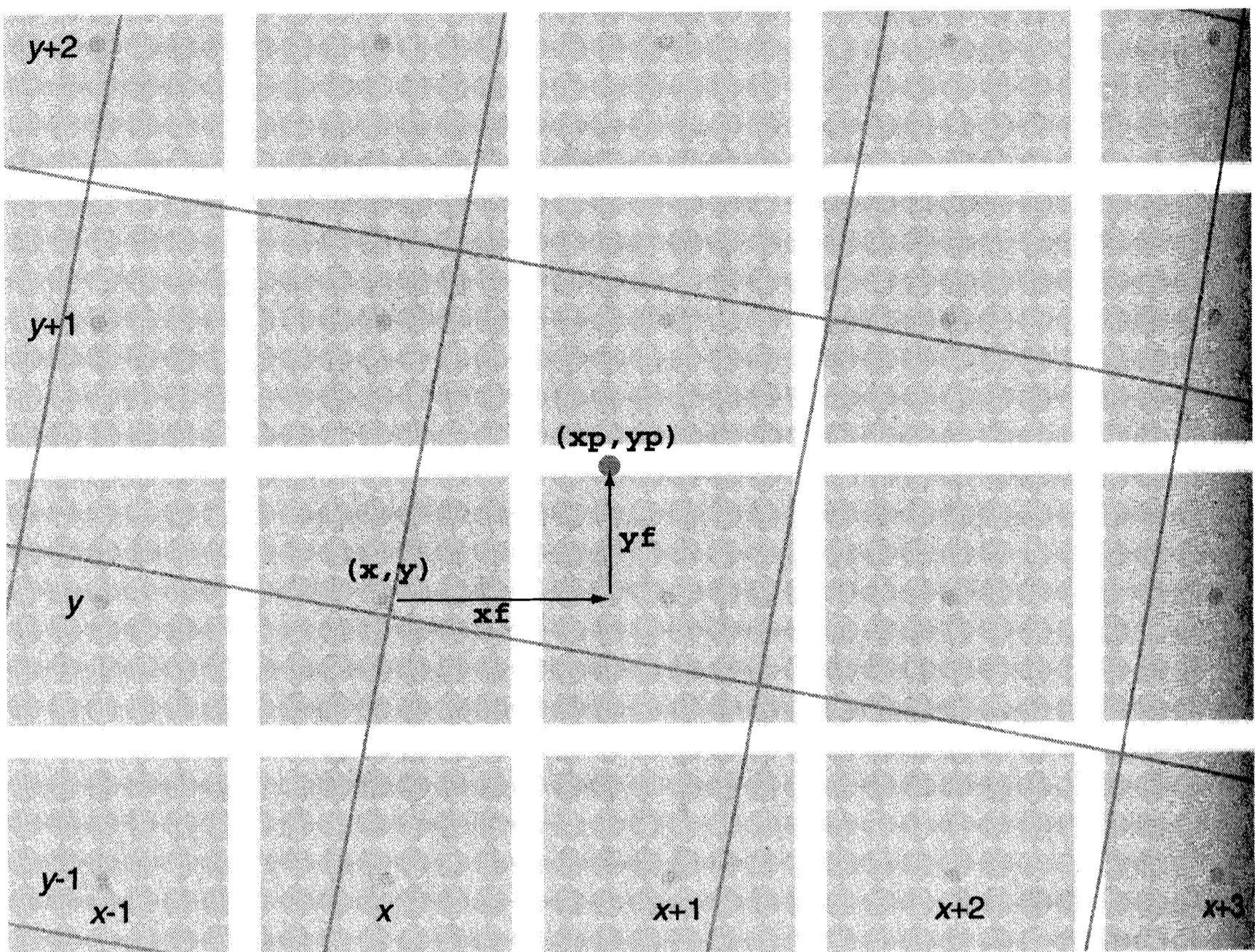


Figure 12.3 Interpolation is necessary whenever a new grid of pixels does not exactly match an original pixel grid. This occurs when images are translated, rotated, scaled, or resampled. New values are computed by interpolating the values of the four pixels around the new pixel's location.

or around a point such as star image. To rotate a point  $(x, y)$  about point  $(x_0, y_0)$ , the equations become:

$$\begin{aligned} x' &= x_0 + (x - x_0) \cos \vartheta + (y - y_0) \sin \vartheta \\ y' &= y_0 - (x - x_0) \sin \vartheta + (y - y_0) \cos \vartheta. \end{aligned} \quad (\text{Eqn. 12.4})$$

Note that the point  $(x_0, y_0)$  does not move; it occupies the same location in the new image as it did in the original.

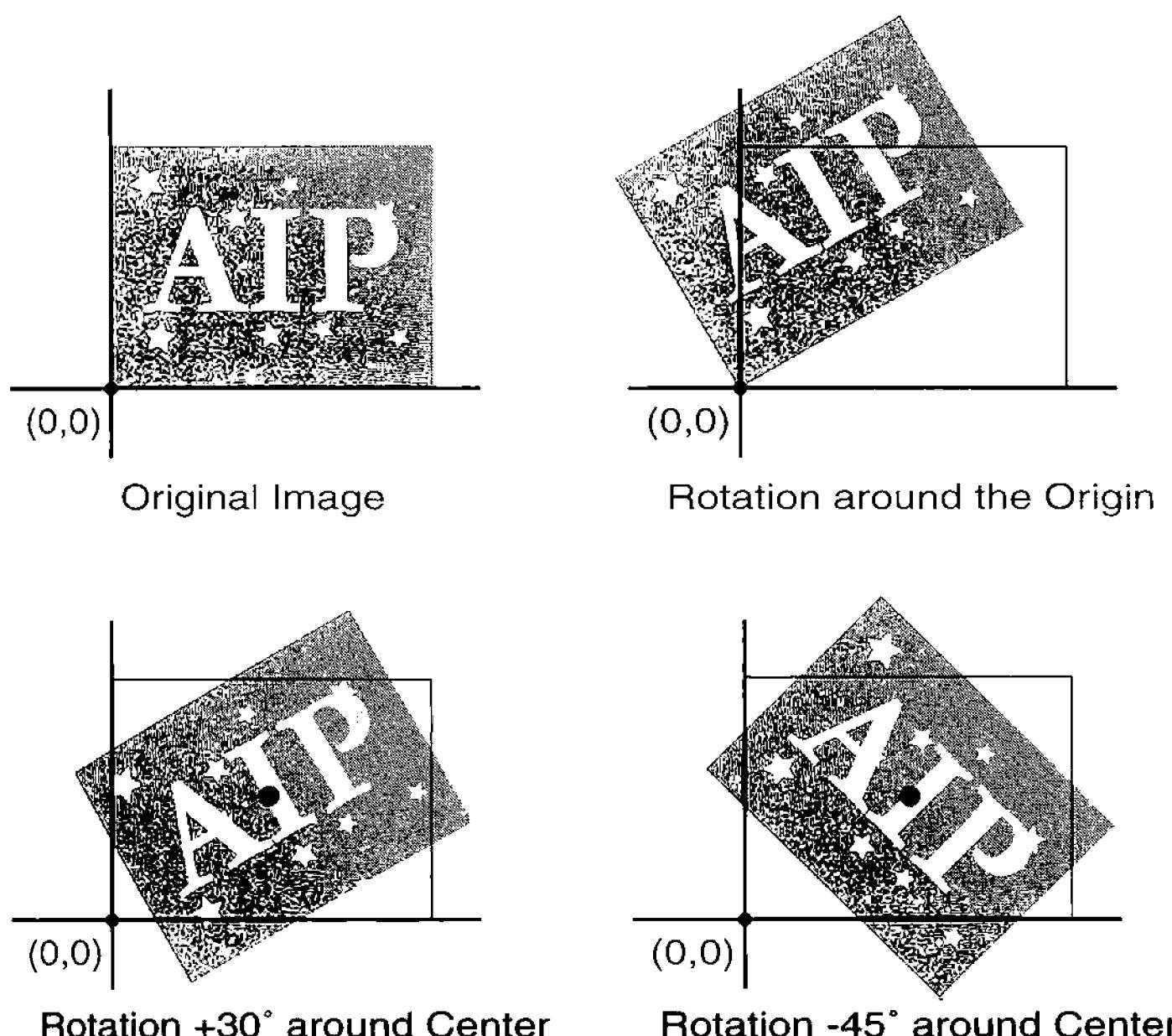
These equations yield the location of the new pixel, but to compute a new image, you need the inverse transform: that is, given the location of a new pixel, where was the original pixel in the original image? The inverse transform is:

$$\begin{aligned} x &= x_0 + (x' - x_0) \cos \vartheta - (y' - y_0) \sin \vartheta \\ y &= y_0 + (x' - x_0) \sin \vartheta + (y' - y_0) \cos \vartheta. \end{aligned} \quad (\text{Eqn. 12.5})$$

Employing these equations, the rotation procedure—taking the argument  $\text{th}$  for the rotation angle,  $x_0$  for the  $x$ -axis center of rotation, and  $y_0$  for the  $y$ -axis center of rotation—looks like this:

```
PROCEDURE ROTATE (th, x0, y0)
  sinth = SIN(th)
```

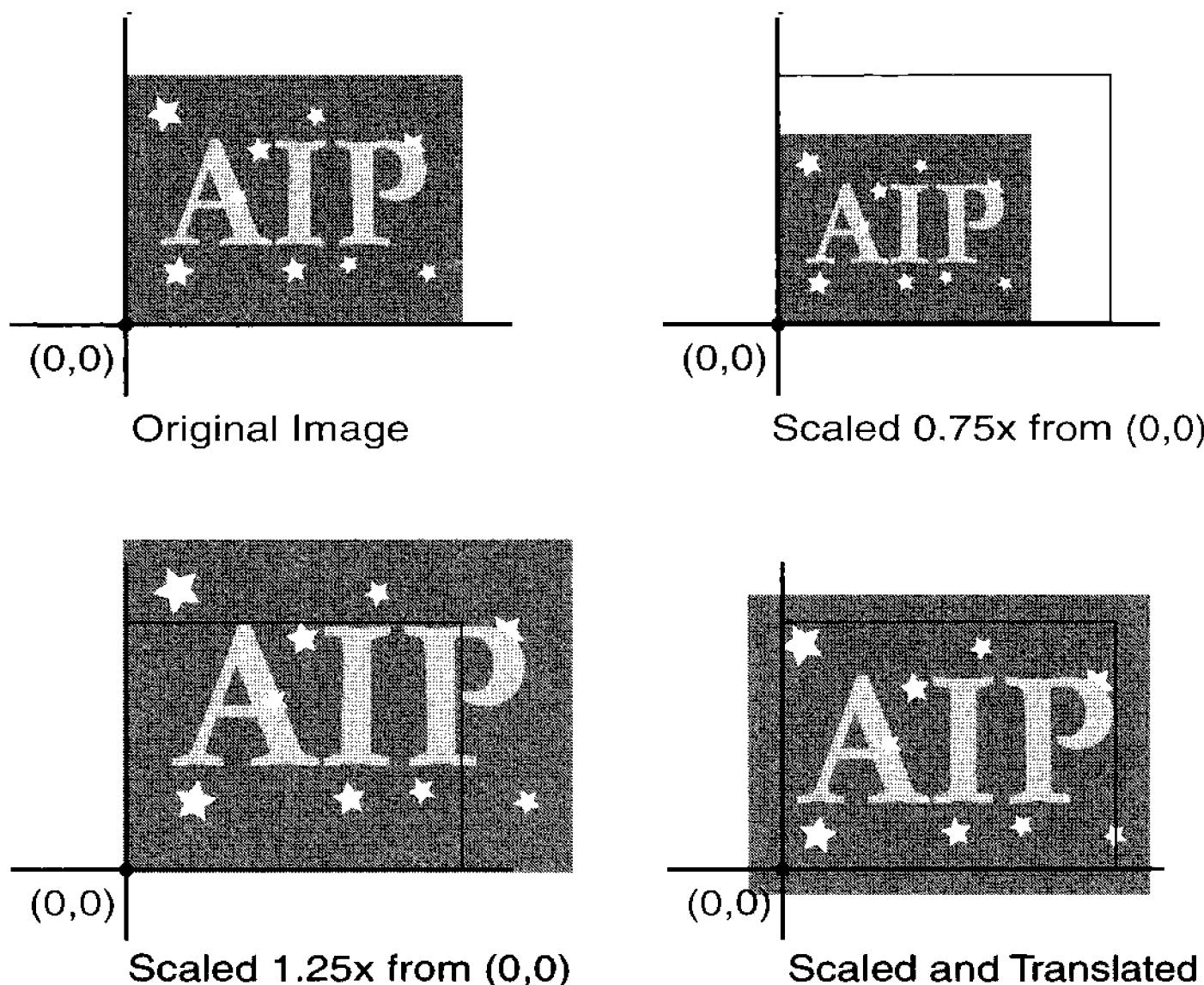
## Section 12.2: Rotation



**Figure 12.4** Rotation can be a bit tricky to understand because we tend to assume that it occurs around the center of the image. Not so. If the center of rotation is not specified in the rotation equations, rotation takes place around the origin of the image coordinate system, usually in a corner of the image.

```

costh = COS(th)
FOR yp = 0 to ymax
    FOR xp = 0 TO xmax
        x = x0 + (xp-x0) * costh - (yp-y0) * sinth
        y = y0 + (xp-x0) * sinth + (yp-y0) * costh
        IF INT(x)<0 OR INT(x)+1>xmax OR_
            INT(y)<0 OR INT(y)+1>ymax THEN
                new(xp,yp) = 0
            ELSE
                xf = x - INT(x)
                yf = y - INT(y)
                a = old(INT(x), INT(y))
                b = old(INT(x), INT(y)+1)
                c = old(INT(x)+1, INT(y))
                d = old(INT(x)+1, INT(y)+1)
                new(xp,yp) = a * (1-xf) * (1-yf) -
                    + b * (1-xf) * yf -
                    + c * xf * (1-yf) -
                    + d * xf * yf
            END IF
    
```



**Figure 12.5** Scaling means changing the size of an image. As with rotation, scaling is carried out relative to the origin of the image coordinate system. If you want to enlarge a small section of an image to frame-filling size, you must specify the center coordinates as well as the scaling factor.

```

NEXT xp
NEXT yp
END PROCEDURE ROTATE

```

Note that it was necessary to compute the sine and cosine functions only once. This results in a tremendous savings in computer time compared to evaluating these slow-to-compute functions for each pixel.

## 12.3 Scaling

The last of the basic image transforms is scaling. Scaling means changing the size of the image. Assuming that the image is to be scaled equally in both axes, the basic equations for scaling a point are:

$$\begin{aligned} x' &= xs \\ y' &= ys \end{aligned} \quad (\text{Equ. 12.6})$$

Note that we are simply multiplying each coordinate by  $s$ , the scale factor. A point at  $(2, 3)$  moves to  $(4, 6)$  when scaled by a factor of 2. Just as with rotation, however, we need to enlarge or shrink the image around the center or around an object of interest. To scale points relative to point  $(x_0, y_0)$ , the equations become:

## Section 12.4: Practical Translation, Rotation, and Scaling

$$\begin{aligned}x' &= x_0 + s(x - x_0) \\y' &= y_0 + s(y - y_0).\end{aligned}\tag{Equ. 12.7}$$

As in the rotation equations, the point  $(x_0, y_0)$  occupies the same location in the new image as it did in the old image.

We can solve the equations above to yield the inverse transformation below. These generate the source coordinates for points in the new image:

$$\begin{aligned}x &= x_0 + \frac{(x' - x_0)}{s} \\y &= y_0 + \frac{(y' - y_0)}{s}.\end{aligned}\tag{Equ. 12.8}$$

Programming the scale transformation is essentially the same as programming rotation, with the exception of the following lines:

```
PROCEDURE SCALE (sc, x0, y0)
```

The parameter `sc` is the scale factor. Of course, we must also replace the rotation equations with the equations for computing scale transform:

```
x = x0 + (xp-x0) / sc  
y = y0 + (xp-y0) / sc
```

This scaling formulation works best for enlarging images. Enlarging an image by a scale factor of 2 or greater means that every pixel in the original image must be sampled as part of a new pixel at least four times, so that almost all of the information contained in the original image is carried forward into the new one.

For scaling factors less than 2.0, pixels in the original image are sampled less than four times. Adjacent pixels must be averaged, thus the image becomes slightly softer. At scaling factors near 1.0, you may see artifacts in the new image because some new pixels are the average of four old pixels, whereas others get their value almost entirely from a single old pixel. In noisy images, the result is a pattern of pixel-averaged smooth regions alternating with unsmoothed regions where the new pixels retain the noisiness of the original. These effects can be minimized by smoothing the entire image with a Gaussian convolution kernel *before* scaling.

At scaling factors less than 0.5, scaling misses some pixels altogether; but because the new image is smaller it may appear sharper despite the loss of image information. For large size reductions, smooth the image with a Gaussian blur before resampling. This insures that every pixel in the old image passes some of its information content into the reduced one.

## 12.4 Practical Translation, Rotation, and Scaling

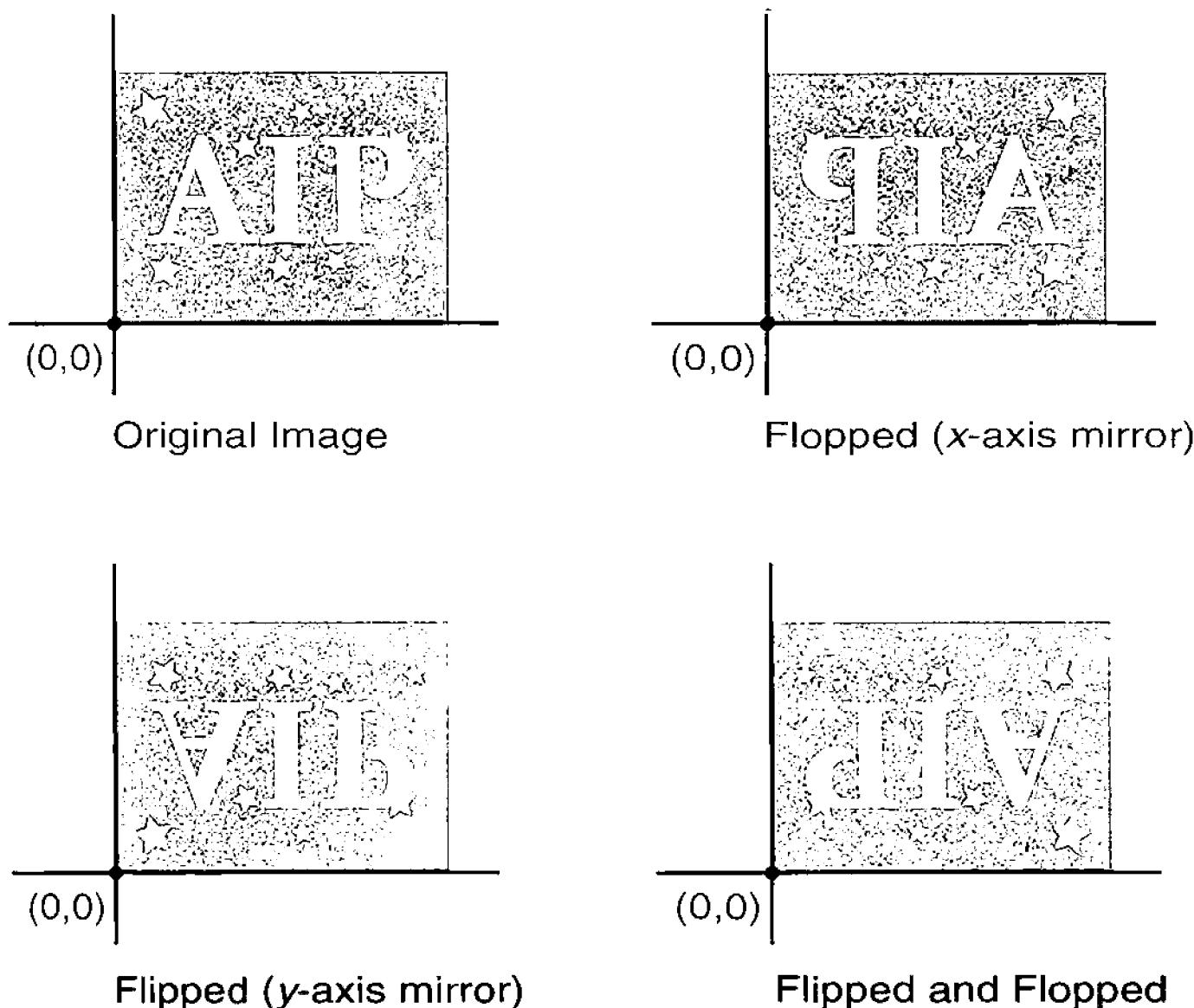
The bulk of practical applications, such as image registration, require three transforms—translation, rotation, and scaling. Although this could be accomplished as

## Chapter 12: Geometric Transforms

three sequential operations, it is more practical to combine the three transforms into one algorithm.

Most important for practical image processing is that the procedure must account for the fact that in many CCDs the pixel dimensions are not the same in the two axes. This is not a problem for translation because all positions in images are measured in pixels; but for proper rotation and scaling, the  $x$  and  $y$  pixel coordinates must be converted into units such as millimeters before the image can be rotated or scaled. Note also that to convert the rotation from degrees to radians, multiply by the conversion factor 0.0174532. In the following, the variables  $xmm$  and  $ymm$  represent the pixel width and pixel height, respectively.

```
PROCEDURE TRANSFORM (xt, yt, x0, y0, th, sc, pw, ph)
    sinh = SIN(.0174532 * th)
    costh = COS(.0174532 * th)
    FOR yp = 0 to ymax
        FOR xp = 0 TO xmax
            x = xp - x0
            y = yp - y0
            x = x - xt
            y = y - yt
            xmm = x * pw
            ymm = y * ph
            xtemp = ((xmm * costh) - (ymm * sinh)) / sc
            ytemp = ((xmm * sinh) + (ymm * costh)) / sc
            x = x0 + xtemp / pw
            y = y0 + ytemp / ph
            IF INT(x)<0 OR INT(x)+1>xmax OR_
                INT(y)<0 OR INT(y)+1>ymax THEN
                new(xp,yp) = 0
            ELSE
                xf = x - INT(x)
                yf = y - INT(y)
                a = old(INT(x), INT(y))
                b = old(INT(x), INT(y)+1)
                c = old(INT(x)+1, INT(y))
                d = old(INT(x)+1, INT(y)+1)
                new(xp,yp) = a * (1-xf) * (1-yf) -
                    + b * (1-xf) * yf -
                    + c * xf * (1-yf) -
                    + d * xf * yf
            END IF
        NEXT xp
    NEXT yp
END PROCEDURE TRANSFORM
```



**Figure 12.6** Mounting your CCD camera on a star diagonal seemed great—until you realized that your images didn’t match the real sky. But software can flip an image top-to-bottom or flop it left-to-right, restoring the orientation that you expect to see. A flip plus a flop each is the same as rotation through  $180^\circ$ .

The procedure above is well-suited to registering CCD images for blinking, registering, stacking, and making movies.

## 12.5 Flipping and Flopping

“Flip” and “flop” are terms used in the graphic arts industry: “flip” to describe interchanging the top and bottom of an image, and “flop” to describe mirroring the left and right sides. Flipping reverses the  $y$ -axis and flopping reverses the  $x$ -axis. Astronomers are familiar with flips and flops because telescopes with an odd number of reflections produce reversed images. You can use flip and flop to restore these images to a standard astronomical orientation.

The basic equations for flipping are:

$$\begin{aligned} x' &= x \\ y' &= -y. \end{aligned} \tag{Equ. 12.9}$$

Unfortunately, these geometrically correct equations place the new image outside the original image, so the image must also be translated along the  $y$ -axis by the width of the image on the  $y$ -axis,  $y_{\max}$ :

$$x' = x \tag{Equ. 12.10}$$

## Chapter 12: Geometric Transforms

$$y' = y_{\max} - y.$$

The basic equations for flopping are the same, except that they apply to the  $x$ -axis. To place the new image within the boundaries of the old image, it must be translated in  $x$ -axis by the height of the image,  $y_{\max}$ :

$$\begin{aligned}x' &= x_{\max} - x \\y' &= y.\end{aligned}\tag{Equ. 12.11}$$

Programming for the flip and flop operations is very simple because the coordinates are integer, so no interpolation is required; and the new image coordinates always lie within the boundaries of the old one.

```
PROCEDURE FLOP ()
FOR yp = 0 to ymax
    FOR xp = 0 TO xmax
        x = xmax - xp
        y = yp
        new(xp,yp) = old(x,y)
    NEXT xp
NEXT yp
END PROCEDURE FLOP
```

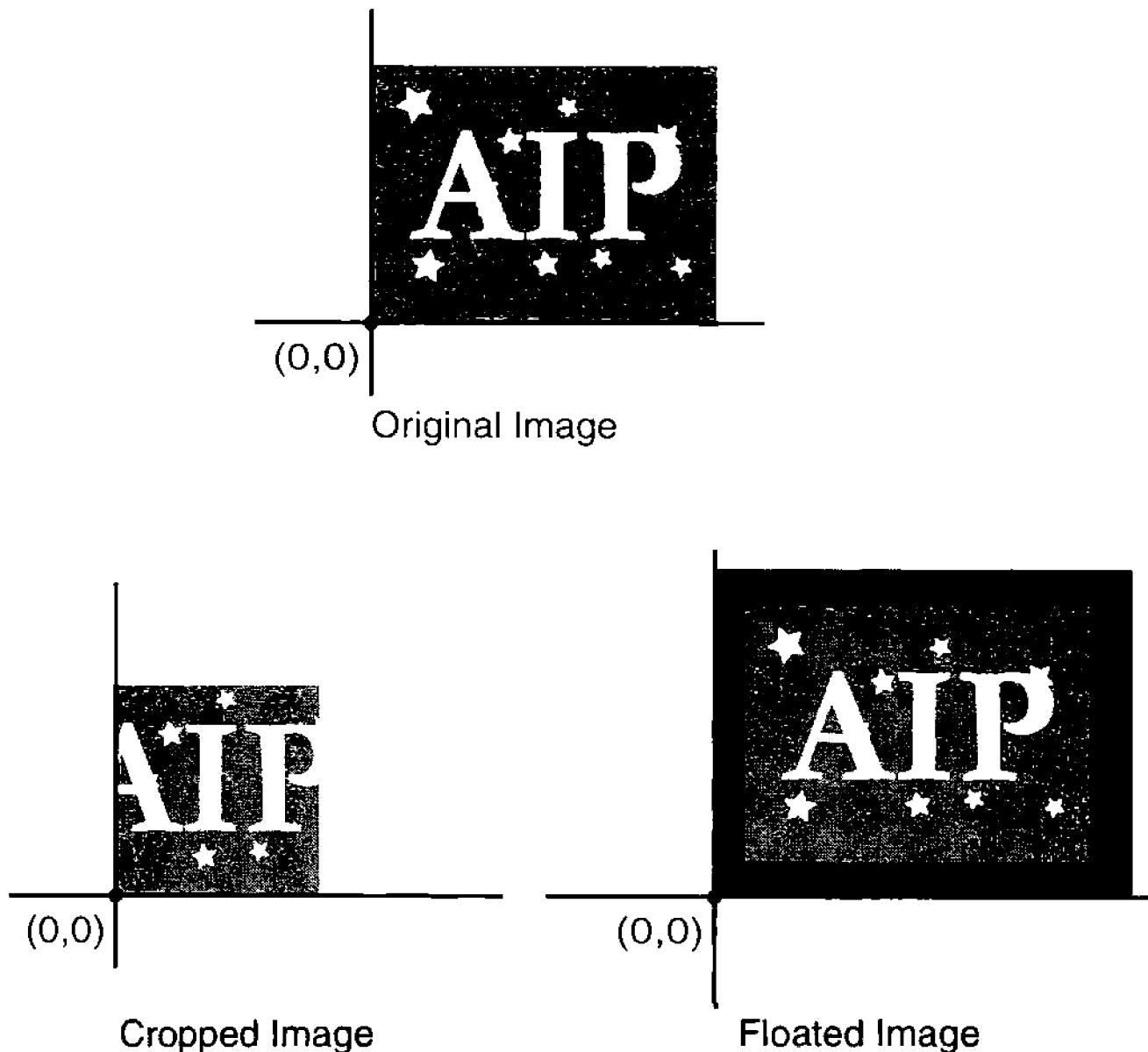
Flipping and flopping have some interesting and useful geometric properties. Flip and flop cannot be replaced by any sequence of translation and rotation, but a flip followed by a flop produces exactly the same result as a  $180^\circ$  rotation about the center of an image. Scaling can replicate flips and flops, but only for negative scaling factors; and the process must be accompanied by an appropriate translation to place the negatively scaled image inside the original boundaries.

## 12.6 Cropping and Floating

Cropping and floating are complementary actions: cropping involves copying part of an old image to a new one, whereas floating involves copying all of an old image into a new and larger one. In both operations, the new image differs in size from the old one.

To crop, you must determine the minimum and maximum values on the  $x$ -axis and the minimum and maximum values on the  $y$ -axis that define the area you wish to copy into the cropped image. Here is a simple procedure for cropping the image `old(0 TO xmax, 0 TO ymax)` given the cropping points `minx`, `maxx`, `miny`, and `maxy`. The procedure assumes that these points lie inside the `old` image.

```
PROCEDURE CROP (minx, maxx, miny, maxy)
xmax = maxx - minx
ymax = maxy - miny
REDIMENSION new (0 TO xmax, 0 TO ymax)
```



**Figure 12.7** Got too much image? Or not enough? Cropping removes edges, clipping away edge artifacts and directing the attention of viewers to specific features that you want them to notice. Floating adds extra rows and columns so that you can rotate or scale without losing the corners of your images.

```

FOR y = 0 TO ymax
    FOR x = 0 TO xmax
        new(x, y) = old(x + minx, y + miny)
    NEXT x
NEXT y
END PROCEDURE CROP

```

Floating is the complement of cropping: to float an image, you copy the entire old one into a new larger image. Floating is useful when you want to rotate an image without losing its corners; the added lines and columns provide a place for the corners to go. The procedure below requires `lox` and `hix`, the number of columns that you want to add to the left and right side, and `loy` and `hiy`, the number of lines that you want to add to the top and bottom.

```

PROCEDURE FLOAT(lox, hix, loy, hiy)
    xmax = lox + oldxmax + hix
    ymax = loy + oldymax + hiy
    REDIMENSION new (0 TO xmax, 0 TO ymax)
    FOR y = 0 TO ymax
        FOR x = 0 TO xmax
            if x >= lox AND x < oldxmax - loy AND_

```

## Chapter 12: Geometric Transforms

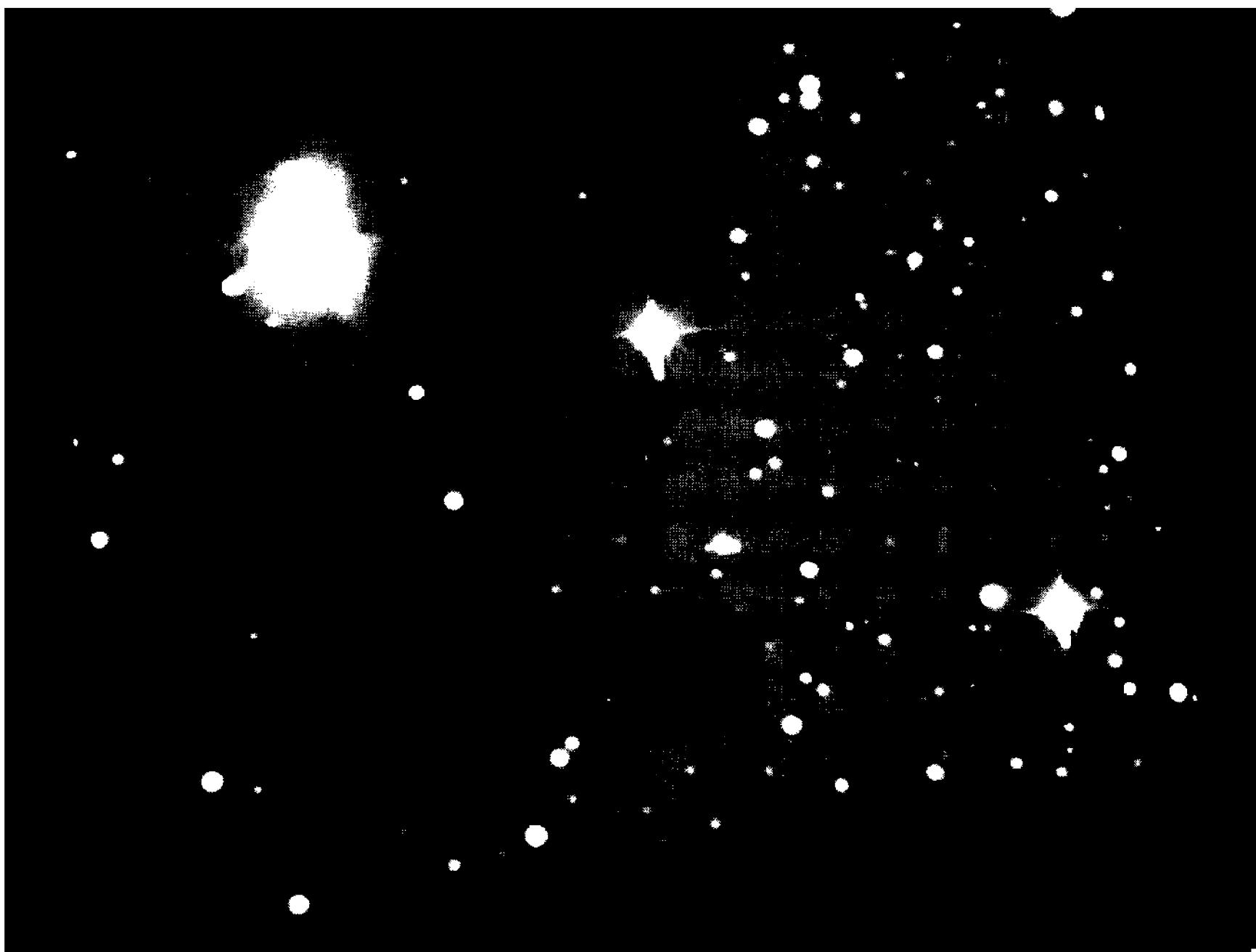


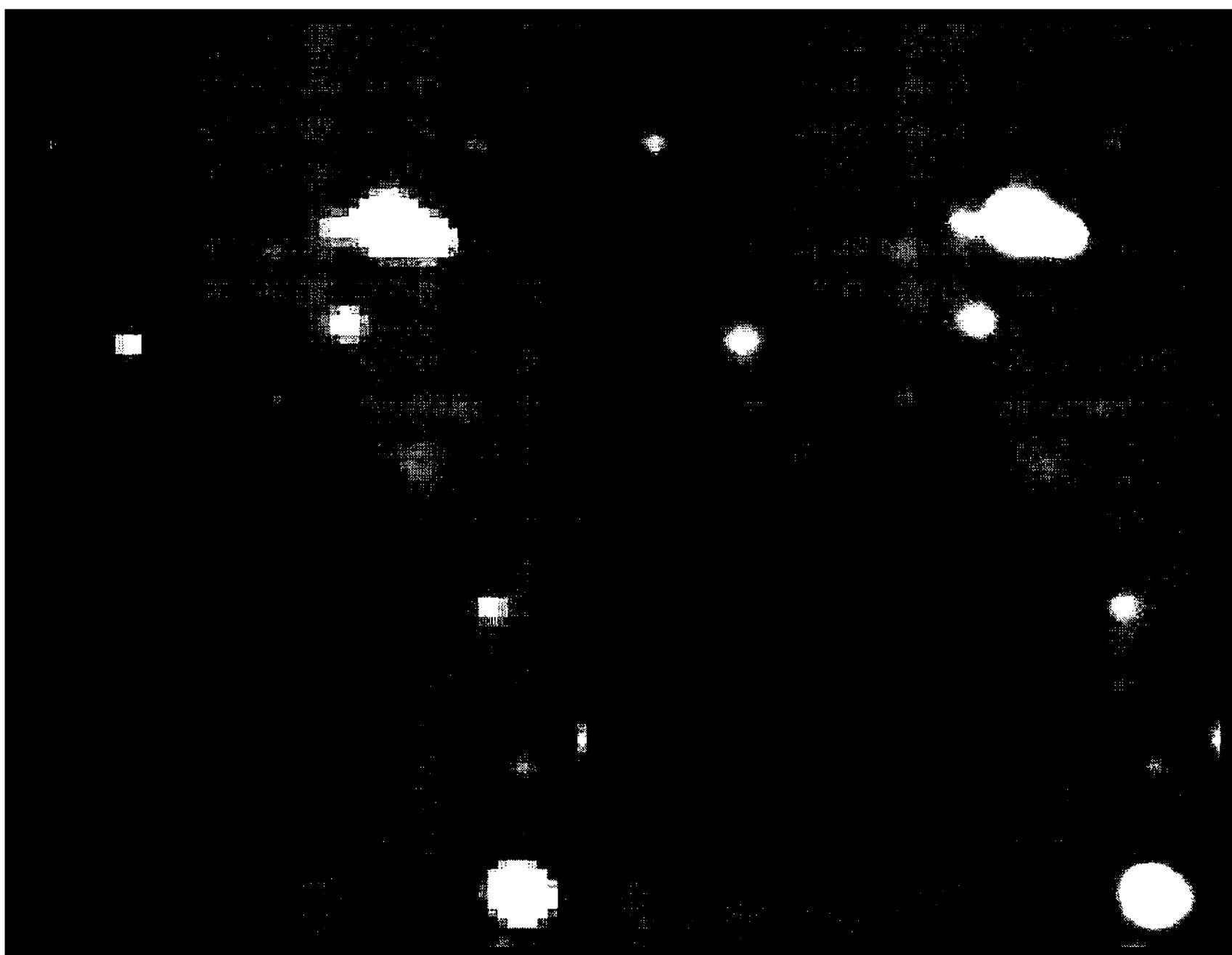
Figure 12.8 This image of the Horsehead Nebula has been floated, translated, rotated, and scaled. Floating added space around the image, translation moved it a few pixels up and to the left, and the rotation and scaling were carried out relative to the center of the image.

```
      y >= loy AND y < oldymax - loy then
          new(x, y) = old(x - lox, y - loy)
      ELSE
          new(x, y) = 0
      end if
      NEXT x
NEXT y
END PROCEDURE FLOAT
```

### 12.7 Resampling

When you take an image, the CCD samples the continuous pattern of light and dark at the focal plane of the telescope, breaking the image into discrete pixels. In resampling, you treat the discrete pixels as points on a continuous flow of light and interpolate a new set of samples; that is, you create a new image by taking new samples—or resampling—the old one.

You can resample to create an image with more pixels than the original by *supersampling*, or with fewer pixels by *subsampling*. While supersampling does not create new information, the interpolated pixels give the image a smooth, rounded appearance, eliminating the blocky “square stars” that many people find



**Figure 12.9** Nobody likes images with that blocky “square stars” look. The cure is to resample the image so that interpolated pixels fill in and smooth boxy edges. In this example, a 60-pixel wide by 80-pixel high Horsehead clip has been resampled to 240-pixels wide by 320 pixels high.

objectionable. Subsampling results in the loss of some information, but can squeeze the image into a smaller screen space and smaller file.

The procedure below resamples the image array `old()` into an image `new()` that is `newwidth` pixels wide and `newheight` pixels high.

```

PROCEDURE RESAMPLE (newwidth, newheight)
  xratio = newwidth / (xmax + 1)
  yratio = newheight / (ymax + 1)
  FOR xp = 0 TO newwidth -1
    FOR yp = 0 TO newheight - 1
      x = xp / xratio
      y = yp / yratio
      xf = x - INT(x)
      yf = y - INT(y)
      a = old(INT(x), INT(y))
      b = old(INT(x), INT(y)+1)
      c = old(INT(x)+1, INT(y))
      d = old(INT(x)+1, INT(y)+1)
      new(xp,yp) = a * (1-xf) * (1-yf) -
                   + b * (1-xf) * yf -

```

## Chapter 12: Geometric Transforms

```
    + c * xf * (1-yf)_
    + d * xf * yf
NEXT yp
NEXT xp
END PROCEDURE RESAMPLE
```

When you multiply the pixel count in the old image by an integer,  $n$ , every  $n$ th pixel goes into the new image without getting averaged with surrounding pixels. If the image is subsequently translated, rotated, or scaled in a procedure requiring interpolation from neighboring pixels, very little information from the original image is lost.

- *Tip:* For maximum quality, before you register images, perform track-and-stack additions, or blink image pairs in **AIP4Win**; we recommend that you resample the image to double its original dimensions.

---

# 13 Point Operations

---

Point operations are important because astronomical images often span an extremely wide range of brightness, but important features in them often span a very narrow range of brightness. Alternatively, interesting features span a range of brightness too great for the limited range of hard copy or computer monitor. A well-chosen point operation can make hard-to-see features more visible by changing their original pixel values to a more appropriate range of pixel values.

In this chapter, you will learn where to find useful information tucked away in a long range of pixel values—and basic techniques that help you change the brightness scale of an image to display that information clearly. There are two parts to the task:

- identifying the range of interesting pixel values, and
- altering the pixel values for optimum display.

To see the information locked inside the image, you need to identify the range of pixel values that contains important information, and then transfer old pixel values to a new range. As you will discover, point operations are the simplest but most powerful tools for processing astronomical images.

## 13.1 Point Operations: An Overview

The four basic types of point operation are distinguished primarily by the method used to identify the numerical range of interest. They are:

- direct function specification,
- direct endpoint specification,
- histogram endpoint specification, and
- histogram specification.

The most basic method of changing pixel values is *direct function specification*. In this method, you specify how the software should change the pixel values in the image, using a mathematical rule such as: “subtract 100 and extract the square root of the remainder” without explicitly stating the range of pixel values. In this example, if, after subtracting 100, the pixel value is negative, it will not be possible to extract the square root and therefore impossible to apply the function.

## Chapter 13: Point Operations

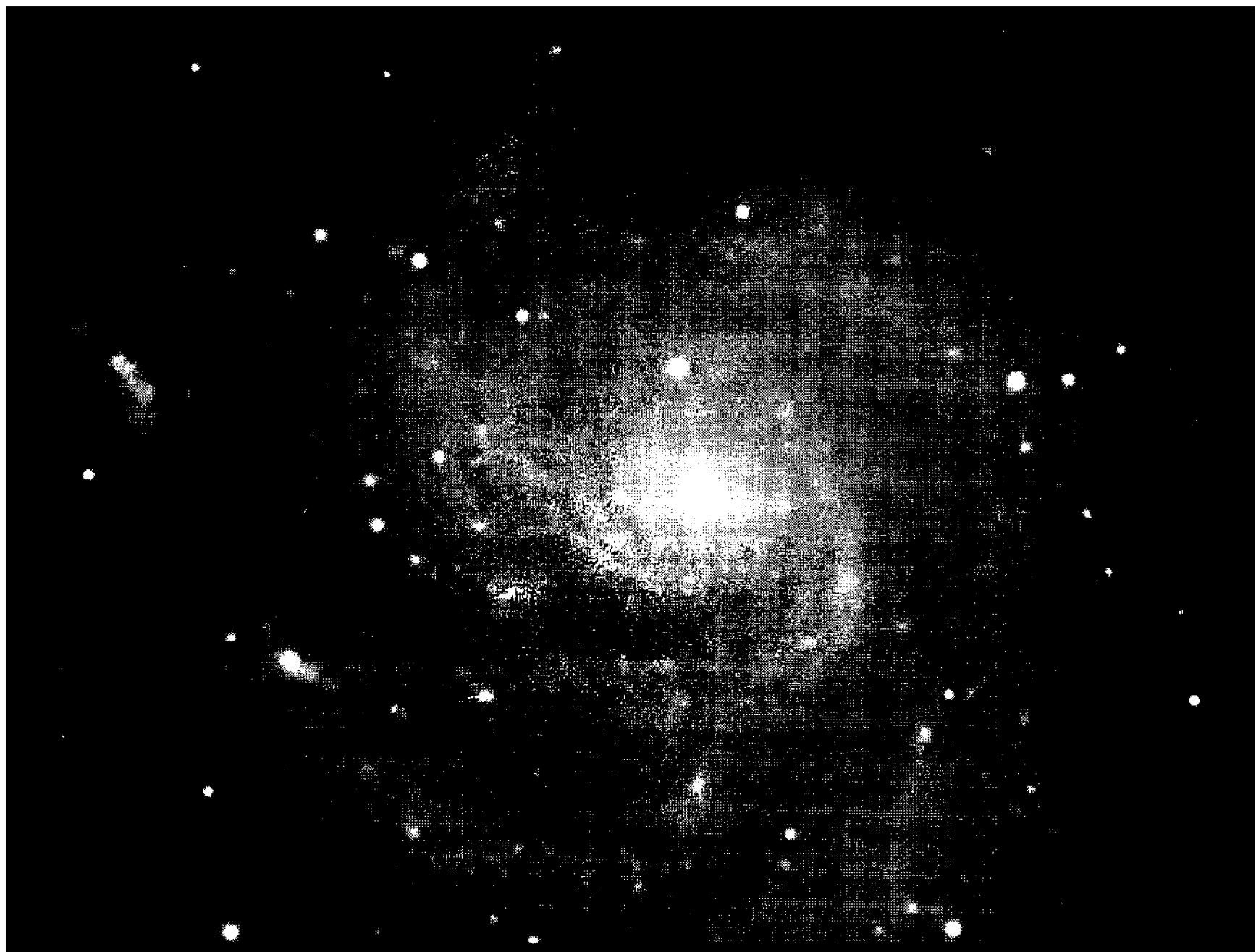


Figure 13.1 This image of the galaxy M101, a 40-minute track-and-stack exposure made by Rob West, shows the object much as it would look to you if you had eyes as sensitive as a CCD chip. On paper, however, the image looks dark. What it needs is a point operation to transform it into a brighter image.

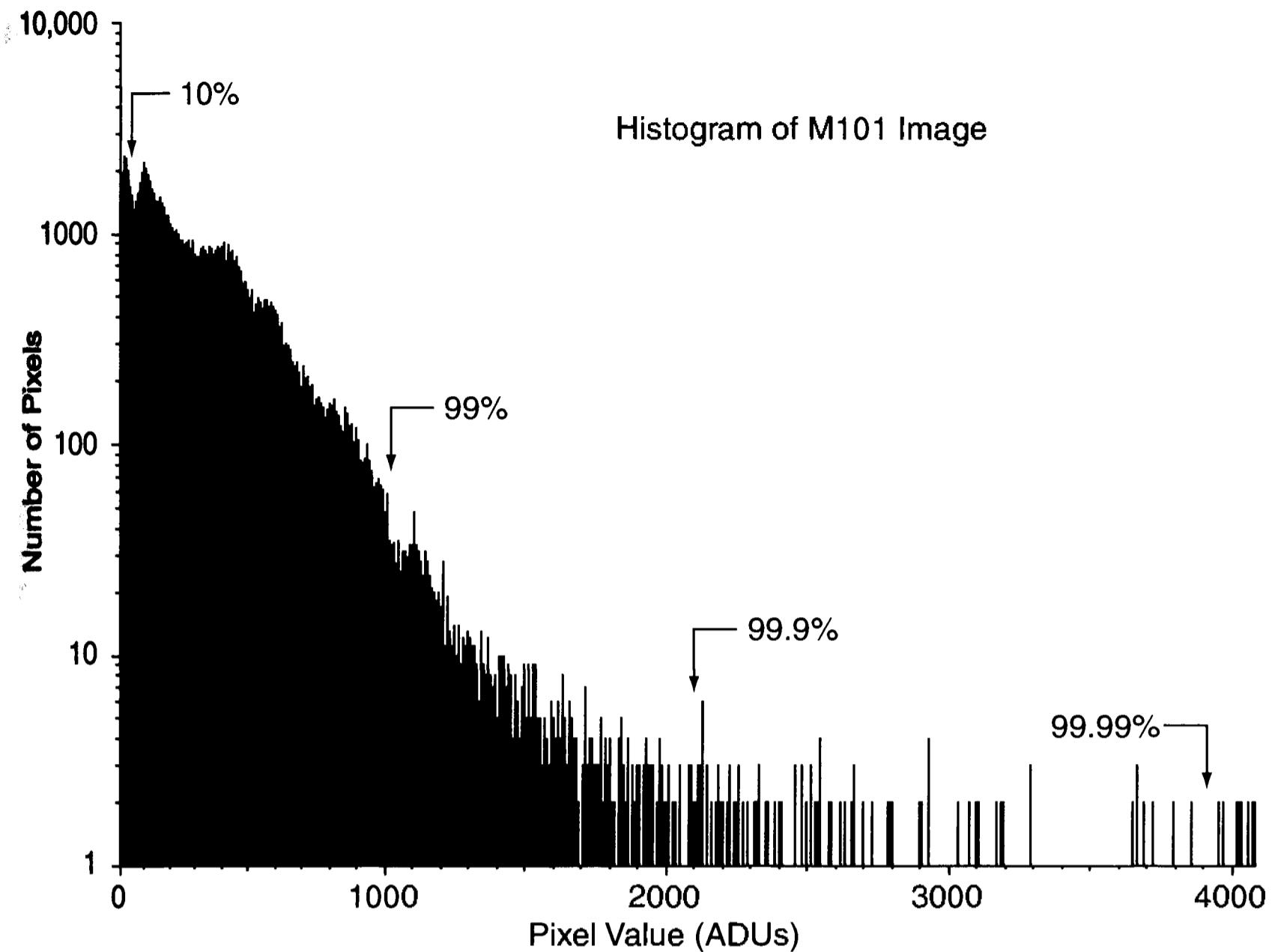
The rule used to convert old to new values is called a *transfer function*. You can transfer pixel values using a linear function; or you can reverse the range, raise pixel values to a power, take their logarithm, raise their logarithm to a power, divide the range of pixel values into steps—anything to make the features that you want to see stand out clearly and distinctly.

*Direct endpoint specification* explicitly separates the point operation into two parts: isolating the range of important pixel values, and converting the values inside the range using a *transfer function*. With direct endpoint specification, you inspect the images and decide the pixel values that will display as black and white; for example, “Set black to 26; set white to 2100,” and you also select a function that is capable of handling the pixel values between those limits.

*Histogram endpoint specification* automates the process of finding the black and white endpoints. Instead of selecting specific pixel values, you specify what percentage of pixels is allowed to saturate to black or white; for example, “Let 0.1% of the image be black; keep 99.9% from saturating white.” The software figures out the black and white pixel values and then applies the transfer function that you have selected to generate the new image.

However, a sophisticated point operation called *histogram specification* or *histogram shaping* automates the choice of transfer function. In histogram shap-

## Section 13.2: Remapping Pixel Values



**Figure 13.2** This is a histogram of the image of M101 in Figure 13.1. Although pixel values in this 12-bit image range from 0 to 4095 ADUs, 99% of them are below 1100, and 99.9% of the pixels have values below 2100. This is why, on a 0-to-4095 scale of pixel value, the image appears so dark.

ing, you specify the desired shape of the new histogram—usually one known to produce good-looking images reliably—and tell the software to generate a transfer function matched to the original image, and then convert pixel values to a new image that will have the histogram you specified.

In the following sections, we explore a variety of transfer functions and how the different methods of endpoint specification operate.

- **Tip:** In **AIP4Win**, you'll find point operations in the *Enhance* menu under *Pixel Math*, *Pixel Ops*, *Brightness Scaling* and *Histogram Shaping*. *Pixel Math* is a tool for direct specification. *Pixel Ops* gives you access to pre-programmed transfer functions. *Brightness Scaling* is a powerful tool that supports both direct and histogram endpoint specification with a handy preview function. The *Histogram Shaping* tool gives you easy access to a variety of histogram profiles, each with previewing.

## 13.2 Remapping Pixel Values

After an image is calibrated, the pixel values in the image are directly proportional to the amount of light that fell on the CCD. In the case of scanned photographs,

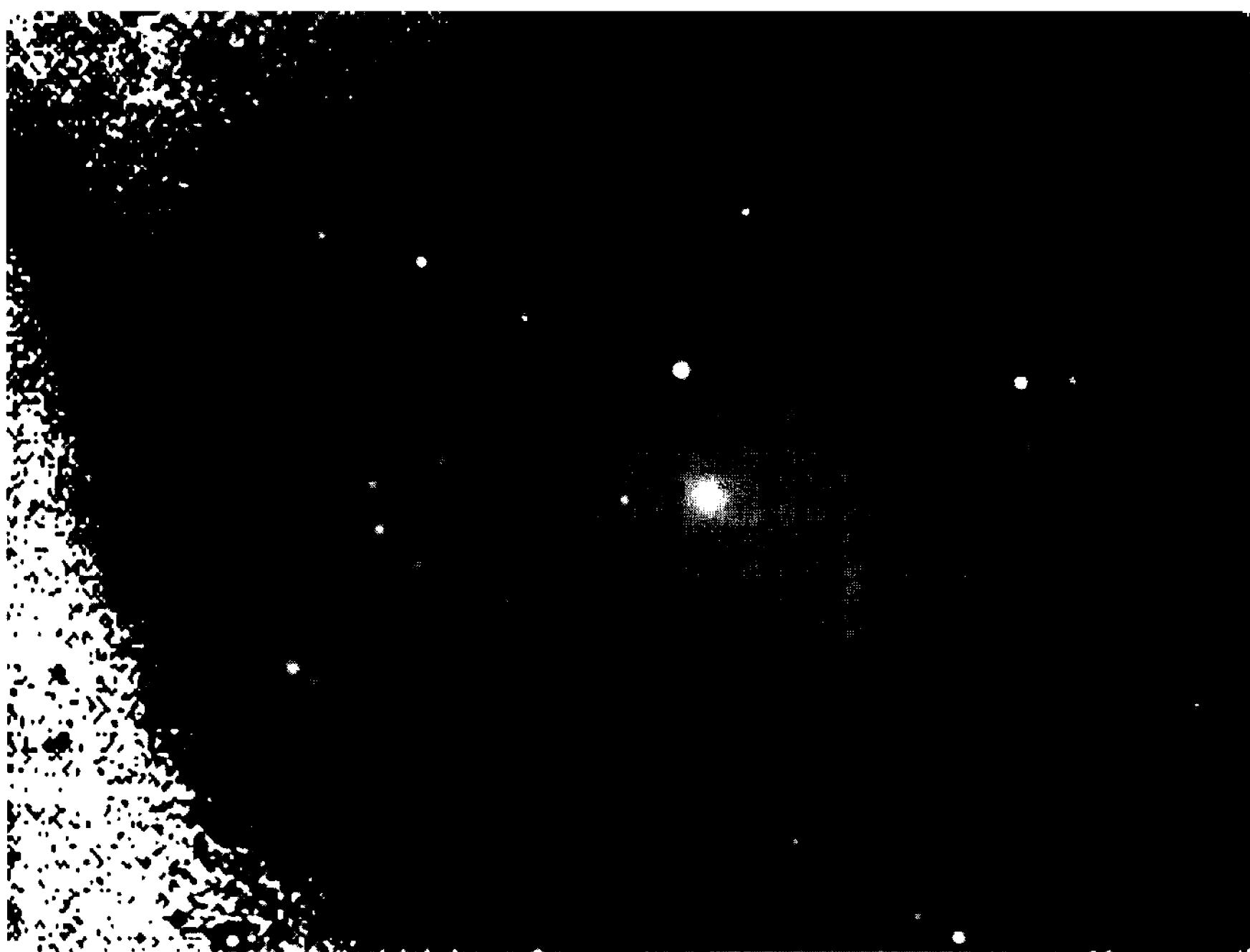


Figure 13.3 In this image, a low endpoint of 0.10 allows the lowest-valued 10% of the pixels to saturate black. To show where information has been lost, black-saturated pixels are printed white. Because these pixels represent sky, their loss may pass unnoticed—but the information they once contained is gone.

the pixel values bear some directly traceable relationship to the light that fell on the film. At this stage, the image contains more information than it ever will again—because every subsequent image-processing operation inevitably destroys some of the information present in the original image. What is essential to remember is that subsequent operations enhance the *visibility* of information that you—the image processing practitioner—decide to emphasize.

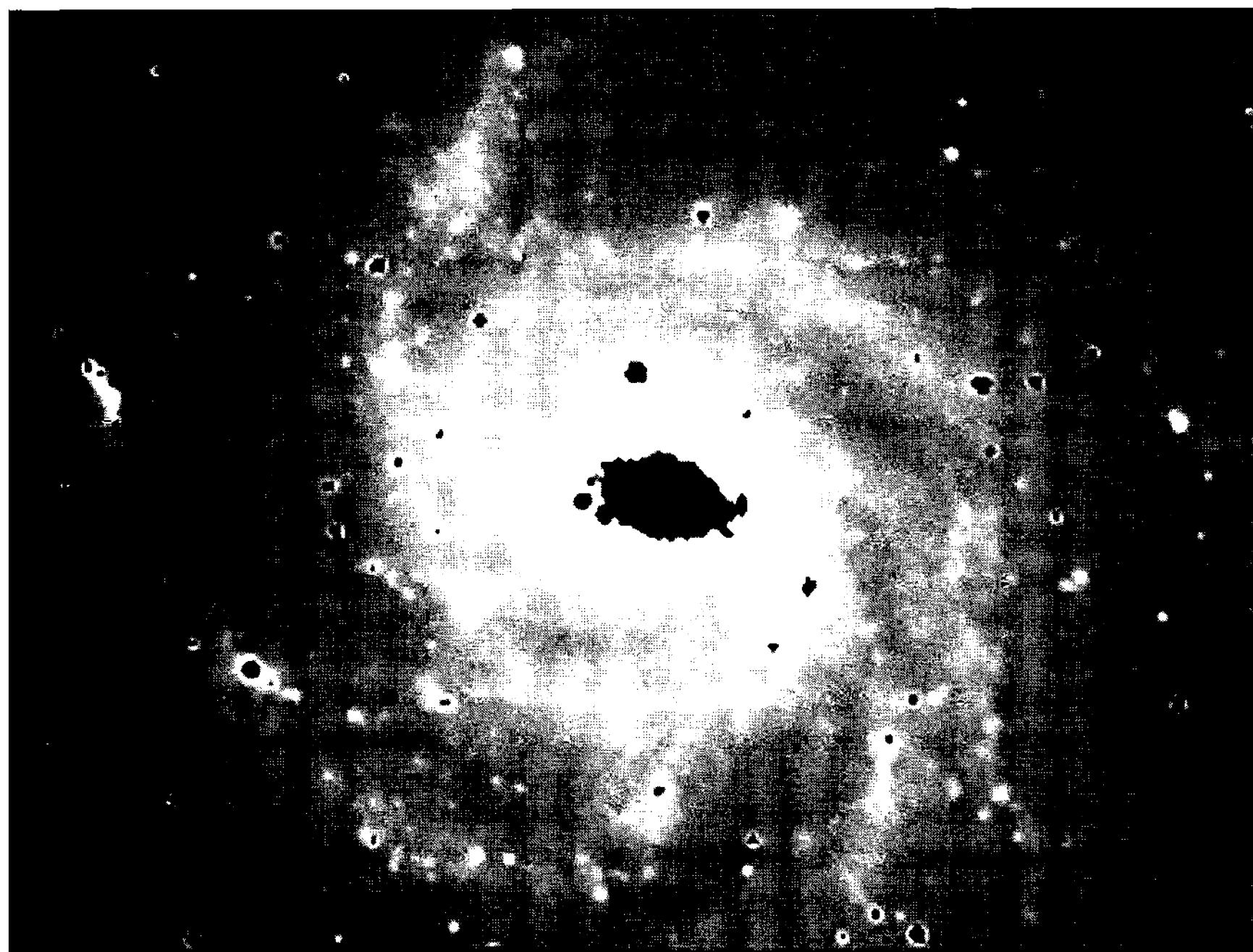
### 13.2.1 Isolating the Range

The first step in deciding how to remap the pixel values in an image is to determine the range of useful pixel values. The image histogram gives a quick look at their distribution. As an example, consider the image of M101 in Figure 13.1, and its histogram shown in Figure 13.2. You can see right away that the vast majority of pixels in the image have low values.

Histograms are the key to understanding how the values of the pixels that make up an image are distributed over the total range of values available. The computer generates a histogram as follows:

```
FOR y = 0 to ymax
```

## Section 13.2: Remapping Pixel Values



**Figure 13.4** Setting the high endpoint to 0.99 allows 1% of the pixels to saturate white. To emphasize the information lost, white-saturated pixels are shown black in this example. With “only” 1% of the pixels in the image allowed to saturate, it is clear that after scaling significant areas in the galaxy, they contain no information.

```
FOR x = 0 to xmax
    pv = image(x, y)
    hist(pv) = hist(pv) + 1
NEXT x
NEXT y
```

where `image()` is the array containing the image, and `hist()` is the array containing the histogram. After the code fragment above has run, the `hist()` array contains the number of pixels having each pixel value. Histograms like the one shown in Figure 13.2 are graphs of `hist()` arrays.

Because the number of pixels with a given value can range from zero to as many as the image contains, the number of pixels is often plotted on a logarithmic scale, as in Figure 13.2. At the low end of the scale, the graph shows 2,000 to 3,000 pixels per pixel value; but at the high end, it often shows only 0, 1, or 2 pixels per pixel value. If it were not plotted logarithmically, the number of pixels at the high end of the graph would not be visible.

Next consider what the histogram shows. Pixels that contribute to the high counts at the low end come from the background sky and the faint outer spiral arms. Pixels at the high end belong to stars and the brighter parts of the galaxy. It is fairly easy to set a pixel value that should display as black. If you allow 10% of

## Chapter 13: Point Operations

the pixels to saturate black, a large section of the sky will be black. You can see this in Figure 13.3, where the lowest 10% of pixels are shown in white. If you allow only the lowest 1% of pixels to saturate black, the black ones will be scattered around the background sky and their loss will hardly be noticed.

Setting the white-saturation value is much trickier. In this image, 99% of pixels lie below 1108 ADUs, 99.9% lie below 2100, and 99.99% lie below 3954. If you select 99%, then the remaining 1% of the pixels will saturate to white. Unfortunately, those pixels depict the nucleus and core of the galaxy. Figure 13.4 shows how much information you lose by setting the white point at 99% (1108 ADUs). The lost pixels are shown black; and they make up the nucleus, core, and centers of all the bright stars.

By setting the black point to 1% (26 ADUs) and raising the white point to 0.9995 (3130 ADUs), very little information is lost either from the background sky or from the center of the galaxy (see Figure 13.5).

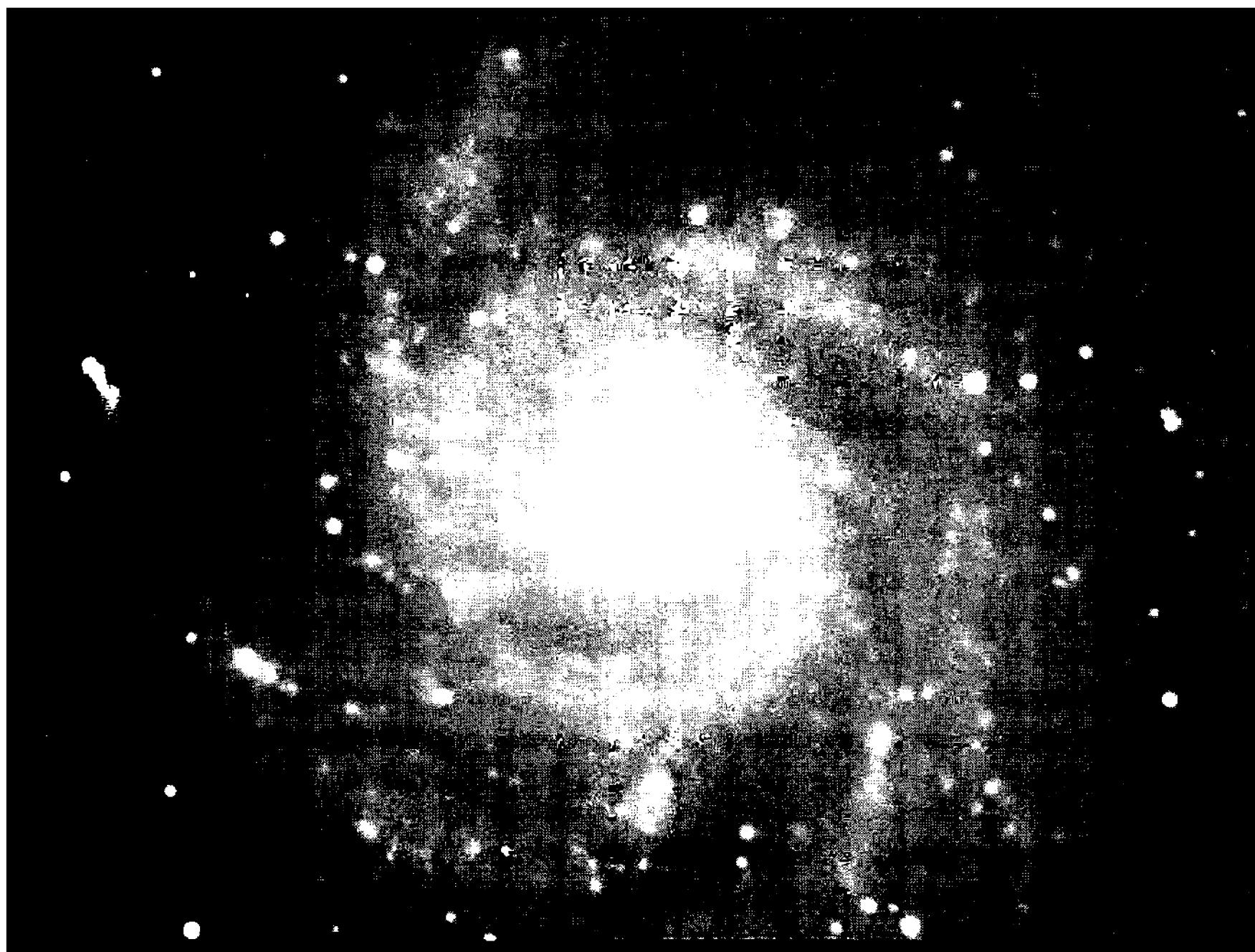
When users specify the black and white points as pixel values, they are doing a *direct* endpoint specification; when users specify the endpoints as percentages of saturated pixels, that is *histogram* endpoint specification. While it is possible to do direct endpoint specification from a plotted histogram, it takes a fair amount of skill and experience to produce consistent results. With histogram endpoint specification, however, novices can produce consistent results by sticking with appropriate endpoints, such as 0.01 and 0.999 for deep-sky images and 0.1 and 0.9999 for planetary and lunar images.

Given histogram endpoints, a simple algorithm analyzes the histogram to find the direct endpoints for an image, as follows:

```
total = (xmax + 1) * (ymax + 1)
pixels = endpoint * total
sum = 0
FOR pvend = 0 TO pvmax
    sum = sum + hist(pvend)
    IF sum => pixels THEN EXIT FOR
NEXT pvend
```

where `total` is the total number of pixels in the image, `endpoint` is the endpoint as a decimal fraction, `pixels` is the number of pixels you want to be unsaturated, `pvmax` is the maximum pixel value expected in the image, and `sum` is a running total of pixels. As the loop variable `pvend` steps through the histogram array, `sum` holds the number of pixels with pixel values less than `pvend`. When the cumulative sum equals or exceeds the desired number of unsaturated pixels, the computer exits the loop, leaving the desired pixel value in the variable `pvend`. In software, this calculation is done twice, once for the black endpoint, `pvblack`, and again for the white endpoint, `pvwhite`.

Once the low and high pixel values that bracket the range of useful information in an image are determined, the transfer function controls what happens to the pixel values between `pvblack` and `pvwhite`.



**Figure 13.5** Endpoints of 0.01 (1%) and 0.9995 (99.95%) preserve almost all of the image information. Compare this image with Figure 13.1. Detail in the sky background and the galaxy core is visible. Pixel values between 26 and 3130 ADUs were remapped using the gammalog transfer function.

### 13.2.2 Transfer Functions

The transfer function is a mathematical relationship between old and new pixel values. In mathematical notation, a function looks like this:

$$f(p) = q . \quad (\text{Equ. 13.1})$$

You should read this as “function of  $p$  equals  $q$ .” The function is embodied in the operator  $f$ . This operator acts like a black box—when you put the value  $p$  into the black box  $f$ , out pops the value  $q$  according to the function’s formula. Consider a few concrete examples. The expression:

$$f(p) = 1 \quad (\text{Equ. 13.2})$$

means that *whatever* value of  $p$  you put into function  $f$ , the value that comes out is 1. It’s a dull function because the output is always the same. Suppose you see:

$$f(p) = p \quad (\text{Equ. 13.3})$$

This one means that when you put  $p$  into the function, you get out the value of  $p$ . This is a valid function even though it doesn’t change anything. Here’s another:

$$f(p) = p^2 . \quad (\text{Equ. 13.4})$$

## Chapter 13: Point Operations

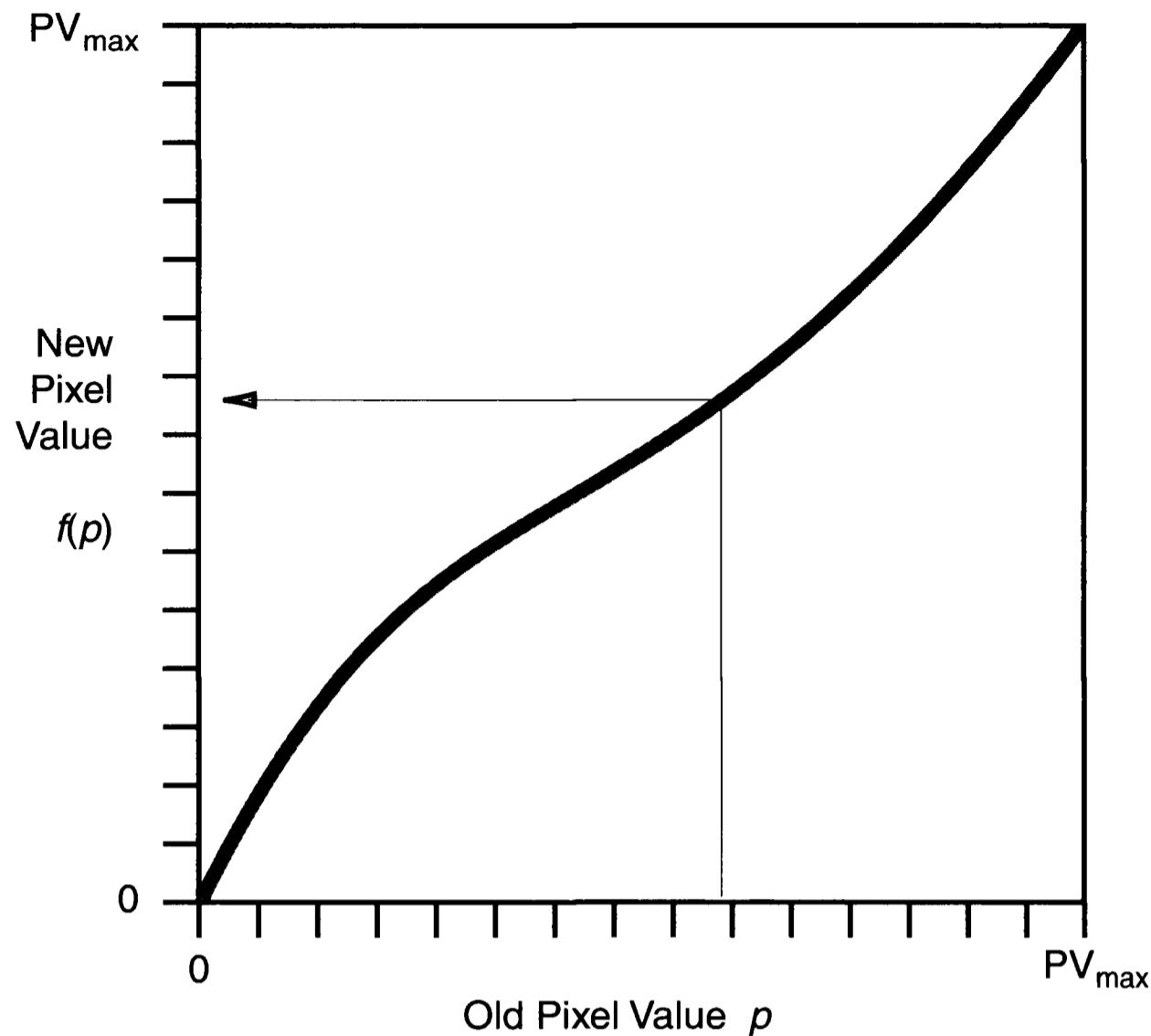


Figure 13.6 The transfer curve is the graph of a transfer function. It depicts the mathematical relationship between original pixel values,  $p$ , and new pixel values,  $f(p)$ . Original pixel values lie between the pixel value in the image that displays as black and the value that displays as white.

This time, when you put  $p$  into function  $f$ , you get out  $p$  squared. Put in 1, you get out 1; but put in 2 and you get out 4. If  $p$  is 25,  $f(p)$  is 625. And so on. The term  $p^2$  tells you how the function  $f(p)$  behaves.

When mathematicians work with functions, they prefer continuous differentiable functions—those that have one output value for each input value and that don't make any sudden jumps. In image processing, however, the transfer function can be anything you can compute. Consider this example:

$$f(p) = \begin{cases} p < 100 \rightarrow f(p) = 0 \\ p = 100 \rightarrow f(p) = 1 \\ p > 100 \rightarrow f(p) = 2 \end{cases} \quad (\text{Equ. 13.5})$$

Here, the function is divided into ranges of good behavior. This notation says: “if  $p$  is less than 100, the value of the function is 0; if  $p$  is 100, then  $f(p)$  equals 1; if  $p$  is greater than 100, the function equals 2.”

In computer languages, such functions are easy to compute. Here is the same function coded as a computer algorithm:

```
FUNCTION F (pv)
SELECT CASE pv
```

## Section 13.2: Remapping Pixel Values

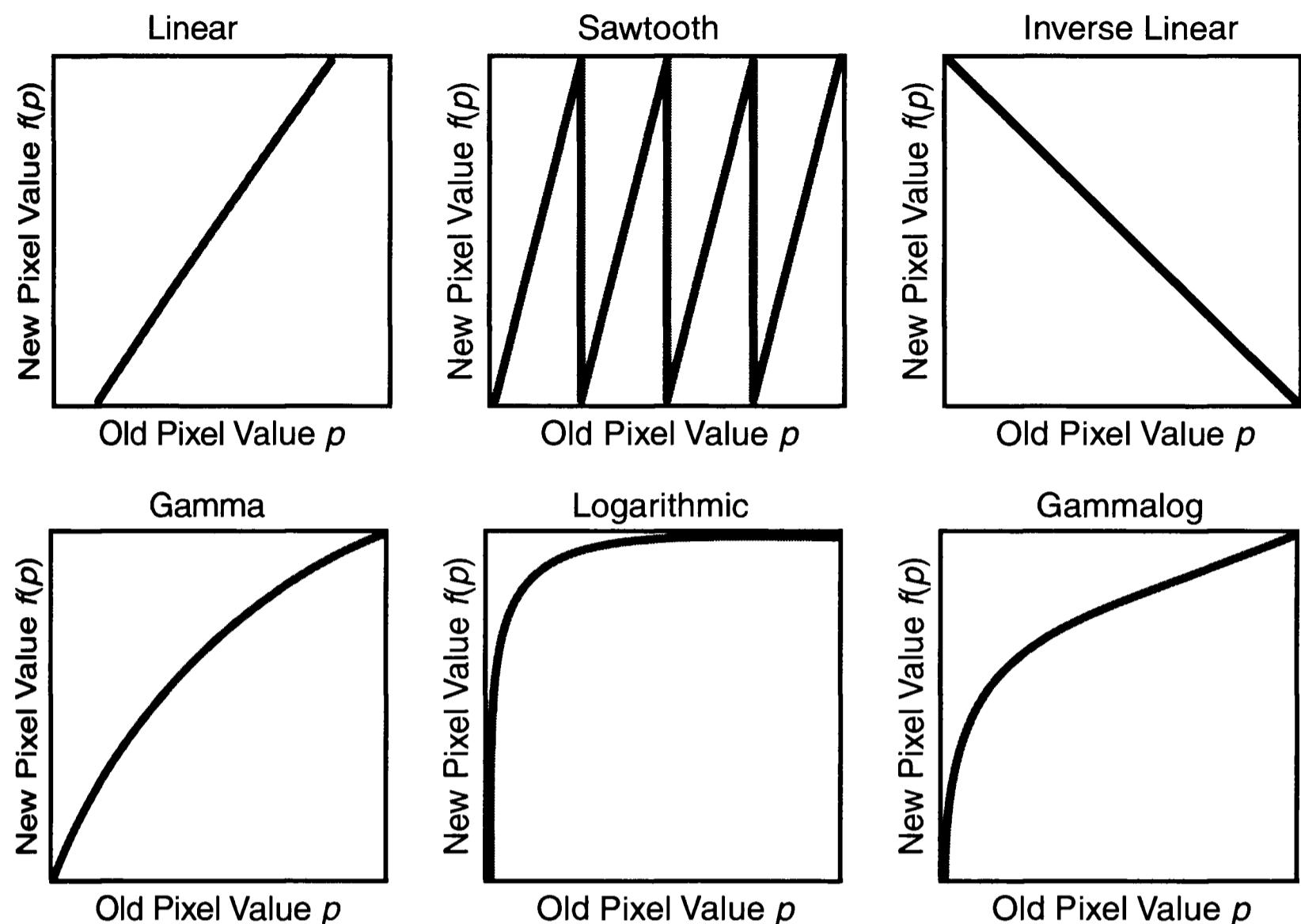


Figure 13.7 The transfer function can assume many different forms—linear, sawtooth, inverse linear, gamma curve, logarithmic, gammalog. The graph can rise, fall, or curve. The only constraint is that for each old pixel value, there must be only one new pixel value—the transfer function must be single-valued.

```

CASE pv < 100
  F = 0
CASE pv = 100
  F = 1
CASE pv > 100
  F = 2
END SELECT
END FUNCTION F

```

When the computer calls  $F(pv)$ , the function computes a new value that depends on the input value of  $pv$ .

In image processing software, transfer functions can be precomputed and the results stored in an array called a *look-up table*. The reason for this is simple: speed. If you are going to determine a new brightness for every pixel in an image that is  $1024 \times 1024$  pixels on a side, you would need to call the transfer function 1,048,576 times. However, if you construct a look-up table for 65,536 pixel values, you have to call the look-up table 1,048,576 times, but each of the lookups is nearly instantaneous.

Look-up tables are called LUTs for short. Here is a short section of a LUT

## Chapter 13: Point Operations

for the function in the example above:

$p$	$f(p)$
97	0
98	0
99	0
100	1
101	2
102	2
103	2

Graphically, the transfer function is usually shown as a curve transversing a square, with the value of  $p$  plotted on the horizontal axis and the value of  $f(p)$  plotted on the vertical axis. The plotted line shows the relationship embodied in the transfer function.

In computer software, the look-up table must be large enough to contain the range of values expected. The look-up table is evaluated using the array index as  $p$  and the value of the array element as  $f(p)$ . In this case, the function  $f$  is the procedure defined above.

```
FOR index = 0 to 65535
    LUT(index) = F(index)
NEXT index
```

where `LUT()` is the look-up table and `F(index)` calls for the function `F` to be evaluated for the value `index`. To apply the look-up table, `LUT()`, the computer takes the value of each pixel in the old image, finds the corresponding value in the array, `LUT()`, and inserts a new value from the look-up table:

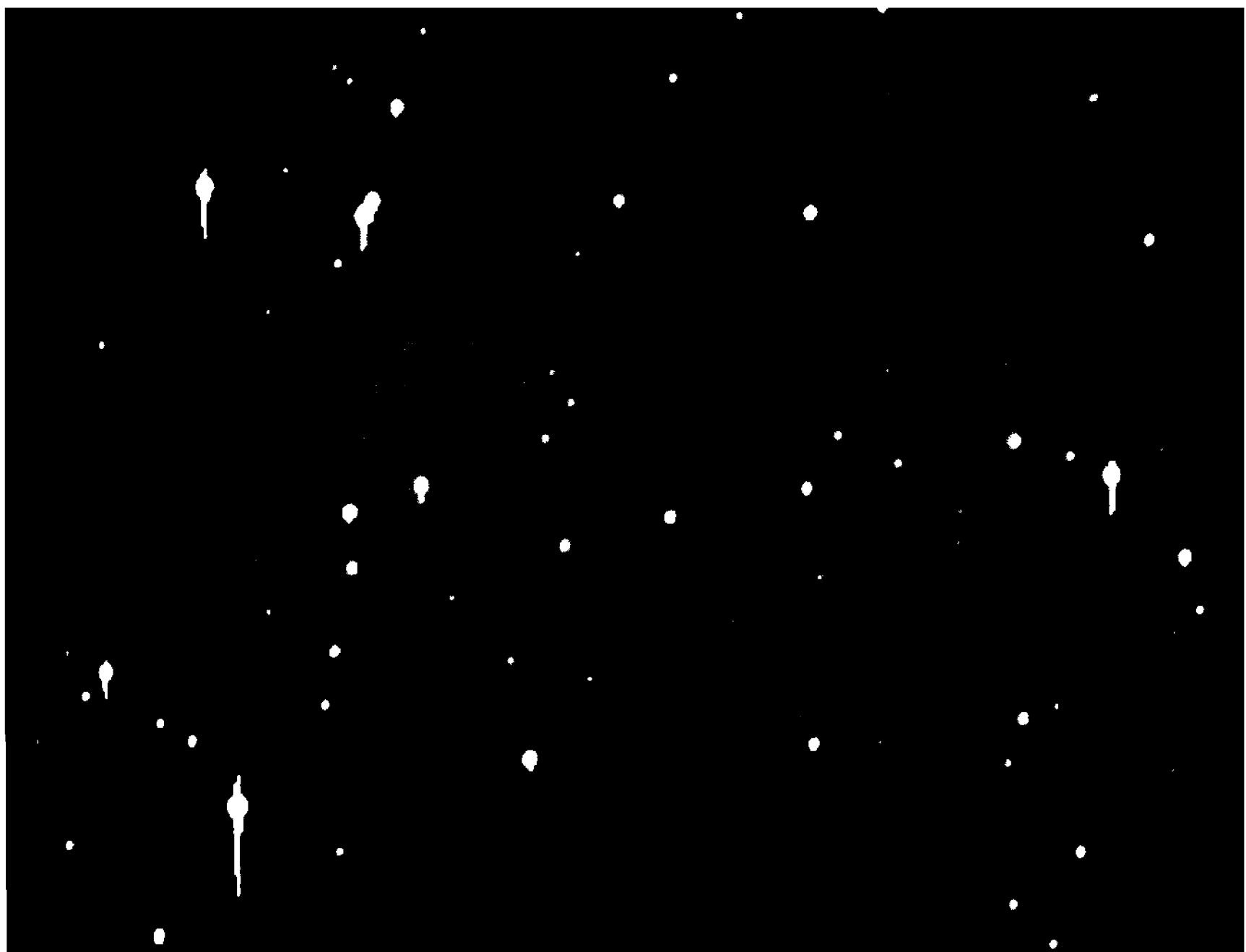
```
FOR y = 0 to ymax
    FOR x = 0 to xmax
        newimage(x, y) = LUT(oldimage(x, y))
    NEXT x
NEXT y
```

where `oldimage()` is the old image and `newimage()` is the new image. Note that you can apply a point operation to an image without creating a new image:

```
FOR y = 0 to ymax
    FOR x = 0 to xmax
        image(x, y) = LUT(image(x, y))
    NEXT x
NEXT y
```

where `image()` is the image that you are processing.

In scientific computing, you may encounter *normalized* pixel values, in which pixel values in an image are scaled into the range between 0 and 1 and treated as floating-point numbers. Normalized pixel values make use of many interest-



**Figure 13.8** Case Study: the Helix Nebula. In this linear stretch, the black pixel value was set slightly below the mean sky brightness, and the white pixel value at twice the brightest value in the nebula itself. This stretch mimics the appearance of the object as it would appear in a dark sky. Image by Neil McMickle.

ing and convenient properties of numbers between 0 and 1, but they cannot take advantage of speedy look-up tables. However, with fast computers, this is no longer as important as it once was.

In direct endpoint specification and histogram specification, the transfer function is applied to the range of values between the black endpoint, `pvblack`, and the white endpoint, `pvwhite`. Although it is possible to use any function at all, the most common transfer functions are the linear, gamma, logarithmic, gammalog, sawtooth, and inverse linear (negative) transfer functions.

- **Tip:** *Earlier versions of AIP for Windows (those with an AIP32 executable file) used a look-up table to speed brightness scaling and image display. With the advent of faster CPUs, **AIP4Win** 2.0 computes functions directly with no loss of speed, but with considerable gain in flexibility and numerical precision.*

### 13.2.2.1 Linear Transfer Function

In the linear transfer function, the pixel values between `pvblack` and `pvwhite` are spread uniformly between the endpoints. The transfer function is a straight line between the black endpoint and the white endpoint:

## Chapter 13: Point Operations

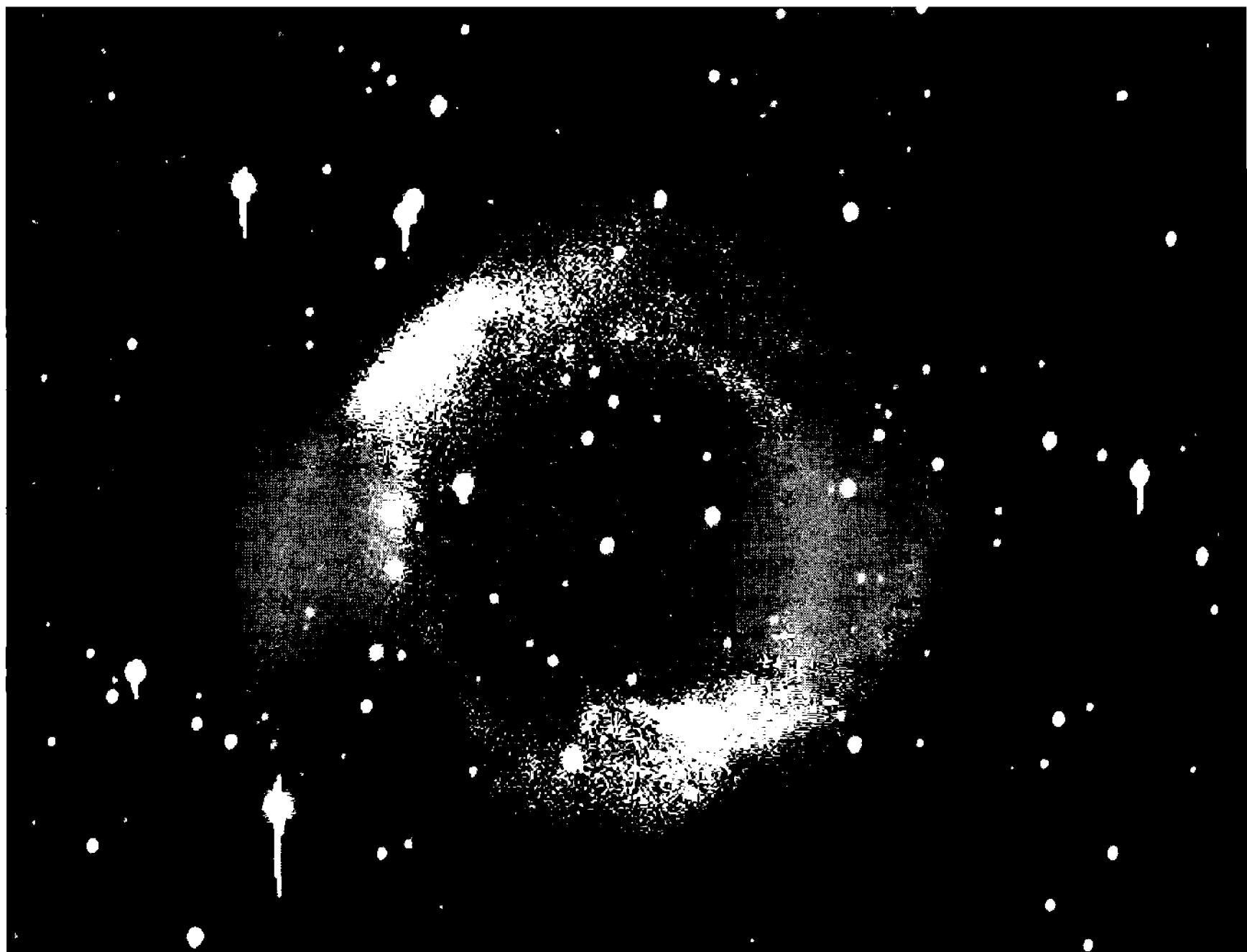


Figure 13.9 Although it has the same endpoints shown in the linearly-scaled image in Figure 13.8, gamma scaling (with gamma = 2) brightens the middle tones that contain the image of the nebula. Gamma scaling is effective for relatively gentle enhancement of deep-sky objects.

$$f(p) = \begin{cases} p \leq p_{\text{black}} \rightarrow f(p) = 0 \\ p_{\text{black}} < p < p_{\text{white}} \rightarrow f(p) = (p - p_{\text{black}}) / \left( \frac{p_{\text{white}} - p_{\text{black}}}{p_{\text{max}}} \right) \\ p \geq p_{\text{white}} \rightarrow f(p) = p_{\text{max}} \end{cases} \quad (\text{Equ. 13.6})$$

where  $p_{\text{black}}$  is the black endpoint,  $p_{\text{white}}$  is the white endpoint, and  $p_{\text{max}}$  is the top of the range of pixel values. To evaluate a linear look-up table between endpoints, the computer carries out the following procedure:

```
PROCEDURE LINEAR (pvblack, pvwhite, LUT())
slope = (pvwhite - pvblack) / pvmax
FOR pv = 0 TO pvmax
    SELECT CASE pv
        CASE pv <= pvblack
            LUT(i) = 0
        CASE pv > pvblack AND pv < pvwhite
            LUT(i) = ((pv - pvblack) / slope)
        CASE pv >= pvwhite
            LUT(i) = pvmax
```

## Section 13.2: Remapping Pixel Values

```

    END SELECT
NEXT pv
END PROCEDURE LINEAR

```

Note that the variable `slope` must be evaluated once, for a significant savings in computer time. After `LUT()` has been evaluated, it is applied to the image as a look-up table.

### 13.2.2.2 Gamma Transfer Function

The gamma transfer function is surely the jack-of-all-trades among point operations. It takes advantage of an interesting property of numbers in the range between 0 and 1, raised to a power. The ends of the range are constant: 0 to any power remains 0, and 1 to any power remains 1. However, raising any number between 0 and 1 to a power generates a smooth curve joining 0 and 1.

When the power is 1, the curve is a straight line, so the transfer function is linear. When the power is less than 1, the curve bows upward; and when the power is greater than 1, the curve bows downward. The gamma transfer function can therefore make the middle range of pixel values in an image appear lighter or darker without significantly changing the darkest or the lightest tones.

The basic gamma transfer function looks like this:

$$f(p) = p^{1/\gamma} \quad (\text{Equ. 13.7})$$

in which  $p$  lies in the range 0 to 1, and  $1/\gamma$  is the exponent. Using  $1/\gamma$  rather than  $\gamma$  as the exponent has the effect of making the midrange increase as  $\gamma$  increases.

To implement this in software, pixel values less than or equal to  $p_{\text{black}}$  are set to 0; those greater than or equal to  $p_{\text{white}}$  are set to  $p_{\text{max}}$ , where  $p_{\text{max}}$  is the value that you want as the maximum in the output image (255 for 8-bit images; 65,535 for 16-bit images).

Pixel values between  $p_{\text{black}}$  and  $p_{\text{white}}$  must be scaled into the range 0 to 1, raised to a power, then scaled to the range between 0 and  $p_{\text{max}}$ , as shown below:

$$f(p) = p_{\text{max}} \left( \frac{p - p_{\text{black}}}{p_{\text{white}} - p_{\text{black}}} \right)^{1/\gamma}. \quad (\text{Equ. 13.8})$$

In addition to its use in processing images, this function is used to correct the image display on computer monitors. Modern video cards use an 8-bit digital-to-analog converter to drive the color guns in the video display tube. In the tube, the electron beam current is proportional to the signal, but the light output,  $L$ , from a typical computer monitor depends nonlinearly on the electron beam current,  $I$ :

$$L = I^{1.8} \quad (\text{Equ. 13.9})$$

where  $L$  is the light output and  $I$  is the electron beam current. The value shown, 1.8, is typical; but the exponent ranges from 1.4 to 3 depending on the monitor. By applying a gamma function to the signal sent to the video card, the nonlinearity

## Chapter 13: Point Operations

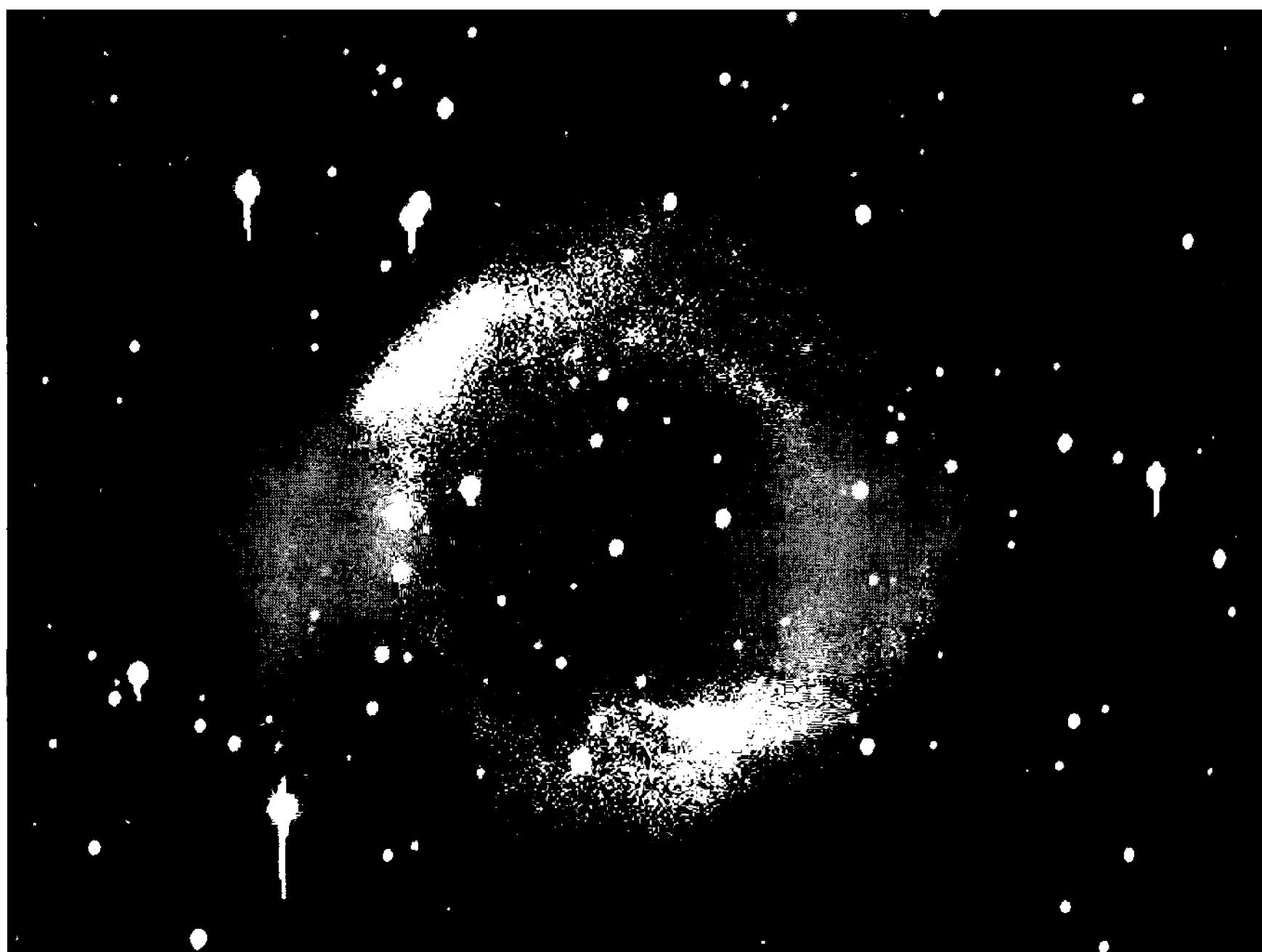


Figure 13.10 Logarithmic scaling increases the lowest pixel values so strongly that the sky background often becomes overly bright. By mixing the logarithm with a linear scaling, logarithmic scaling can produce strong enhancement without loss of detail in the brightest parts of an image.

is cancelled, so that the light output on the monitor is directly proportional to the numerical values in the image.

- **Tip:** In **AIP4Win**, you set the monitor gamma using the *Image Display control*.

### 13.2.2.3 The Logarithmic Transfer Function

The stellar magnitude scale is a logarithmic brightness scale because the human eye and brain system perceives light on a strongly nonlinear and perhaps logarithmic scale. In any event, the logarithm function compresses a large dynamic range of values into a relatively small range—a property that is often desirable in astronomical images.

The common logarithm of 10 to the  $x$  power is  $x$ :

$$\log 10^x = x. \quad (\text{Equ. 13.10})$$

Thus the common logarithm of 1 is 0,  $\log 10$  is 1,  $\log 100$  is 2,  $\log 1,000$  is 3, and so on. In the small space of 0 to 3, you can see that the logarithm holds a dynamic range of 1,000 to 1. It is not necessary to use 10 as the base: natural logarithms use the number  $e = 2.718\dots$ , and, of course, the number of bits in an im-

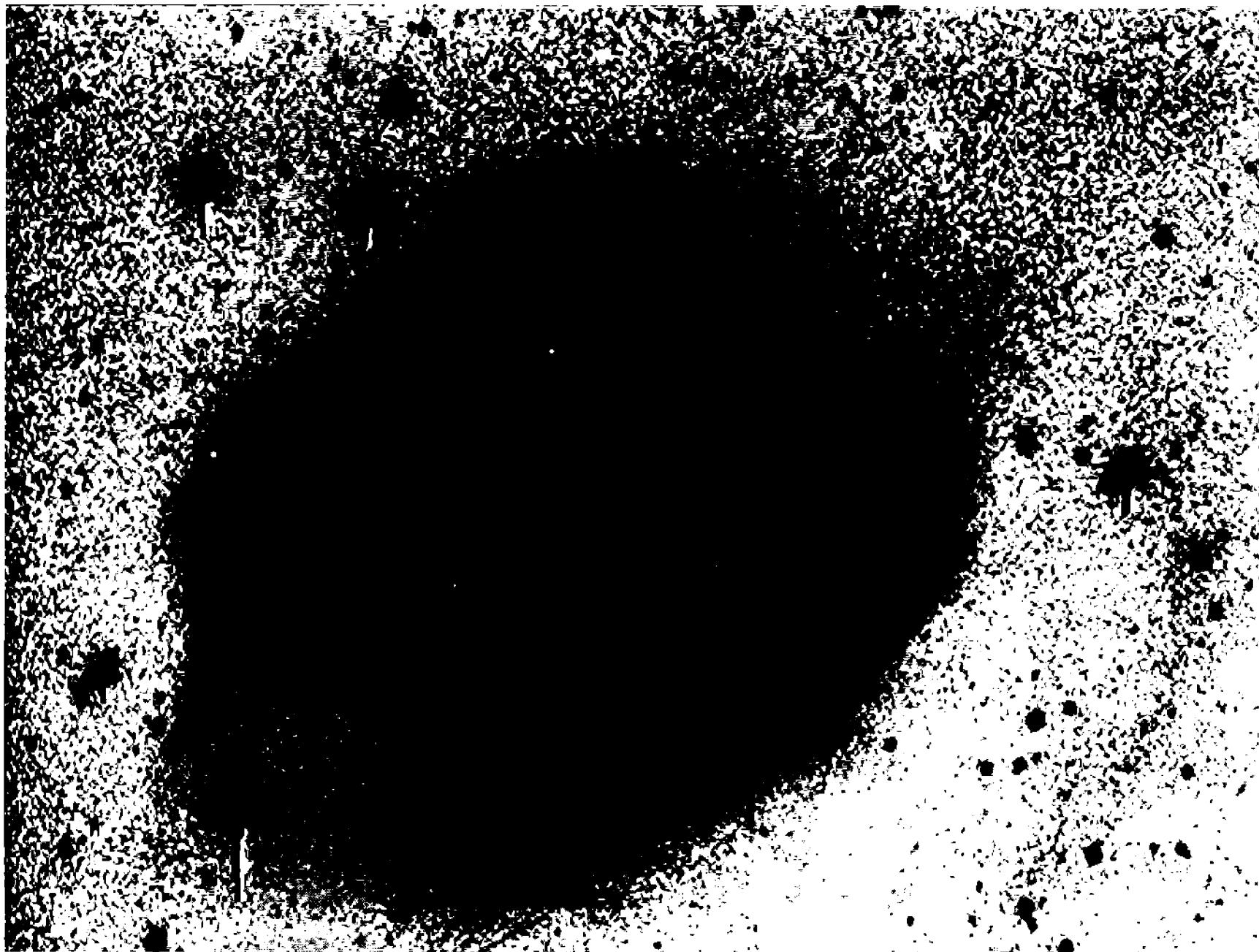


Figure 13.11 Negative scaling changes the emphasis in an image. In this example, the white endpoint was set to the highest pixel values in the nebula and the black endpoint below the sky background—and then the image was negatively scaled. Negative scaling is often used to make faint features stand out.

age is the logarithm to base 2 of the number of gray levels.

To implement a logarithmic transfer function, it is necessary to scale the range of values in the logarithm to match  $p_{\max}$  as follows:

$$f(p) = p_{\max} \left( \frac{\log(p - p_{\text{black}})}{\log(p_{\text{white}} - p_{\text{black}})} \right). \quad (\text{Equ. 13.11})$$

As in previous functions, pixel values less than or equal to  $p_{\text{black}}$  are set to 0; those greater than or equal to  $p_{\text{white}}$  are set to  $p_{\max}$ .

- *Tip: In practical image processing, the logarithm function has a greater dynamic range than needed. To deal with this, AIP4Win allows you to average the logarithm function with a linear scaling, to produce a gentler transfer function.*

#### 13.2.2.4 The Gammalog Transfer Function

The gammalog function is a compound of the logarithm and gamma functions that produces a visually pleasing nonlinear response. The gammalog equation is:

$$f(p) = (\log p)^{1/\gamma}. \quad (\text{Equ. 13.12})$$

## Chapter 13: Point Operations

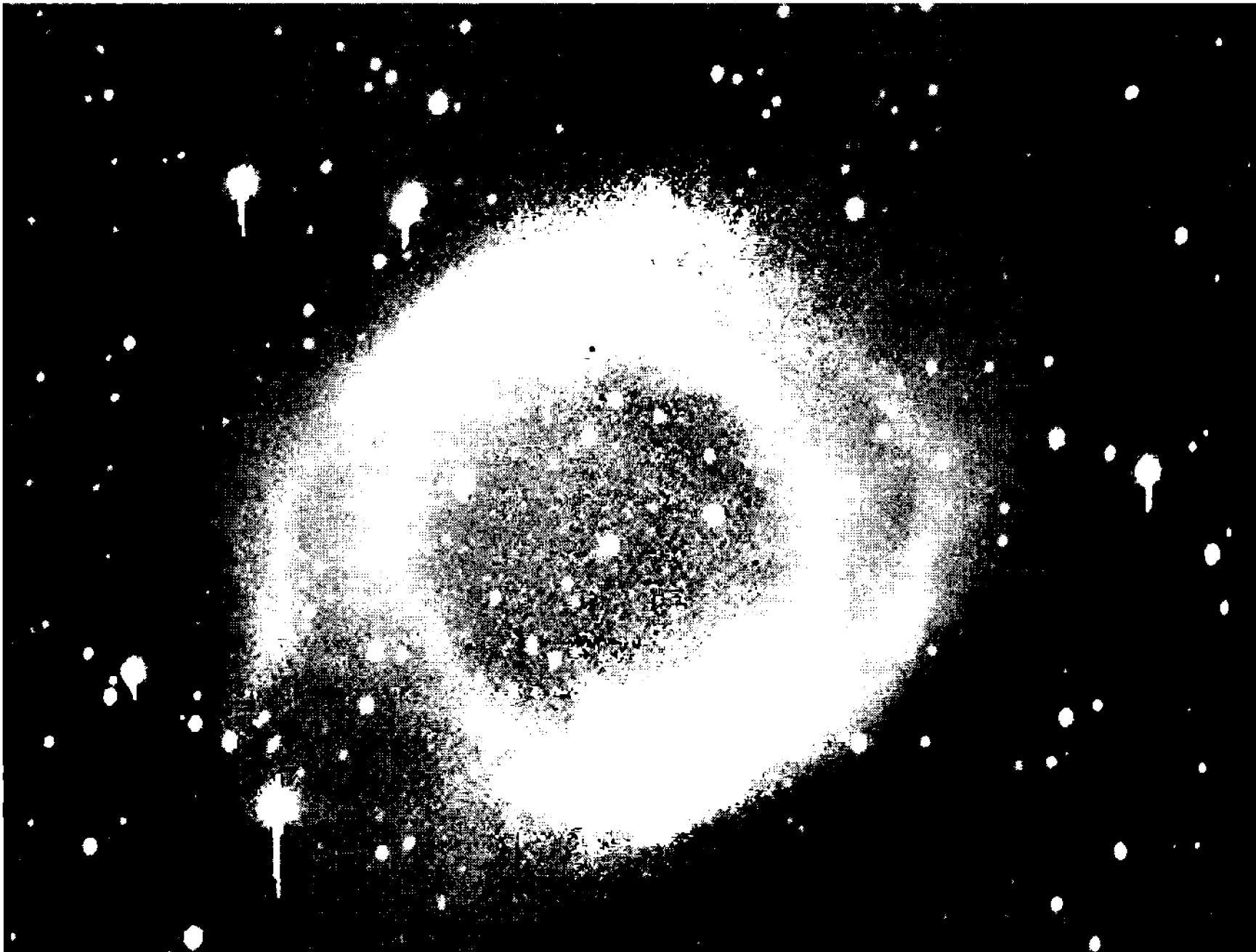


Figure 13.12 Many observers consider gammalog scaling to be the most effective function for deep-sky images. While preserving features in the bright parts of the image and enhancing the middle tones, the gammalog function also enhances faint features nearly buried in the sky background.

When  $\gamma$  is set to values less than 1, the gammalog function increases less rapidly than the logarithm function but more rapidly than a gamma function, and for large values becomes almost linear. With deep-sky images, this behavior strongly enhances the contrast between the sky and the object; but in bright parts of the image, the gammalog function retains image detail better than either the logarithm or the gamma function.

- *Tip:* Used with histogram endpoint specifications of 0.01 and 0.9995, the gammalog function does an outstanding job improving deep-sky images, enhancing faint structures in the object while retaining detail in the bright parts.

### 13.2.2.5 The Negative Transfer Function

Back in the days when professional astronomy relied on photographic plates, astronomers studied negatives and made negative prints. The reason given was that faint details are more visible on a negative—as a slightly darker shade of light gray against a lighter shade of light gray—than they are between two very similar shades of dark gray on a positive print. Negative images also seem to be easier on the eyes, and when you print them with an inkjet printer, they preserve faint details better in hard copy—and use a lot less ink.

### Section 13.3: Direct Specification of the Transfer Function

The negative transfer function is a special case of the linear transfer function:

$$f(p) = p_{\max} - p . \quad (\text{Equ. 13.13})$$

Unlike the previous functions, however, black and white are reversed, so pixel values less than or equal to  $p_{\text{black}}$  are set to  $p_{\max}$ , while those greater than or equal to  $p_{\text{white}}$  are set to 0.

#### 13.2.2.6 The Sawtooth and Quantize Transfer Functions

In astronomy, it is sometimes important to be able to trace the size and shape of a soft-edged feature such as a comet's tail. One effective way to do this is to use a transfer function with abrupt discontinuities, so that the eye can easily pick out isophotes, or regions with the same brightness.

Astronomers have long used two transfer functions for this purpose: the sawtooth function and the quantize function.

The sawtooth function is a set of linear transfers each covering part of the input range of brightness. As  $p$  increases,  $f(p)$  runs through several cycles from 0 to  $p_{\max}$ . Between the discontinuities, the image appears as a linear grayscale; and at each discontinuity, the eye can easily follow the contour created by the sudden change in brightness.

In the quantize function, as the input value rises steadily, the output rises in a series of steps. Between steps, all pixel values appear with the same shade of gray. The steps are clearly visible as sudden changes in the shade of gray, and their edges follow contours of constant brightness.

- **Tip:** *AIP4Win allows you to set the number of “teeth” in the sawtooth function, or the number of “steps” in the quantize function, and to preview the new image before creating it.*

### 13.3 Direct Specification of the Transfer Function

The most obvious approach to transfer functions is to specify mathematical operations that should be applied to the input values to produce the desired output values. For example, the transfer function  $f(p)$  could be specified as:

$$f(p) = 2p . \quad (\text{Equ. 13.14})$$

In this case, the transfer function replaces each pixel with a new pixel having two times the original pixel value. To carry out this operation, the software creates a look-up table using the function to compute  $f(p)$ :

## Chapter 13: Point Operations

$p$	$f(p)$
0	0
1	2
2	4
3	6
...	...
253	506
254	508
255	510
...	...

To use the look-up table, the program looks at each pixel in the image and reads its value,  $p$ . The program then finds the  $p$ th element in the table, and places that new pixel value from the table into the image. Except with very small images, it is faster to look up a precomputed value in a table than it is to compute new pixel values each time.

**Identity transfer function.** As the name implies, the identity function returns the value identical to the value entered; i.e.,  $f(p) = p$ . Although it is a “trivial” case, the identify function is a legitimate member of the class of “image math” transfer functions.

**Add transfer function.** This simple function returns the value entered plus a constant; i.e.,  $f(p) = p + c$ . It can be used to add a constant (negative or positive) to the pixel values in an image. However, to implement even this simple function, the look-up table must be constructed so that all possible new pixel values lie within the range of acceptable values, according to the rule:

$$f(p) = \begin{cases} p + c < p_{min} \rightarrow f(p) = p_{min} \\ p_{min} < p + c < p_{max} \rightarrow f(p) = p + c \\ p + c > p_{max} \rightarrow f(p) = p_{max} \end{cases} \quad (\text{Equ. 13.15})$$

Thus, an operation as simple as adding a constant requires range checking. An advantage of using the look-up table is that range checking need be carried out only once during its construction, rather than for every pixel in the image if the additions were done directly.

The minimum pixel value is normally zero, and the maximum pixel value is normally 255, 4095, or 65535, depending on whether the images contain 8-bit, 12-bit, or 16-bit data.

**Multiply transfer function.** Multiplication by a constant is another elementary transfer function,  $f(p) = kp$ , that must be range-checked so that the output values in the look-up table lie between the minimum and maximum allowable ones. The constant  $k$  can have any value, and may be used to perform division by making  $k$  the reciprocal of the divisor.

**Plus-times-plus transfer function.** A sequence of adding a constant, multi-

## Section 13.4: Direct Endpoint Specification

plying, and adding another constant is a very handy transfer function because it enables the user to remove a known offset from an image, scale the corrected pixel values, and then restore the offset. The function is:  $f(p) = k(p + c_1) + c_2$ .

This sequence is particularly useful for tasks such as scaling a dark frame to match an image made with a different integration time. Dark frames from CCD cameras often have a fixed bias value, such as 100. To match a 300-second dark frame to a 60-second image frame, the bias would be subtracted ( $c_1 = -100$ ), the pixel values in the dark frame divided by 5 ( $k = 0.200$ ), and the bias added back ( $c_2 = 100$ ). This scaled dark frame could then be subtracted from the image.

This function and others like it could obviously be carried out as three sequential operations, but the need for the add-times-add sequence arises so often that it is convenient to have it available as a single operation.

**Brightness-contrast function.** Most general-purpose image-processing programs provide slider bars for adjusting image brightness and contrast. Brightness-contrast inputs are usually implemented as a linear transfer function in the form:

$$f(p) = p + k(p + c) \quad (\text{Equ. 13.16})$$

where  $k$  is the value of the contrast and  $c$  is the brightness. When these inputs are made by a slider bar, the initial value for  $k$  is 1 and the initial value for  $c$  is 0; but the numbers on the slider bar usually range from 0 to 100 and have no clear relation to the values of  $k$  and  $c$  that the slider generates.

**Other transfer functions.** Direct specification of transfer functions such as square, square root, exponential, logarithm, sine, cosine, and so forth are sometimes found in image processing software written for chemistry or physics applications, where input pixel values can be converted into meaningful physical quantities. However, in astronomy it is generally more efficient to specify a transfer function in terms of a particular image. These techniques are covered in the sections that follow.

## 13.4 Direct Endpoint Specification

Histograms of astronomical images often show that very few pixels have values outside a well-defined range. At the low end, the cutoff may be the brightness of the night sky. If the average sky pixel has a value of 120, for example, there may be virtually no pixels with values below 115. At the high end, if there is no bright celestial object in the field of view, then all pixels with high values are found in star images, with just a handful having values above 600.

In direct endpoint specification, the user examines a histogram and determines the low and high pixel values—the *endpoints*—from it. Values less than the low endpoint are saturated black, and those greater than the high endpoint are saturated white. Although direct endpoint specification gives user flexibility, it tends to be difficult to predict the optimum low and high values, especially if the only way to obtain them is by inspecting a graphic display of the histogram.

## Chapter 13: Point Operations

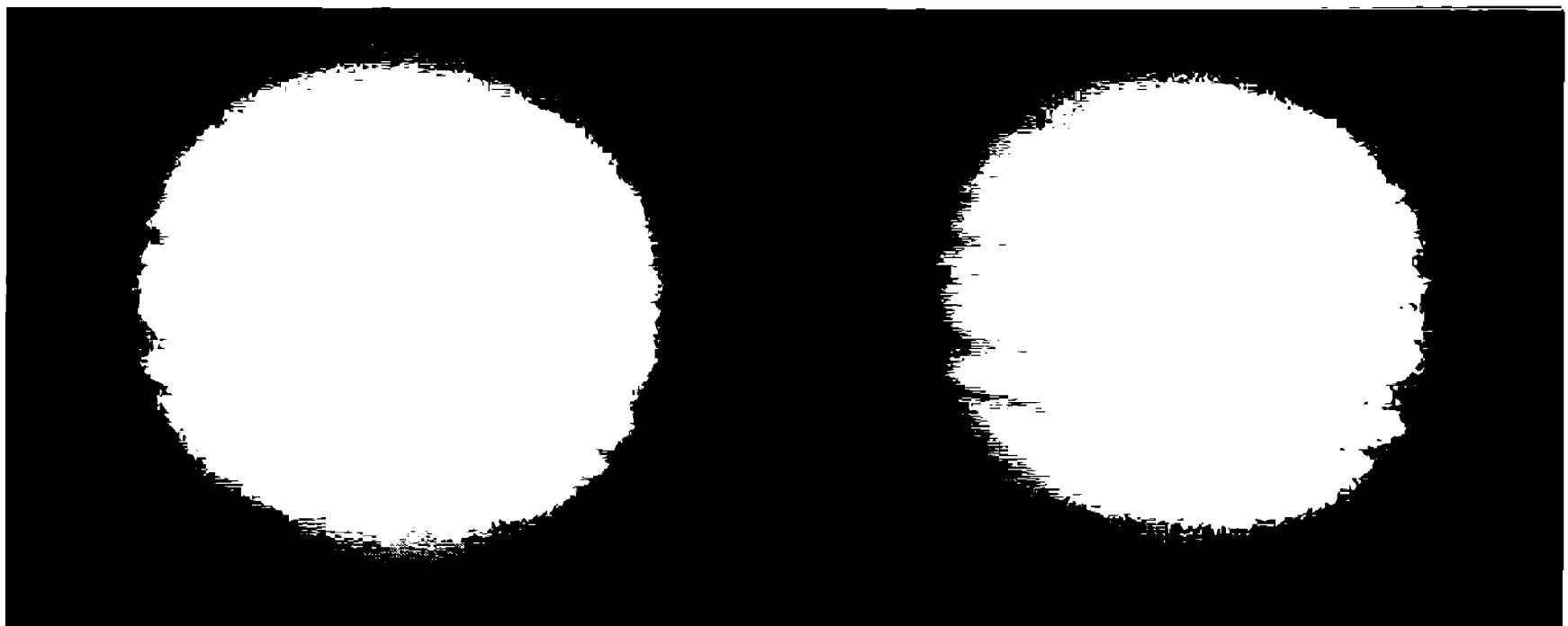


Figure 13.13 Histogram endpoint specification gives you precise control over the number of saturated pixels. The endpoint for the left image is 0.98, resulting in 2% white-saturated pixels and severe loss of detail. In the image on the right, the endpoint is 0.9999, resulting in only 0.01% saturated pixels.

### 13.4.1 Stretch Scaling

In stretch scaling, the user estimates the low and high endpoints, and the intermediate pixel values are “stretched” uniformly between these minimum and maximum values. The transfer function is:

$$f(p) = \begin{cases} p \leq p_{lo} \rightarrow f(p) = p_{min} \\ p_{lo} < p < p_{hi} \rightarrow f(p) = \left( (p - p_{lo}) / \left( \frac{p_{hi} - p_{lo}}{p_{max} - p_{min}} \right) \right) + p_{min} \\ p \geq p_{hi} \rightarrow f(p) = p_{max} \end{cases} . \quad (\text{Equ. 13.17})$$

Or, as a computer algorithm:

```
slope = (pvhi - pvlo) / (pvmax - pvmin)
for pv = pvmin to pvmax
    select case pv
        case (pv <= pvlo)
            LUT(i) = pvmin
        case (pv >= pvhi)
            LUT(i) = pvmax
        case else
            LUT(i) = ((pv - pvlo) / slope) + pvmin
    end select
next pv
```

After the array LUT () has been evaluated, it is used as a look-up table for changing the pixel values in the image.

Stretch scaling can be used to isolate a feature that has a narrow range of pixel values in an image, or simply to expand the range in an image that has a narrow range of pixel values. For example, if you have an image of a faint galaxy where

## Section 13.5: Histogram Endpoint Specification

the average sky pixel value is 120, and the highest values in the galaxy's nuclear region are 250, then it would make sense to stretch the image using a pixel value of 110 as the low and 250 as the high endpoint. Of course, the pixels in bright star images will be saturated white, and a few in the background sky might be saturated black; but those that comprise the image of the galaxy will all be middle shades of gray.

The low and high pixel values in stretch scaling are sometimes called the black point and white point because after stretching, any value less than or equal to the former value appears black, and any pixel greater than or equal to the high value appears white.

A variation on specifying the low and high value is to specify a background and range. In this nomenclature, the low pixel value is called the background, and high endpoint minus the low endpoint is the range. Aside from different terminology, background-range scaling is exactly the same as stretching between black and white.

### 13.4.2 Nonlinear Stretch Scaling

In nonlinear stretch scaling, the user estimates the low and high endpoints, and the intermediate pixel values are spread between the minimum and maximum pixel values using the gamma, logarithm, gammalog, or other nonlinear function.

As a computer algorithm, gamma scaling is carried out as follows:

```
power = 1 / gamma
for pv = pvmin to pvmax
    select case pv
        case (pv <= pvlo)
            LUT(i) = pvmin
        case (pv >= pvhi)
            LUT(i) = pvmax
        case else
            fr01 = (pv - pvlo) / (pvhi - pvlo)
            LUT(i) = pvmax * (fr01 ^ power) + pvlo
    end select
next pv
```

Once LUT() has been evaluated, it serves as the look-up table for changing the pixel values in the image.

In the algorithm, the fraction  $(pv - pvlo) / (pvhi - pvlo)$  ranges from 0 to 1 as pv runs from pvlo to pvhi. The properties of numbers between 0 to 1 make it simple to scale them nonlinearly, to lighten, darken, and otherwise manipulate pixel values between pvlo and pvhi.

## 13.5 Histogram Endpoint Specification

In histogram endpoint scaling, the user selects the fraction of pixels that can sat-

## Chapter 13: Point Operations

rate black and the fraction that can saturate white, as well as parameters describing the scaling between the endpoints. The software evaluates the histogram, finds pixel values corresponding to the low and high endpoints, computes a transfer function, and alters the image.

Other than the method of finding the endpoints, linear and nonlinear stretches with histogram endpoint specification work exactly the same as direct endpoint specification. Although specifying them differs, computing a result does not.

From the user's point of view, the principle advantage of *histogram* endpoint specification over *direct* endpoint specification is that it is result-oriented. Because you specify the result you want, results are predictable. In addition, the size of the image and the number of gray levels in the image do not influence the output. The same fraction of pixels is saturated black or white, regardless of the image itself. For each type of image—deep-sky, lunar, or planetary—well-chosen histogram endpoint specifications produce dependable and repeatable results.

### 13.6 Histogram Specification

Histogram specification takes the process of specifying the desired characteristics for the new image a step further than histogram endpoint specification: the user specifies the histogram of the new image. A scaling algorithm then computes the transfer function necessary to reshape the histogram of the original image into the desired histogram for the new image—regardless of the shape of the original one. The user does not need to know *anything* about the old histogram; it is necessary only to specify the shape of the new one.

Histogram specification is effective with deep-sky images because all good-looking deep-sky images have histograms that are basically the same: they start at zero, rise very rapidly to a peak representing the sky background, and then descend exponentially to zero. Of course, personal taste in the appearance of the images does vary, so it is useful to be able to select from a variety of target histogram shapes.

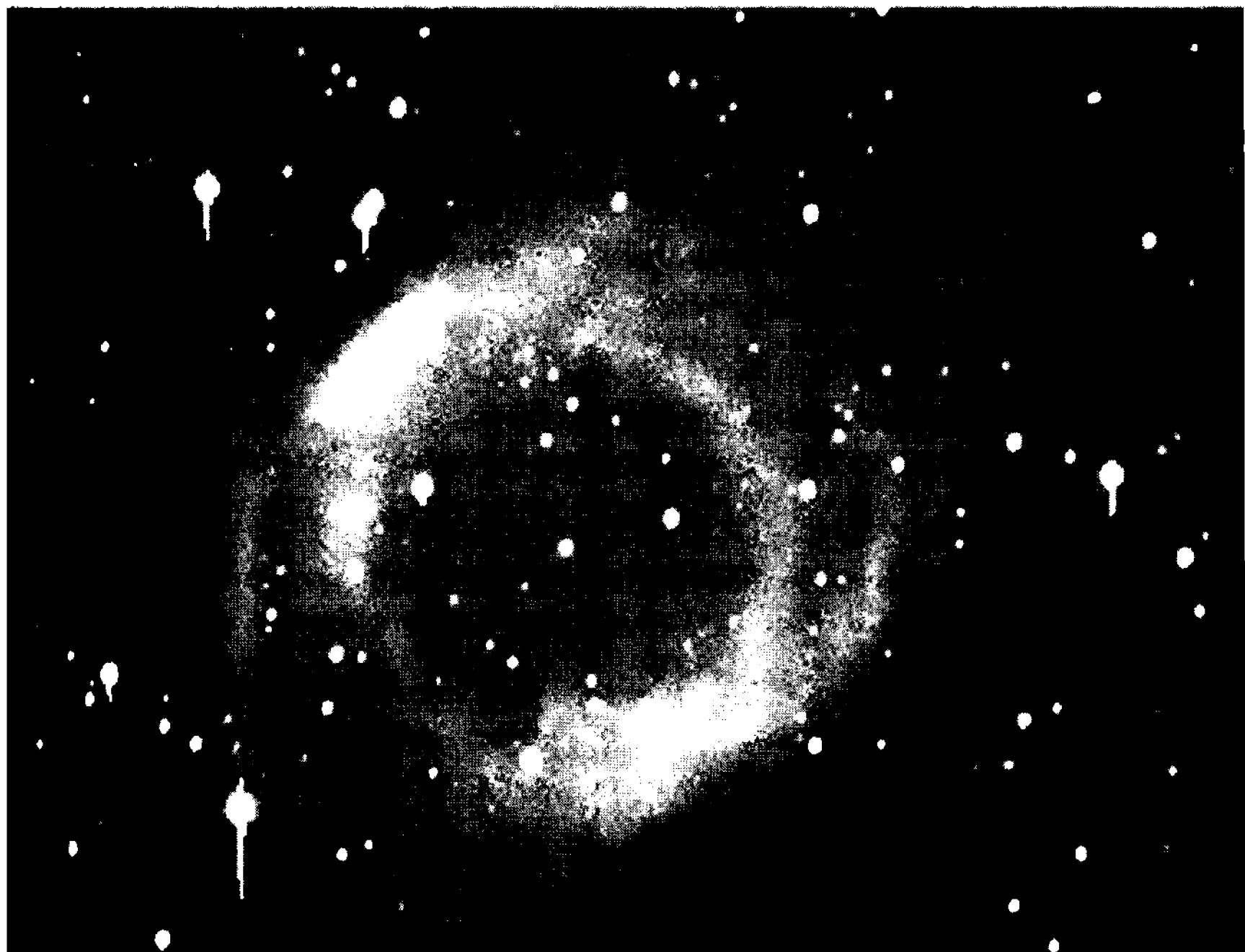
The first step in reshaping a histogram is to compute a histogram for the original image:

```
pixels = 0
FOR y = 0 to ymax
    FOR x = 0 to xmax
        pv = old(x, y)
        histo(pv) = histo(pv) + 1
        pixels = pixels + 1
    NEXT x
NEXT y
```

In this algorithm, array `histo()` contains the histogram, and the variable `pixels` counts the total number of pixels in the original image.

We next generate a cumulative histogram; that is, at each pixel value, we

## Section 13.6: Histogram Specification



**Figure 13.14** Forcing the histogram of the Helix into a Gaussian shape produced an image with a pleasingly dark sky, clean rendering of middle tones, and little saturation in bright regions. The key to good results is experimenting with the parameters that control the histogram shape.

compute the total number of pixels having a value less than or equal to that given value. The cumulative sum of pixels, `histosum`, begins at zero and rises until it equals the total number of pixels in the image. As the computation proceeds, we overwrite the original histogram with the cumulative histogram:

```
histosum = 0  
  
FOR pv = 0 TO pvmax  
    histosum = histosum + histo(pv)  
    histo(pv) = histosum  
NEXT pv
```

When this is done, we have a cumulative histogram for the original image in the array `histo()` and the total number of pixels in the variable `pixels`.

In the next three steps, we're going to compute a "target" histogram having the same total number of pixels as the original histogram and the desired distribution of pixel values. This method is not computationally efficient, but it is easy to follow, and it lends itself to virtually any target histogram distribution.

First, we generate a target histogram:

```
FOR pv = 0 TO pvmax  
    target(pv) = CALL gaussian(pv / pvmax)
```

## Chapter 13: Point Operations

```
NEXT pv
```

In this example, we have called the function Gaussian, but we could just as easily have called it any other function—a sine function, an exponential—whatever we want. Note that the value of the argument,  $pv / pvmax$ , runs from 0 to 1. The Gaussian distribution looks like this:

$$h(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\bar{x}}{\sigma}\right)^2} \quad (\text{Eq. 13.18})$$

where  $e$  is the base of natural logarithms,  $\sigma$  is the standard deviation and  $\bar{x}$  is the mean value of  $x$ . (To avoid confusion with a transfer function, we have called the histogram function  $h(x)$ .) We can discard the scaling constant in front of the exponential and generate the function as follows:

```
FUNCTION gaussian(x)
    sig = 1/6
    x0 = 0.5
    gaussian = EXP( (- (x - x0) / sig) ^ 2) / 2
END FUNCTION gaussian
```

where  $sig$  is the width of the Gaussian curve and  $x0$  is its midpoint.

In this example, a plot of the `target()` array would be the Gaussian bell curve running from 0 to 1 with its peak at 0.5. Note, however, that we can compute *any* function in the range 0 to 1 and use it as the target histogram.

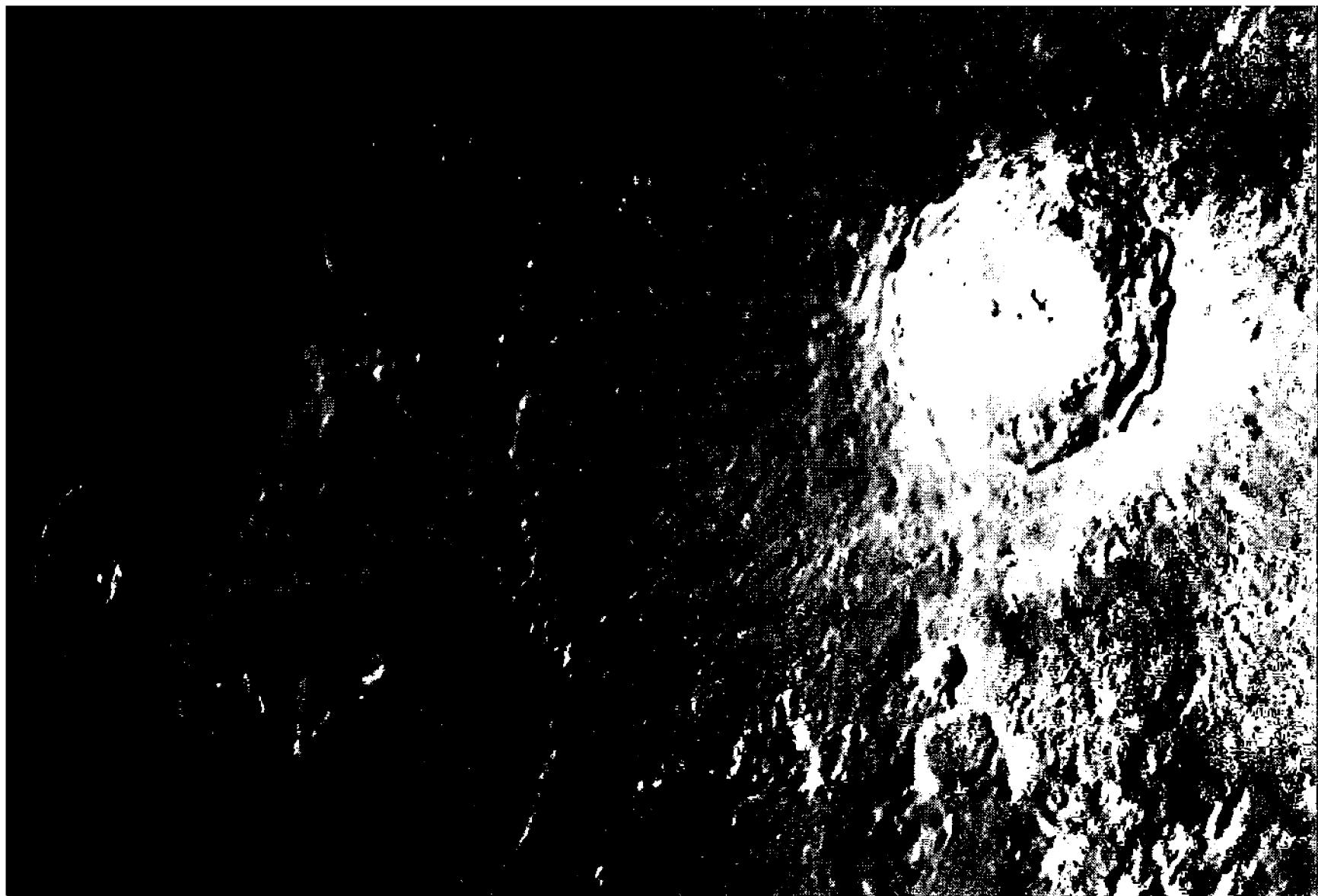
The next step computes the cumulative histogram of the function, just as we computed that of the image:

```
targetsum = 0
FOR pv = 0 TO pvmax
    targetsum = targetsum + target(pv)
    target(pv) = targetsum
NEXT pv
```

We now have cumulative histograms for both image and target. The difference is that sum totals are different. To make them the same, we simply multiply each value in the target array by the ratio of their sums:

```
ratio = histosum / targetsum
FOR pv = 0 TO pvmax
    target(pv) = ratio * target(pv)
NEXT pv
```

After scaling, the two cumulative histograms are identical except that the distribution of pixel values is not the same. Both histograms begin at the same place—zero—and end at the same place—the total number of pixels in the image—but the two curves rise at different rates. The final step is to generate a look-up table that will force the image histogram to track the target histogram in lock-



**Figure 13.15** The Gaussian histogram ( $\sigma = 4$ ) gives this image of Copernicus and Eratosthenes a pleasing distribution of gray values above and below an average middle gray tone. Histogram shaping can be very effective with lunar images. Image by Thierry Legault.

step. The following algorithm does just that:

```

pvnew = 0
FOR pv = 0 TO pvmax
    DO UNTIL target(pvnew) => histo(pv)
        pvnew = pvnew + 1
    LOOP
    LUT(pv) = pvnew
NEXT pv

```

What happens in the algorithm is not especially obvious. It is this: the pixel values start at zero and move higher. At each pixel value, the algorithm compares the cumulative pixel count of the image with that of the target histogram. The action then branches, depending on whether the cumulative count of image pixels is “behind” or “ahead” of the pixel count in the target.

If the target is behind, program execution skips the loop. This means that in the look-up table,  $pv$  increases by one, but  $pvnew$  stays the same. When the pixels in the image are later run through the look-up table, pixels with the value of  $pv$  will become  $pvnew$ , increasing the pixel count in the new image.

However, if the target is ahead of the image, the loop executes. The value of  $pvnew$ , the value to be written into the look-up table, increases until the image count catches up to the target count. As a result, the pixel count in the new image

## Chapter 13: Point Operations

does not increase until the image is once again ahead of the target.

On reflection, you will see that there will be fewer different values of  $pv_{\text{new}}$  than there are of  $pv$  because  $pv_{\text{new}}$  can be assigned to multiple values of LUT( $pv$ ), and some values of  $pv_{\text{new}}$  can be skipped.

To create a new image, every pixel in the old one gets passed through the look-up table. This operation runs very rapidly:

```
FOR y = 0 to ymax
    FOR x = 0 to xmax
        new(x, y) = LUT(old(x, y))
    NEXT x
NEXT y
```

Although the algorithm above is not efficient, it is easier to understand than an efficient algorithm; and it works for any target function.

After reshaping a histogram, it is tempting to examine the histogram of the new image to make sure it has the desired shape. Often, the new histogram looks rather odd, consisting of high peaks separated by gaps. This happens because the interesting features in the original image have a limited number of gray levels. When the algorithm computes the look-up table, it must skip over target pixel values until the target histogram catches up with the image histogram. Limited by the original data, it is forced to generate a “gappy” histogram. Even though it may look messy, the new one has the same statistical properties as the target histogram.

### 13.6.1 Histogram Equalization

Histogram equalization, found in many different types of image processing software, is simply a special case of histogram specification in which the target histogram is a straight line. The flat-line histogram means that in the new image, every pixel value should be present in equal numbers. The equation for the histogram shape is:

$$h(x) = c. \quad (\text{Equ. 13.19})$$

where  $c$  is a constant.

Histogram equalization is too brutal for astronomical work. In deep-sky images, far too many pixels wind up with values in the middle and high end of the gray scale, so the sky looks bright and washed out. In planetary images, the large number of pixels in the sky become low to middle shades of gray, and the relatively small numbers of pixels with high values end up as very light tones.

### 13.6.2 Gaussian Histogram Shaping

Nature loves the Gaussian distribution, so it seems reasonable to specify one in astronomical images. However, the Gaussian distribution implicitly assumes variation about a central mean, whereas most astronomical images are dominated by sky pixels with an “exponential” decline in their numbers that are brighter than the night sky. Be that as it may, shifting the peak of the Gaussian distribution toward

## Section 13.6: Histogram Specification

a low mean value produces a good-looking result with deep-sky images.

- **Tip:** *AIP4Win allows you to change the width of the Gaussian distribution, and also to shift the peak toward lower or higher pixel values using the Histogram Specification Tool. This tool previews settings so that you easily find the optimum values.*

### 13.6.3 Exponential Histogram Shaping

Although deep-sky images often have histograms that are exponential in appearance, a true exponential histogram often looks too dark. This occurs because the greatest number of pixels in a true exponential distribution have a value of zero. The equation of the exponential histogram is:

$$h(x) = e^{-kx} \quad (\text{Equ. 13.20})$$

where  $e$  is the base of natural logarithms. To implement this function, a value for the constant  $k$  must be discovered. Setting an appropriate default value of  $k$  gives good image brightness and produces deep-sky images that have a pleasantly realistic appearance.

- **Tip:** *In AIP4Win, the exponential histogram works very nicely with images of bright deep-sky objects, preserving information in the bright core while simultaneously bringing out detail in the faint outer parts.*

## Chapter 13: Point Operations

---

# 14 Linear Operators

---

Among the most useful tools in the image-processing repertoire are those that generate a new pixel value based on the relationship between the value of a given pixel and the values of those that surround it. This chapter and the next describe image processing with “neighborhood” operations—that is, processes that involve the pixel’s immediate (and not-so-immediate) neighbors.

This chapter focuses mainly on convolutions—operations in which new pixel values result from linear combinations (i.e., addition and multiplication) of a pixel with its neighbors. Chapter 15 deals with neighborhood operations in which the new values result from nonlinear operations (such as sorting) on the neighboring pixels.

## 14.1 Convolution in One Dimension

In convolution, two functions are overlaid and multiplied by one another. One of the functions is an image and the other is a convolution kernel; that is, one function is represented by a large array and the other by a relatively small array. In convolution, the kernel “operates” on the image—which is why it is called an operator.

Consider a one-dimensional section extracted from a sharp image:

$$[\dots 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 90 \ 90 \ 90 \ 90 \ 90 \ 90 \ 90 \dots].$$

These pixel values might be a slice from any abrupt change in brightness in an image, such as the edge of a building or the limb of the Moon. As shown, the image is in sharp focus, so that the large change in brightness is abrupt, occurring entirely between one pixel and the next. We could represent the distribution of light falling on the detector as an array of numbers:

$$[0.00 \ 1.00 \ 0.00].$$

This kernel says that 100% of the light from a point source such as a star falls on a spot one pixel wide.

Suppose, however, that the exposure was made on a night of poor seeing; the image would be the result of convolving the source sharp image with the blur in-

## Chapter 14: Linear Operators

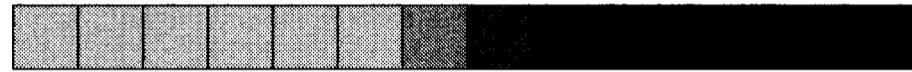
A sharp transition in the original image...



...convolved with a soft kernel...



...produces an image with soft transitions.



**Figure 14.1** In your telescope, convolution is an analog process: the sharp image outside the atmosphere and telescope optics is softened and blurred. In your computer, convolution is a digital process. The convolution kernel operates on each pixel in the original image to produce a new pixel.

duced by the turbulent atmosphere. The light might have spread so that only half of it reached the target pixel, with the other half falling on the pixels to either side. We could represent the point-source spread by air turbulence as:

$$[0.25 \text{ } \mathbf{0.50} \text{ } 0.25] .$$

This little array says that light destined for the target pixel (indicated in boldface) spreads across three pixels, with 50% going to the target and 25% to those on either side. It is obvious that the image will not be as sharp as it had been because the image has “mixed” with the atmospheric point-spread function, in a process called convolution. After convolution, the image has become:

$$[\dots \text{ } 10 \text{ } 30 \text{ } 70 \text{ } 90 \text{ } \dots] .$$

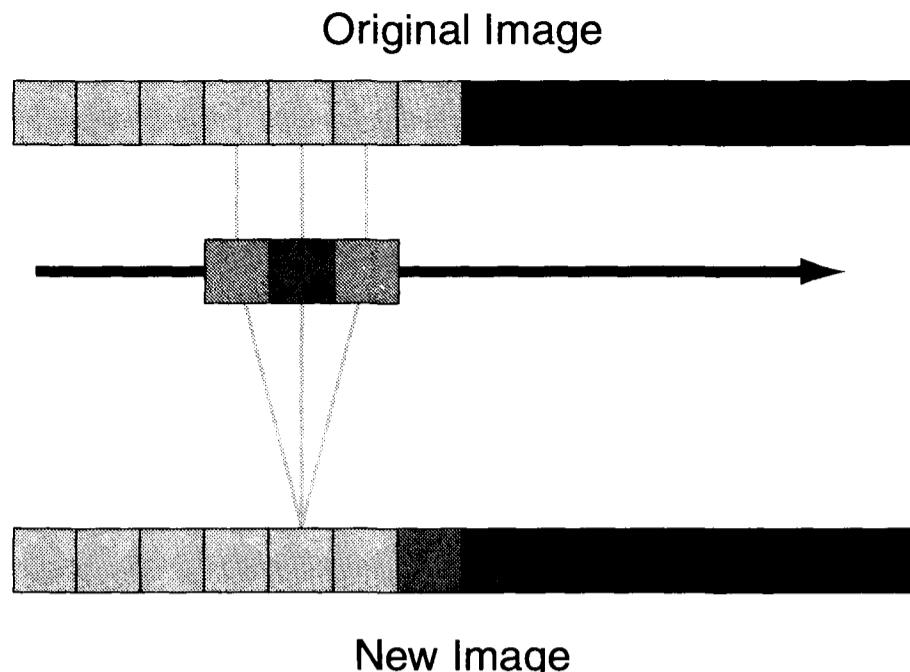
The sharply defined edge in the source image has become soft, so that instead of making an abrupt transition from low pixel values to high ones, the transition is gradual.

In a one-dimensional convolution, each pixel in the convolved image is the sum of the product of each element in the kernel times the corresponding pixel value in the source image:

$$N_x = \frac{\sum_{i=-r}^r k_i O_{x+i}}{\sum_{i=-r}^r k_i} \quad (\text{Equ. 14.1})$$

where  $N_x$  is the  $x$ -th pixel in the new image generated by convolution,  $O_x$  is the  $x$ -th pixel in the original image, and  $k_i$  is the  $i$ -th element in a convolution kernel

## Section 14.1: Convolution in One Dimension



**Figure 14.2** You can think of a one-dimensional digital convolution as an operation that moves down a row of pixels. It stops at each one, reads pixel values beside it, multiplies them by the values in the kernel, sums them, divides by the sum of the kernel, writes a pixel value to the new image, and then moves on.

that extends from  $-r$  to  $+r$ . Since the convolution is carried out for every pixel in the destination image, as  $x$  progresses, the kernel “steps” along the image generating each pixel in the destination image from elements in the source image.

The equation treats the index of the convolution kernel as if it runs from  $-r$  to  $+r$ , a convention that often proves convenient. This implies that the convolution kernel consists of a symmetrical arrangement of  $2r + 1$  elements with a zero-th element in the center. In writing out a convolution kernel, the zero-th element is shown in boldface type.

Bear in mind that the notation you see in this chapter is simply a compact way to write the elements of the kernel:

$$[0.25 \text{ } \mathbf{0.50} \text{ } 0.25] \equiv k(-1) = 0.25 \text{ } k(0) = 0.50 \text{ } k(1) = 0.25 \quad . \quad (\text{Equ. 14.2})$$

Just as real blurs are not necessarily symmetrical, a convolution kernel need not be symmetrical. Computationally, kernels that are not symmetrical can be “padded” with zeros to make them so. For example, the asymmetric kernel [1 1] can be padded [0 1 1], which has an index that runs from  $r = -1$  to  $r = +1$ . The zero elements have no effect on the outcome of the calculation. Convolution automatically normalizes the destination image: it divides the convolution product by the sum of the elements in the kernel. By so doing, numerical convolution takes into account the physical fact that light is neither lost nor created when an image is blurred.

However, in image processing, normalization is sometimes ignored. The rationale given most often is that integer computation is easier to understand and faster, that the division can be carried out at a later stage in the computation; and that if the sum of the elements is zero, the destination image is indeterminate. Do not be surprised when division by the sum of elements in the kernel is ignored, as-

## Chapter 14: Linear Operators

sumed to be implied, or left to the user to specify.

Consider the atmospheric convolution kernel, [0.25 **0.50** 0.25], used in the example above. Even though the total amount of light reaching the detector remains constant no matter how the light is redistributed by atmospheric turbulence, the kernel could be written as  $0.25 \times [1 \ 2 \ 1]$  or abbreviated to [1 2 1]. The different ways of writing the convolution kernel are equivalent in the sense that the destination images are proportional, differing only by a scaling factor equal to the sum of the elements in the convolution kernel.

In the discussions that follow, different forms of kernel are used interchangeably, as they are in the literature of image processing. In practice, the somewhat slippery conventions for representing convolution kernels present little difficulty because their interpretation is almost always clear from context.

The following algorithm is for a one-dimensional convolution with normalization by the sum of the elements in the kernel. The original image is the array `old(0 to xmax)`, the new image is `new(0 to xmax)`, and the kernel is `k(-r to r)`. After loading values into the arrays `old()` and `k()`, the convolution is computed:

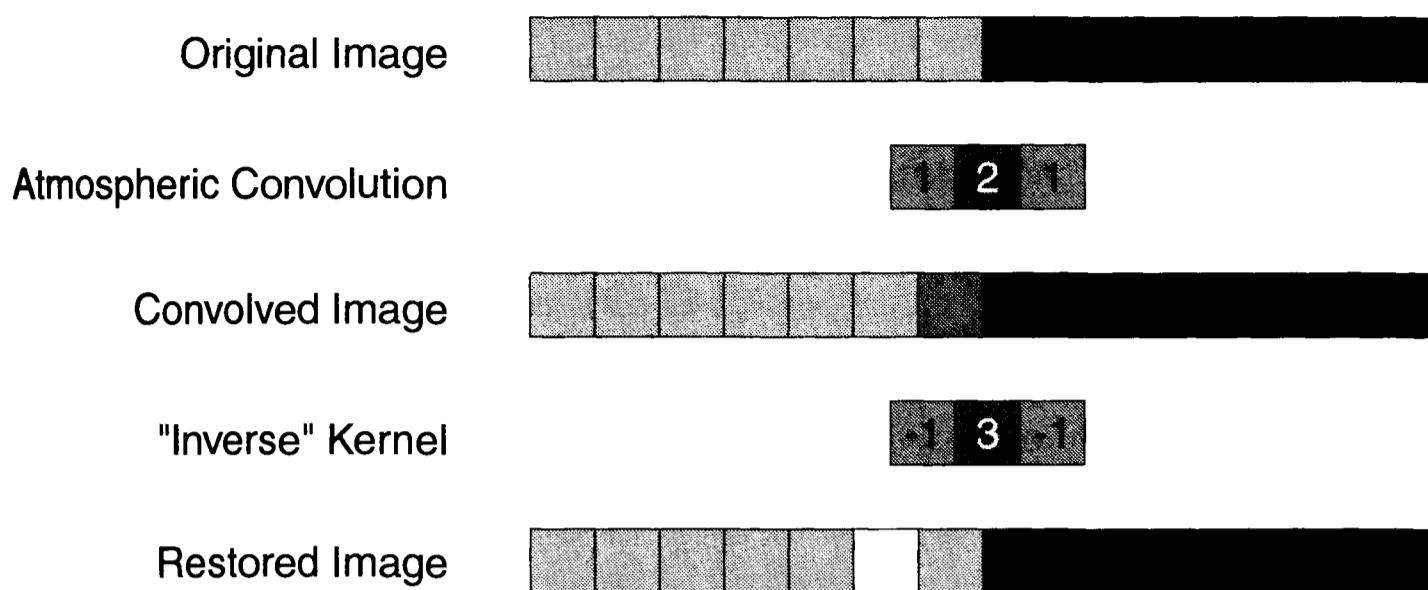
```
FOR i = -r TO r
    ksum = ksum + k(i)
NEXT i
FOR x = r TO xmax - r
    FOR i = -r to r
        new(x) = new(x) + k(i) * old(x + r)
    NEXT i
    new(x) = new(x) / ksum
NEXT x
```

The algorithm first computes the sum of the kernel; next for each pixel in the `new()` array, it computes the sum of the products and then divides by the sum of the kernel. Note that `x` can be computed only between `r` and `xmax - r` because the range between 0 and `xmax` would require values outside the bounds of `old()`. Array bounds are an annoyance when computing convolutions.

**What good is it?** Often, when performing convolutions in astronomy, the real goal is to *undo* an image that has already been convolved with the atmospheric blur function; that is, the goal is really to *deconvolve* the image—to undo a convolution. If we could deconvolve an image, we could restore to a blurred telescopic image all of the detail lost to atmospheric turbulence and less-than-perfect telescope optics.

The bad news is that deconvolution is difficult and sometimes impossible. We often do not know the convolution kernel. However, when it is precisely known, which is seldom the case, the image has been corrupted by noise; and even when the noise is very slight, there is no guarantee that an “inverse” kernel exists. The good news is that fairly crude approximations of the “ideal” inverse kernel do exist, and often do a remarkably good job of restoring blurry images. For example,

## Section 14.1: Convolution in One Dimension



**Figure 14.3** After the analog process of convolution has degraded the image reaching the CCD, can the “inverse” digital convolution restore lost detail? In this example, the inverse kernel restores a sharp edge but adds some artifacts—a light pixel to the left and a dark pixel to the right of the edge.

convolving the image softened by atmospheric blurring using an “inverse” kernel  $[-1 \ 3 \ -1]$  produces a sharpened version of the atmospherically convolved image:

$$[\dots \ 10 \ 10 \ 10 \ 10 \ -10 \ 10 \ 90 \ 110 \ 90 \ 90 \ 90 \ 90 \ \dots].$$

Although  $[-1 \ 3 \ -1]$  is not the exact inverse of  $[1 \ 2 \ 1]$ , it is better than nothing. The sharp transition marking the edge has been restored at the cost of picking up some erroneously high and low pixel values (called “ringing”) nearby. Nevertheless, to the detail-greedy eye of an astronomer looking for lost features in an image, this somewhat less-than-perfectly restored picture represents a significant gain over an atmospherically degraded one.

### 14.1.1 One-Dimensional Examples

This section illustrates the behavior of one-dimensional convolution kernels, beginning with some that are so simple as to seem trivial. Nevertheless, it is important to understand very simple cases before attempting to work with complex two-dimensional convolution kernels.

In the first set of examples, the one-dimensional test image consists of the short section of an image. You can think of it as a star image:

$$[\dots \ 0 \ 0 \ 0 \ 0 \ 100 \ 0 \ 0 \ 0 \ 0 \ \dots].$$

The simplest convolution kernel is the unity operator,  $[1]$ . This kernel is one element wide by one element high, and the single element in the kernel is 1 (unity). When the source image is convolved with the unity kernel, the destination image is identical to the source image:

## Chapter 14: Linear Operators

$$[\dots \ 0 \ 0 \ 0 \ 0 \ 100 \ 0 \ 0 \ 0 \ 0 \ \dots].$$

The formal definition of convolution specifies that when an image is convolved with the kernel, the values in the image are multiplied by the value of the single element in the kernel and divided by their sum. When the kernel consists of negative or floating-point values, as in  $[-3.14159]$ , and division by the sum of the kernel is performed during convolution, an intermediate destination is:

$$\frac{1}{-3.14159} \times [0 \ 0 \ 0 \ 0 \ -314.59 \ 0 \ 0 \ 0 \ 0],$$

which reduces to:

$$[\dots \ 0 \ 0 \ 0 \ 0 \ 100 \ 0 \ 0 \ 0 \ 0 \ \dots].$$

However, if division is left to the discretion of the user to carry out, the pixel values in the destination image can be scaled in any way desired.

The displacement kernel is nearly as simple as the unity kernel. In it, the unity element is displaced from the zero position:  $[0 \ 0 \ 1]$ . When convolved with the test image, the pixels in the destination image are shifted in the direction opposite the displacement of the unity element:

$$[\dots \ 0 \ 0 \ 0 \ 100 \ 0 \ 0 \ 0 \ 0 \ 0 \ \dots].$$

Convolution with the delta function (an image with a single non-zero pixel) produces a mirror-image of the convolution kernel.

Convolving the test image with the kernel  $[1 \ 2 \ 3 \ 4]$  produces:

$$[\dots \ 0 \ 400 \ 300 \ 200 \ 100 \ 0 \ 0 \ 0 \ 0 \ \dots].$$

In the above example, the zero-th element in the kernel was the first, so the index of the elements ranged from  $r = 0$  to  $r = 3$ . The location of the zero-th elements determines the positions of the pixel values in the destination. For example, changing the kernel from  $[1 \ 2 \ 3 \ 4]$  to  $[1 \ 2 \ 3 \ 4]$  produces:

$$[\dots \ 0 \ 0 \ 0 \ 0 \ 400 \ 300 \ 200 \ 100 \ 0 \ \dots].$$

Finally, if the elements in the kernel sum to zero, as they do in  $[-1 \ 2 \ -1]$ , the destination cannot be normalized because it would require division by zero:

$$[\dots \ 0 \ 0 \ 0 \ -100 \ 200 \ -100 \ 0 \ 0 \ 0 \ \dots].$$

If this result is supposed to be a real image, we immediately know that it cannot be correct. Negative values cannot exist in a real image, because negative light does not exist. Depending on the result desired, we can simply ignore negative values by setting them to zero:

## Section 14.1: Convolution in One Dimension

$$\left[ \dots \quad 0 \quad 0 \quad 0 \quad 0 \quad 200 \quad 0 \quad 0 \quad 0 \quad 0 \quad \dots \right],$$

or, by adding a constant, we can force the negative values to be positive:

$$\left[ \dots \quad 100 \quad 100 \quad 100 \quad 0 \quad 300 \quad 0 \quad 100 \quad 100 \quad 100 \quad \dots \right].$$

Finally, the output may be left as is—with negative values—and passed on to the next step in a larger computation.

### 14.1.2 Multiple Convolutions with One-Dimensional Kernels

By convolving an image repeatedly with a small kernel, it is possible to create the same result produced by a larger kernel. For example, observe what happens to the destination images during four successive passes with the two-element convolution kernel [1 1]:

$$\text{Pass 1: } \frac{1}{2} \times \left[ \dots \quad 0 \quad 0 \quad 0 \quad 100 \quad 100 \quad 0 \quad 0 \quad 0 \quad 0 \quad \dots \right],$$

$$\text{Pass 2: } \frac{1}{4} \times \left[ \dots \quad 0 \quad 0 \quad 100 \quad 200 \quad 100 \quad 0 \quad 0 \quad 0 \quad 0 \quad \dots \right],$$

$$\text{Pass 3: } \frac{1}{8} \times \left[ \dots \quad 0 \quad 100 \quad 300 \quad 300 \quad 100 \quad 0 \quad 0 \quad 0 \quad 0 \quad \dots \right],$$

$$\text{Pass 4: } \frac{1}{16} \times \left[ \dots \quad 100 \quad 400 \quad 600 \quad 400 \quad 100 \quad 0 \quad 0 \quad 0 \quad 0 \quad \dots \right].$$

The last image results in exactly the same destination as would convolving the source image with the five-element kernel [1 4 6 4 1].

In fact, the product of two kernels is computed in exactly the same way that the product of a kernel and an image are computed. To find the array  $k_3()$ , the convolution of  $k_1()$  and  $k_2()$ , apply the following algorithm:

```

FOR i = -r1 TO r1
    FOR j = -r2 TO r2
        k3(i + j) = k3(i + j) + k1(i) * k2(j)
    NEXT j
NEXT i

```

Note that given the array  $k_1(-r1 \text{ to } r1)$  and the array  $k_2(-r2 \text{ to } r2)$ , the resulting array,  $k_3$ , has dimensions  $(-(r1 + r2) \text{ to } (r1 + r2))$ . In this way, large hard-to-compute kernels can be built up from several applications of a small easy-to-compute kernel.

Convolution is shown by the circled multiplication sign:

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix},$$

## Chapter 14: Linear Operators

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 3 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 3 & 3 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}.$$

The resulting kernel is the extremely useful binomial filter, which is an excellent approximation of the Gaussian blur so typical of well-sampled star images. Its two-dimensional counterpart—used as a filter for smoothing noisy images—can be built up from very simple kernels.

## 14.2 Convolution in Two Dimensions

When starlight falls on a detector, it is distributed two-dimensionally on the array of pixels—the star image spread by the telescope optics and atmospheric turbulence. In reality, the distribution of starlight is a continuous function, of course, called the *point-spread function*. However, on the spatially quantized grid of pixels on the CCD, the point-spread function probably looks something like this:

$$\begin{bmatrix} 0.06 & 0.13 & 0.06 \\ 0.13 & \mathbf{0.24} & 0.13 \\ 0.06 & 0.13 & 0.06 \end{bmatrix}.$$

Even before the image arrives at the detector, it has already undergone analog convolution with the optical system and atmosphere. Consider a small section from a sharp image of an abrupt edge such as the limb of the Moon convolved with the telescope/atmosphere point-spread function:

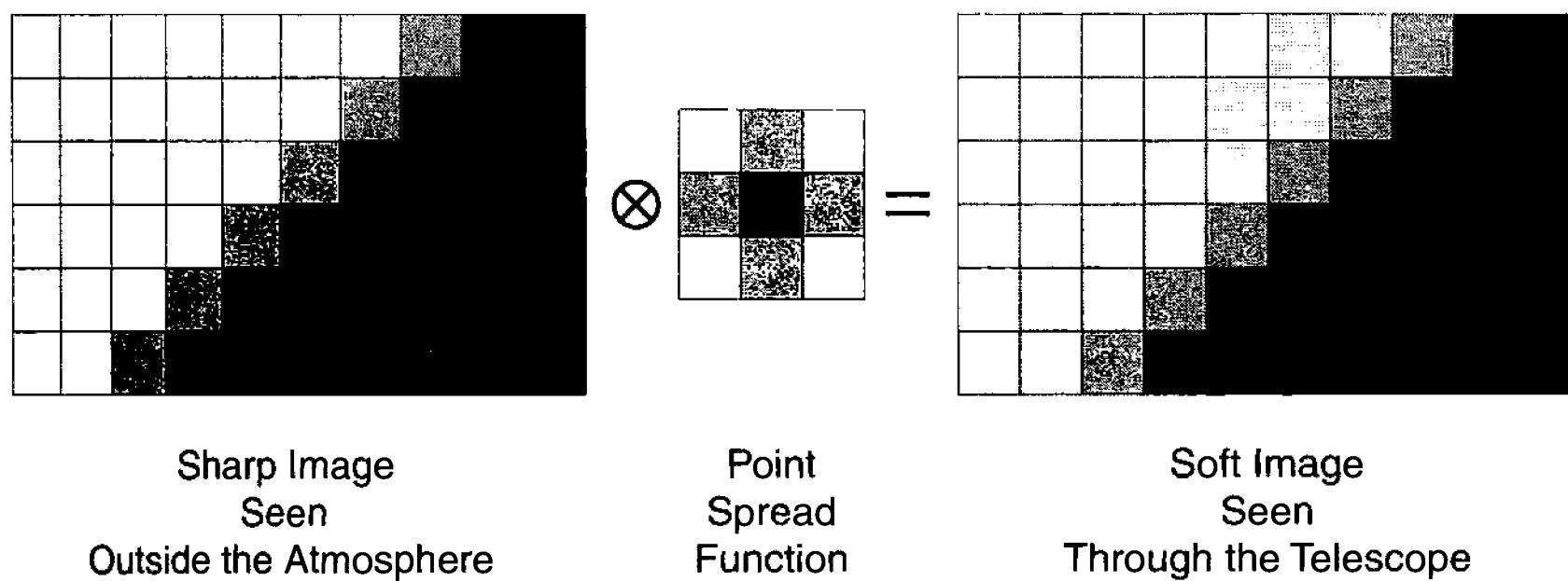
$$\begin{bmatrix} 10 & 10 & 10 & 10 & 10 & 10 & 10 & 50 & 90 & 90 \\ 10 & 10 & 10 & 10 & 10 & 10 & 50 & 90 & 90 & 90 \\ 10 & 10 & 10 & 10 & 10 & 50 & 90 & 90 & 90 & 90 \\ 10 & 10 & 10 & 10 & 50 & 90 & 90 & 90 & 90 & 90 \\ 10 & 10 & 10 & 50 & 90 & 90 & 90 & 90 & 90 & 90 \\ 10 & 10 & 10 & 50 & 90 & 90 & 90 & 90 & 90 & 90 \end{bmatrix} \otimes \begin{bmatrix} 0.06 & 0.13 & 0.06 \\ 0.13 & \mathbf{0.24} & 0.13 \\ 0.06 & 0.13 & 0.06 \end{bmatrix}$$

After convolution, the edge has become broad and fuzzy:

$$\begin{bmatrix} 10 & 10 & 10 & 10 & 10 & 12 & 25 & 50 & 75 & 88 \\ 10 & 10 & 10 & 10 & 12 & 25 & 50 & 75 & 88 & 90 \\ 10 & 10 & 10 & 12 & 25 & 50 & 75 & 88 & 90 & 90 \\ 10 & 10 & 12 & 25 & 50 & 75 & 88 & 90 & 90 & 90 \\ 10 & 12 & 25 & 50 & 75 & 88 & 90 & 90 & 90 & 90 \\ 12 & 25 & 50 & 75 & 88 & 90 & 90 & 90 & 90 & 90 \end{bmatrix}.$$

Convolution is one of the most powerful tools available in astronomical im-

## Section 14.2: Convolution in Two Dimensions



**Figure 14.4** In two dimensions, the image and the convolution kernel have a more familiar look. The sharp image is just an area of the sky, and the kernel representing the point-spread function is typical of a well-focused star image. The resulting soft edge looks like the limb of the Moon in a typical CCD image.

age processing. As we shall see below, this versatile neighborhood operation can smooth or sharpen images, detect edges, and produce a variety of effects that aid us in extracting all the information present in the image.

### 14.2.1 Convolution using Kernels

Convolution in two dimensions is defined in much the same way as it is in one dimension; i.e., as the sum of the product of the kernel with the pixel in the corresponding location in the image:

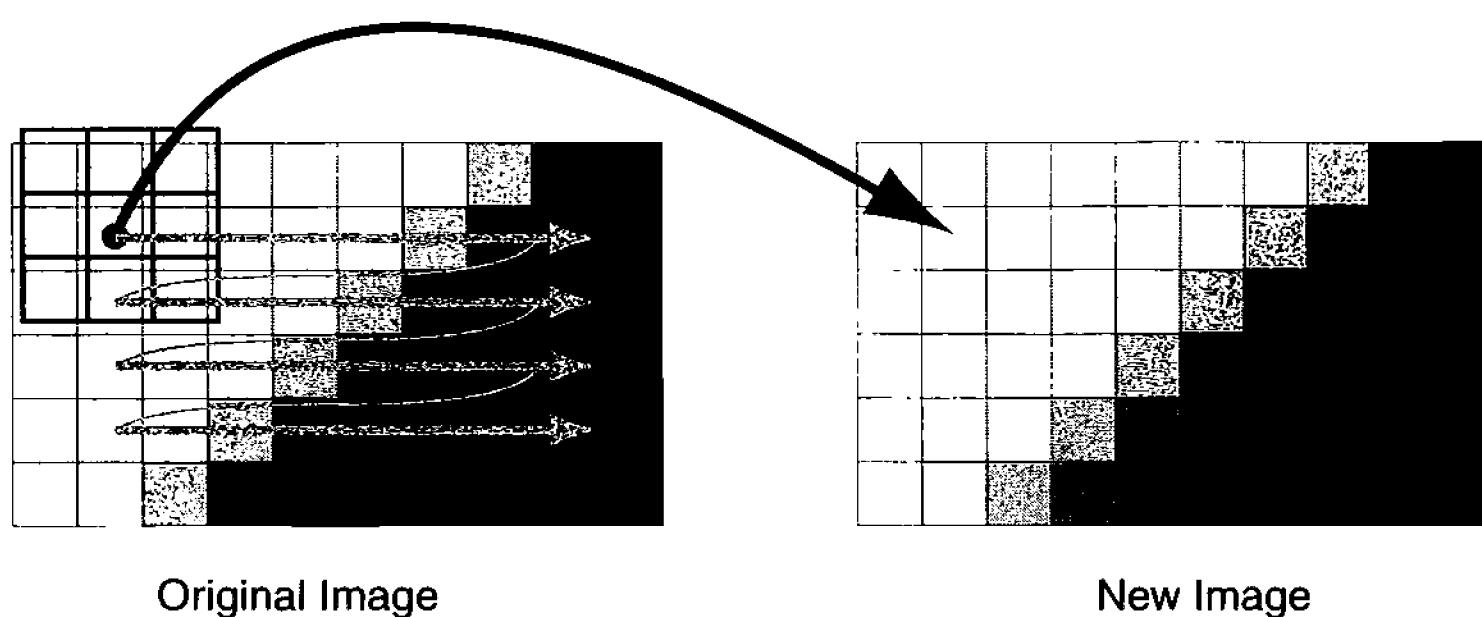
$$N_{x,y} = \frac{\sum_{i=-r}^r \sum_{j=-r}^r k_{i,j} O_{x+i, y+j}}{\sum_{i=-r}^r \sum_{j=-r}^r k_{i,j}}. \quad (\text{Equ. 14.3})$$

This operation may be a bit difficult to visualize. Imagine the convolution kernel superimposed on the image array, one pixel at a time for the entire image, with the zero-zero element of the kernel directly over the current pixel. Each element in the kernel is multiplied by the pixel value directly under it; the products summed and then divided by the sum of the elements in the kernel. This operation is repeated for *each* pixel in the entire image.

Convolution kernels come in all sizes. In the following sections, you will see examples of  $3 \times 3$  kernels ( $r = 1$ ),  $5 \times 5$  kernels ( $r = 2$ ), and  $7 \times 7$  kernels ( $r = 3$ ). Using functions to generate separable kernels, however, it is feasible to create and apply kernels of much greater size. Furthermore, they need not have equal dimensions, so kernels of  $3 \times 5$  elements or  $1 \times 9$  elements are perfectly legitimate.

Finally, the kernel need not be filled: zeros can hold as many locations in it as the user wishes. From the definition of convolution, it is clear that the zero-value elements have no effect on the outcome. Kernels with even dimensions (i.e., 2

## Chapter 14: Linear Operators



**Figure 14.5** To compute a two-dimensional digital convolution, the kernel sweeps right to left and top to bottom, visiting in turn each pixel in the original image. Each pixel in the original image and neighboring pixels are multiplied by the corresponding elements from the kernel, summed, and normalized.

$4 \times 4$ ) can be treated as ones with odd dimensions ( $3 \times 5$ ) having zero-padded rows or columns.

Convolution draws heavily on computing resources. To convolve an image  $512 \times 512$  pixels with a convolution kernel that is  $7 \times 7$  elements requires  $7 \times 7 \times 512 \times 512 = 12$  million multiplications and 12 million additions—with a hundred thousand divisions thrown in as small change. In the early days of image processing, it took a mainframe to carry out such major operations. Microcomputers now handle the job with dispatch.

The following pseudocode shows an algorithm for carrying out a two-dimensional convolution and normalization by the sum of the elements in the kernel. The first step is to compute the sum of the kernel:

```

FOR j = -r TO r
    FOR i = -r TO r
        ksum = ksum + k(i,j)
    NEXT i
NEXT j

```

The convolution routine that follows is computationally intensive because multiplication and addition steps are carried out inside four nested loops:

```

FOR y = r TO ymax - r
    FOR x = r TO xmax - r
        FOR j = -r TO r
            FOR i = -r TO r
                new(x,y) = new(x,y)
                    + k(i,j) * old(x+i,y+j)
            NEXT i
        NEXT j
        IF ksum<>0 THEN new(x,y) = new(x,y) / ksum
    
```

## Section 14.2: Convolution in Two Dimensions

```
NEXT x  
NEXT y
```

If the sum of elements in the kernel is zero, normalization is impossible.

The resulting image may have pixels with negative values. To display the result as a gray-value image, the user can add a constant to all of the pixel values in it. Many image-processing packages allow the user to specify a value to take the place of ksum or a constant to add to the resulting pixel value.

The algorithm above does not evaluate the edge pixels for the destination image because there are no pixels in the old image for the corresponding elements in the kernel. To compute values for edge pixels, the kernel can be truncated so as not to extend past the edge of the old image, a sum for the overlapping section computed, and a result assigned to the edge pixel in the destination image.

### 14.2.2 Properties of the Convolution Kernel

Because it is a fundamentally simple scheme of multiplications and additions, convolution shares many of the properties of those operations. These properties lead to extremely versatile ways to carry out convolution with images. In particular, convolution is a linear shift-invariant (LSI) process. It is these two properties that distinguish convolution kernels discussed in this chapter from the nonlinear neighborhood operators discussed in the next one.

**Linearity.** This property means that convolution kernels and the images they operate on can be separated, computed, and then added to produce the same new image as that derived from computing them together. At the most basic level, suppose that you separated an image into a large number of images, each containing just one pixel, and computed the convolution for each one-pixel image. After summing the images into one, you will have the same new image you would have obtained if you had computed the convolution of the complete image. Although this seems a bit esoteric, convolution algorithms rely on this property because they compute the image one pixel at a time and sum the destination into the final image.

The linearity property also means that if you convolve an image with one kernel and then convolve it with another, the destination is the same as convolving the image with a kernel whose elements are the sums of the corresponding elements of the other two kernels:

$$\mathbf{G} \otimes \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} + \mathbf{G} \otimes \begin{bmatrix} k & l & m \\ n & o & p \\ q & r & s \end{bmatrix} = \mathbf{G} \otimes \begin{bmatrix} (a+k) & (b+l) & (c+m) \\ (d+n) & (e+o) & (f+p) \\ (g+q) & (h+r) & (i+s) \end{bmatrix}. \quad (\text{Equ. 14.4})$$

In the equation above,  $\mathbf{G}$  represents the image. In the sections that follow, we will apply this property again and again.

**Shift Invariance.** The result of convolution with a kernel is the same regardless of where in the image a particular set of pixels is located. Thus, convolution with the image of a planet will produce the same output whether the planet is lo-

## Chapter 14: Linear Operators

cated in the center of an image or near the edge.

**Commutativity.** This property means that the destination of a sequence of convolutions does not depend on the order in which you apply them. Thus:

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}. \quad (\text{Equ. 14.5})$$

If nothing else, this means you don't have to worry when you apply several passes with an unsharp mask to a breathtakingly detailed planetary image: the image you produce does not depend on which convolution you apply first.

**Associativity.** This property means that you can build up large kernels from small ones, or you can apply small kernels in succession to produce the same destination as applying a large one. Kernels are convolved with each other in much the same way that a kernel is convolved with an image. The following pseudocode creates a new kernel,  $k_3()$ , from two small one,  $k_1()$  and  $k_2()$ :

```

FOR y = -ry TO ry
    FOR x = -rx TO rx
        FOR j = -rj TO rj
            FOR i = -ri TO ri
                k3(x+i,y+j) = k3(x+i,y+j)
                    + k1(i,j) * k2(x,y)
            NEXT i
        NEXT j
    NEXT x
NEXT y

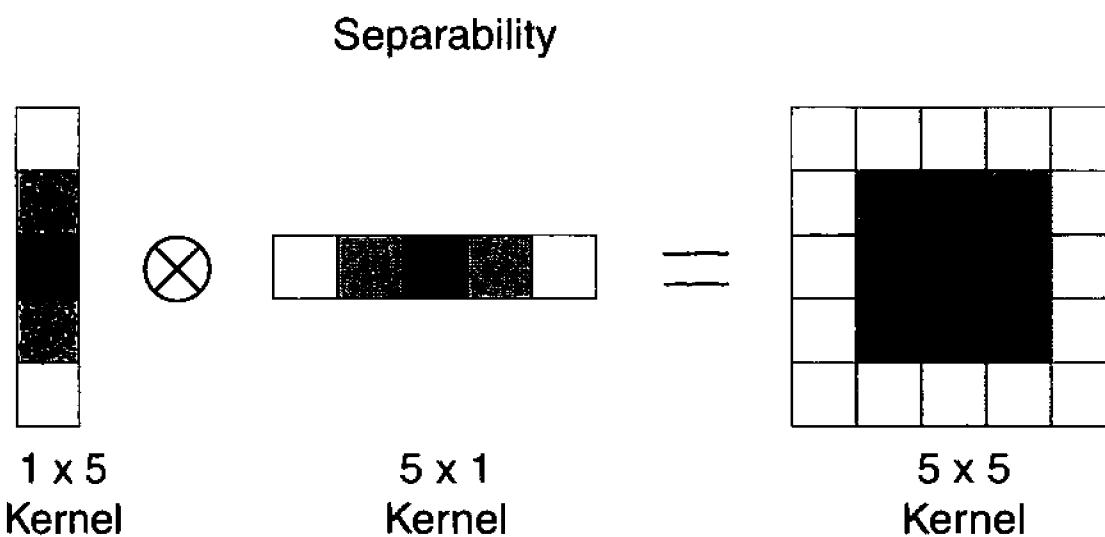
```

In the sections that follow, you will see the associativity property used to create a large kernel from two small ones. If an image were convolved with each of the small kernels in turn, the destination would be the same as convolving the image with the resulting large kernel. You may find it helpful to try to visualize the kernels. The numbers for the pair below represent a donut shape and an averaging kernel. The destination, as you might expect, is a fuzzy donut shape:

$$\begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 2 & 0 & 0 & 0 & 2 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 3 & 4 & 3 & 1 & 0 \\ 1 & 3 & 9 & 14 & 9 & 3 & 1 \\ 3 & 9 & 11 & 6 & 11 & 9 & 3 \\ 4 & 14 & 6 & 4 & 6 & 14 & 4 \\ 3 & 9 & 11 & 6 & 11 & 9 & 3 \\ 1 & 3 & 9 & 14 & 9 & 3 & 1 \\ 0 & 1 & 3 & 4 & 3 & 1 & 0 \end{bmatrix}. \quad (\text{Equ. 14.6})$$

As an exercise, you may wish to verify the accuracy of several of the sample con-

## Section 14.2: Convolution in Two Dimensions



**Figure 14.6** The separability of kernels means a big savings in computer time. To apply a  $5 \times 5$  kernel to a pixel requires 25 multiplications and 25 additions; whereas computing two  $1 \times 5$  kernels in sequence requires only 10 multiplications and 10 additions—a savings of 60% in computer time.

volutions, since it will help you to appreciate the sheer amount of number-crunching involved in a rather simple image processing operation.

**Separability.** Because large kernels can be built up from smaller ones, it should come as no surprise that two one-dimensional kernels applied in succession can produce the same result as a convolution with a single large kernel. This property is extremely valuable in synthesizing the effect of a large kernel used in unsharp masking. Below, a  $5 \times 5$  kernel is synthesized from two  $5 \times 1$  kernels:

$$\begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}. \quad (\text{Equ. 14.7})$$

Convolving one pixel with a  $5 \times 5$  kernel requires 25 multiplications and 25 additions per pixel; whereas convolving the same pixel with a  $1 \times 5$  kernel and then a  $5 \times 1$  kernel requires only 10 multiplications and 10 additions per pixel, a 2.5-time improvement in speed. As kernels become larger, the gain in computational efficiency due to the separability property becomes even greater.

### 14.2.3 Kernels as Spatial Filters

Kernels act as selective spatial “filters” for images. Images can be decomposed into collections of sine waves of varying amplitude; the high-frequency components have peaks that are close together and create the fine detail in the image. The low frequencies have long wavelengths, so that just a few waves fit across the width of an image; they comprise its gross structure. The action of a spatial filter is to block some frequencies while passing others. Filters that block high spatial frequencies are called low-pass filters. Those that block low spatial frequencies and pass high ones are called high-pass filters.

## Chapter 14: Linear Operators

At an intuitive level, it seems obvious that kernels that average together adjacent pixels would attenuate high frequencies without affecting long-wavelength image-spanning frequencies. Kernels that smooth images are called low-pass filters. Likewise, kernels that enhance differences between adjacent pixels while ignoring large-scale features would clearly promote high spatial frequencies. Kernels that detect interpixel differences are high-pass filters.

A brief aside on nomenclature: image processing has drawn concepts and terminology from a variety of disparate fields. Electrical engineering, mathematics, signal processing, and information theory have all played important roles in its development; therefore the nomenclature is a mixture of terminology from many disciplines. In one textbook, neighborhood processes are done with kernels, but in another you will find they are called filters, and in another, they are called operators. In this book, the terms are used interchangeably, as they are in the world of image processing.

In the following sections, we discuss operators (kernels) for smoothing and sharpening images. While recognizing their effects in the frequency domain, the discussion is focused on the spatial behavior of kernels. For the interested reader, frequency filtering is covered in detail in Chapter 17, on the Fourier transform.

### 14.2.3.1 Smoothing Kernels (Low-Pass Filters)

Smoothing kernels average the central pixel with those in its immediate neighborhood. Smoothing is sometimes used to reduce noise; but unfortunately, it also averages out detail in the image. The relative “pass-through” rating of each of the kernels below is the normalized value of the center element. The lower the pass-through, the more effectively the filter blocks high frequencies. This rating is a measure of how much weight the central pixel receives relative to the pixels in the neighborhood.

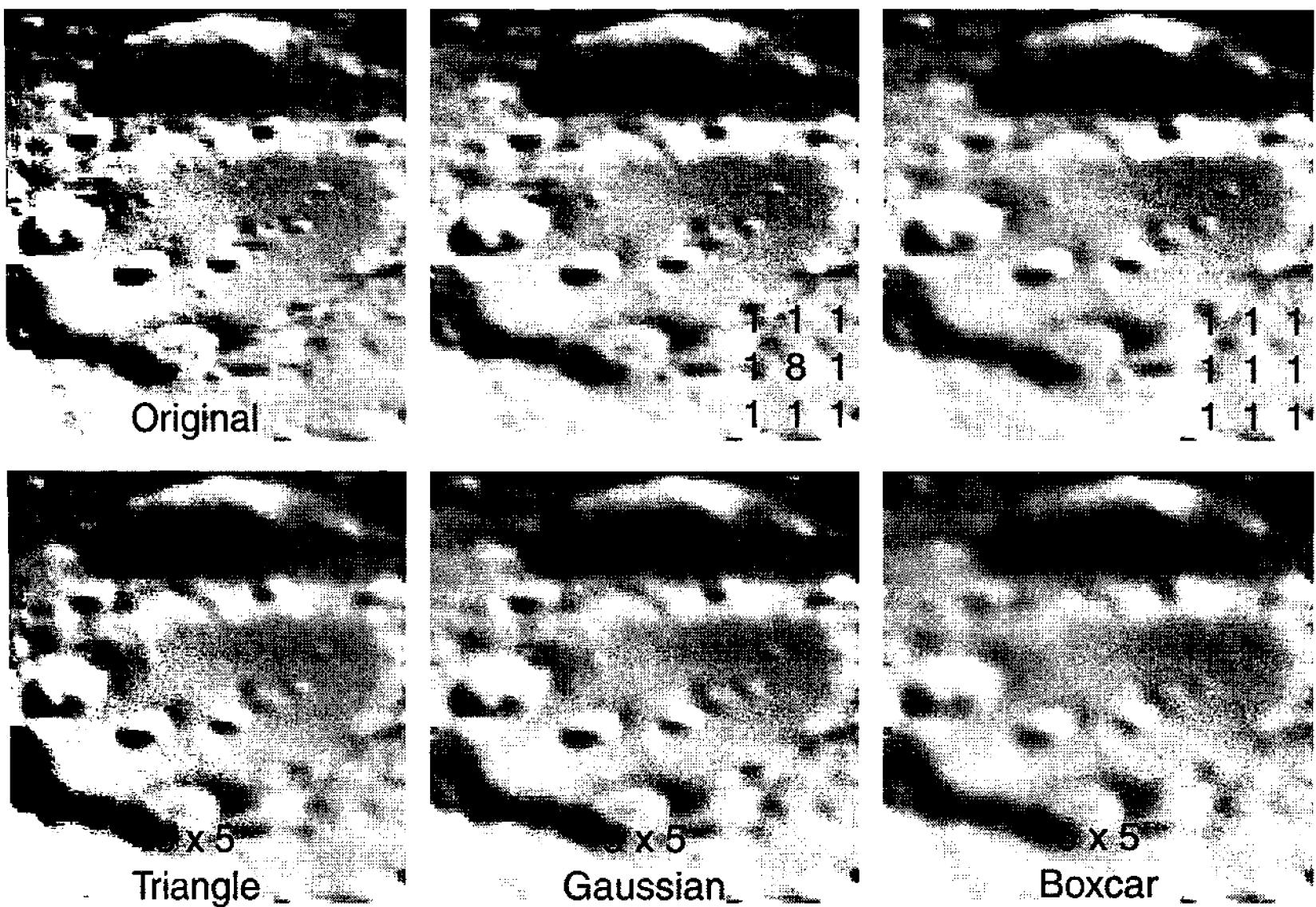
$$\text{The “boxcar” blur kernel: } \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

gives all pixels equal weighting. It is the most “egalitarian” of the smoothing kernels, and can lead to severe loss of small-scale detail in an image that appears sharp. Its pass-through rating is 0.111.

$$\text{This 50%-blur kernel: } \frac{1}{10} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

gives each of the surrounding pixels equal weight, but gives the more important central pixel twice the weight. It preserves more detail while still softening the image. The pass-through is 0.2.

## Section 14.2: Convolution in Two Dimensions



**Figure 14.7** The action of blur kernels is easy to understand—each pixel in the new image is the kernel-weighted average of a group of pixels in the old image. In the sequence above, progressively more weight is given to outer neighbors over inner neighbors. Image of the lunar crater Clavius by Steve Lee.

$$\text{The gentle blur kernel: } \frac{1}{16} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

endows the central pixel with the same weight as the sum of the other eight pixels in the neighborhood. The smoothing effect is gentle enough that it should be called a “softening” kernel. The filter pass-through rating is 0.5.

$$\text{The minimal blur kernel: } \frac{1}{12} \times \begin{bmatrix} 0 & 1 & 0 \\ 1 & 8 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

credits the central pixel twice the input of the four adjacent pixels, so the effect from applying it to an image is minimal: it will give a harsh-looking image a slightly “gentled” appearance. Its pass-through rating is 0.667.

$$\text{The } 3 \times 3 \text{ Gaussian blur: } \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

applies a smoothly weighted average of the pixels in the neighborhood. This ker-

## Chapter 14: Linear Operators

nel is equivalent to a truncated Gaussian smoothing with a radius of 0.25 pixels. It passes 0.25 of the central pixel.

$$\text{The “hole in the donut” kernel: } \frac{1}{8} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & \mathbf{0} & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

creates a new central pixel that totally ignores the value of the central pixel in the old one. If this filter is applied to an extremely noisy image, it will eliminate noise spikes; or more precisely, it will spread each noise spike into a fairly inconspicuous ring of eight pixels. Its pass-through rating is zero.

Larger smoothing kernels work in a similar fashion, but they operate over a larger neighborhood; so the potential for the desired smoothing and undesired loss of detail are both greater.

$$\text{The 5x5 “boxcar” kernel: } \frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & \mathbf{1} & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

is the egalitarian averaging kernel all over again, but in a bigger format. The central pixel has a pass-through rating of 0.04, so this kernel has a powerful (and indiscriminate) smoothing action. The term “boxcar” refers to the square sides and flat top. As a spatial filter, it leaks high frequencies because it has abrupt edges.

$$\text{This 5x5 “triangle” kernel: } \frac{1}{19} \times \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & \mathbf{3} & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

gets its name from its flat-sided, pointy-top profile. With a pass-through rating of 0.16, it is a strong filter, but also lets high-frequency detail leak through.

$$\text{The 5x5 “Gaussian” kernel: } \frac{1}{256} \times \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & \mathbf{36} & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

is already familiar from some of the examples above. It rates a pass-through of 0.14, so its filtering action is strong; but the weights are smoothly distributed over

## Section 14.2: Convolution in Two Dimensions

the surrounding neighborhood. As a filter, it boasts a smooth transition from passing low-frequency structure to effectively blocking high-frequency detail. Because of this, the Gaussian kernel produces the smoothest smoothing.

It is very important to understand that the effect of a filter depends critically on the sharpness of detail in the image. If the image contains significant detail on one-pixel-to-the-next scale, then applying a smoothing filter will reduce the detail present. However, if the image has already been convolved with a smoothing function such as atmospheric turbulence, then the appropriate smoothing filter—one in which the radius of the smoothing kernel is about 0.6 times the full-width half-maximum of the point-spread function of the image—can reduce image noise without producing a significant loss in the (already reduced) image sharpness.

### 14.2.3.2 Sharpening Kernels (High-Pass Filters)

Sharpening kernels increase the difference between a central pixel and the pixels in its immediate neighborhood. Sharpening enhances the contrast of detail in an image, but also increases the visibility of noise in it. Most sharpening filters are designed so that the sum of the elements of the kernel is 1. A useful measure of the filter's strength is its contrast enhancement, which equals the value of the central pixel divided by the sum of the elements in the filter.

Technically, sharpening filters and high-pass filters are not identical. A high-pass filter passes high spatial frequencies and blocks low ones, but a sharpening filter detects high spatial frequencies and adds them to the existing image. The output of a high-pass filter is an image containing *only* high frequencies, whereas the output of a sharpening filter is a copy of the source with the high spatial frequencies enhanced so that the image looks sharper and more detailed.

The effect of a high-pass filter depends critically on the sharpness of detail in the image. If the image contains significant detail on a one-pixel-to-the-next scale, then applying a sharpening filter enhances the contrast of the detail markedly. However, if the width of the filter kernel is larger than the point-spread function inherent in the image, the convolution increases the visibility of the noise in the image without producing an improvement in the contrast of image detail.

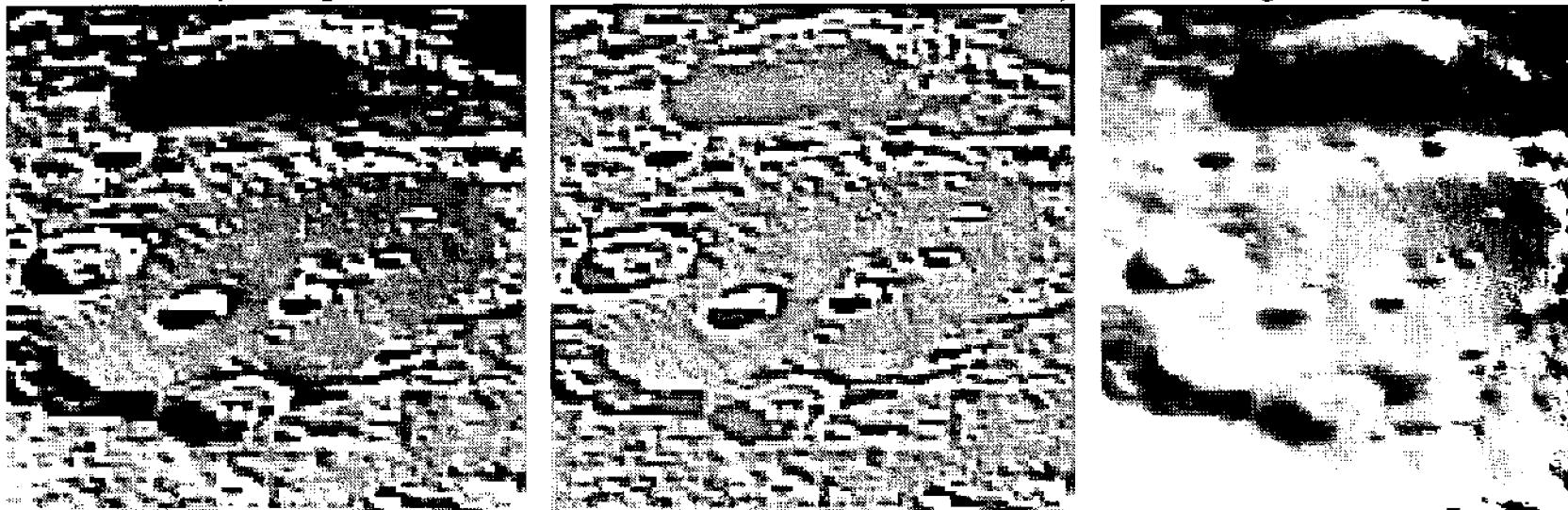
The classic sharpening kernel:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

enhances the contrast of small-scale detail by a factor of 9. Its effect can be dramatic, as if a layer of mushy “fog” covering the source image were suddenly removed. The filter's operation is difficult to grasp at first sight. It is useful to envision the filter as consisting of two components, using its linearity property, thus:

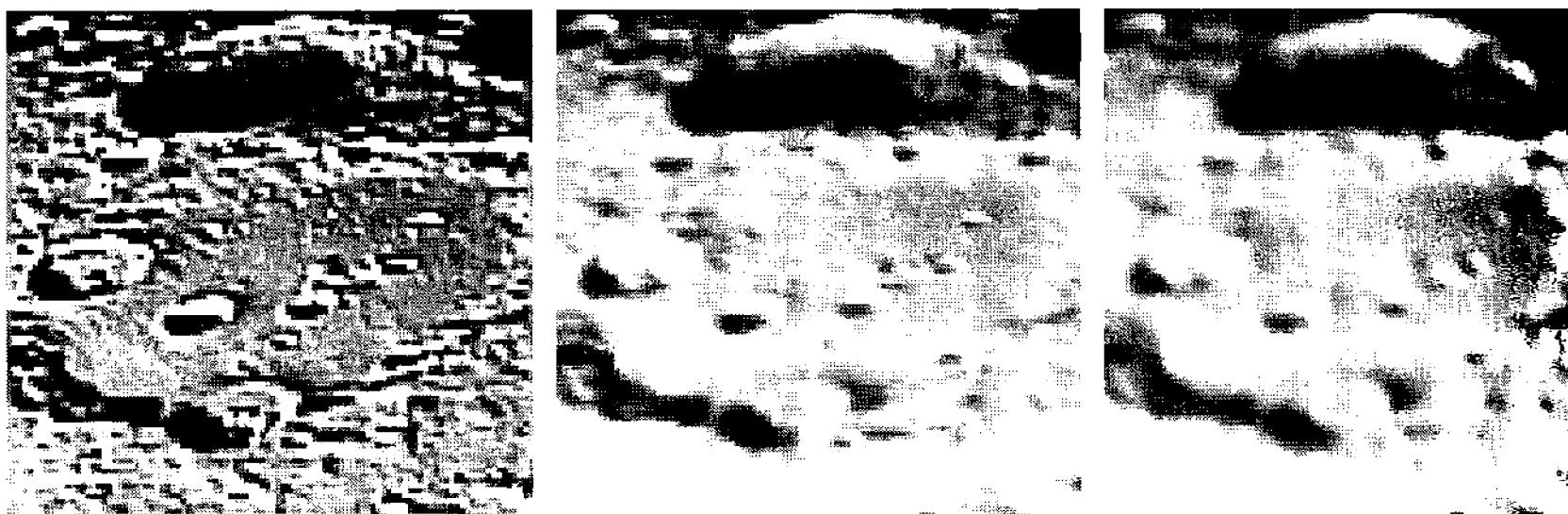
## Chapter 14: Linear Operators

Sharpening seen as the sum of the local difference plus the original image.



$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Sharpening seen as the original image minus a blurred copy of the original image.



$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

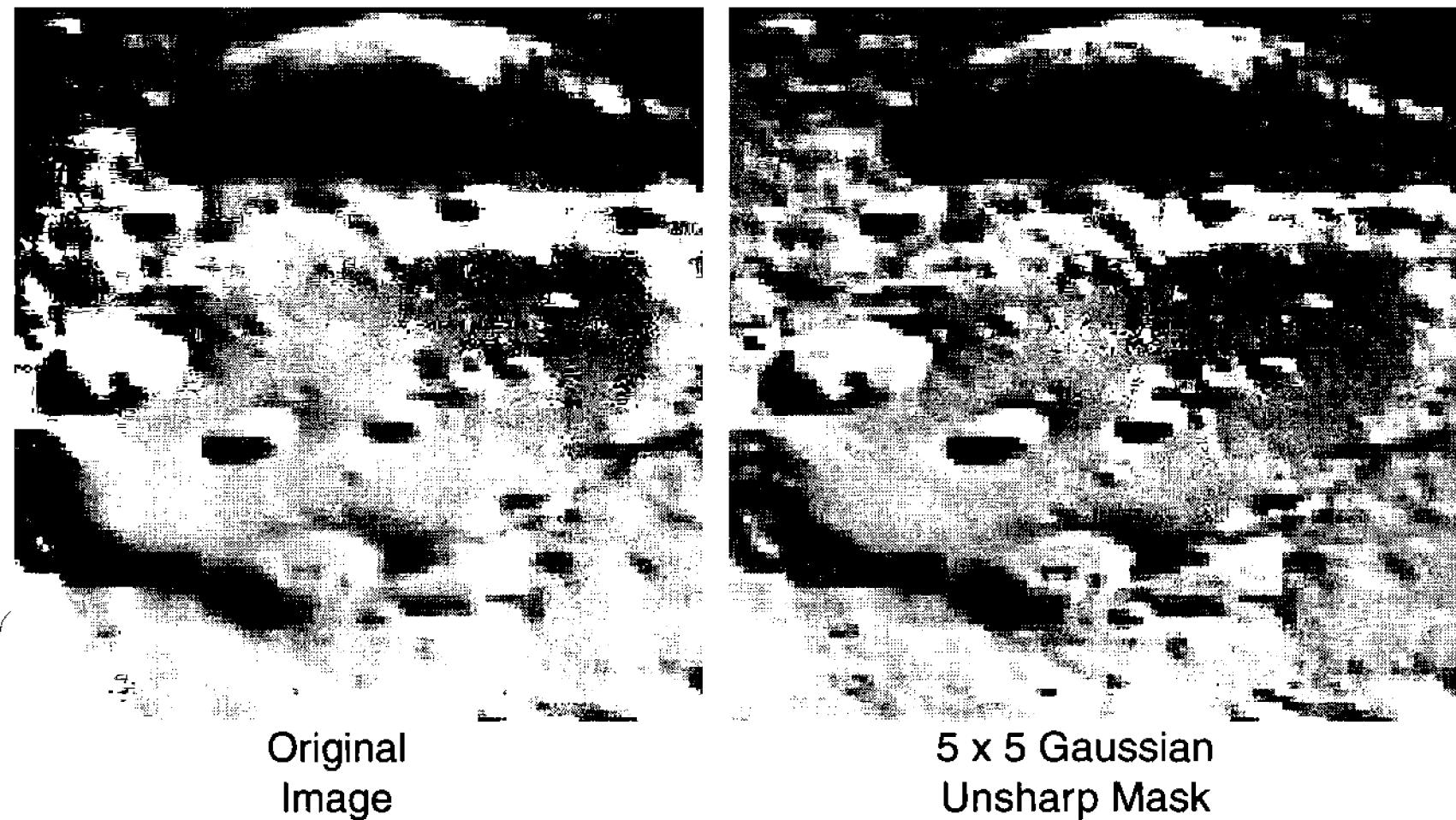
Figure 14.8 Whether viewed as local difference plus the original or as an original minus a blurred image, the effect is the same: Sharpening heightens the difference between each pixel and its neighbors. In these examples, the image scale is large so that you can see how sharpening works on a pixel-by-pixel basis.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (\text{Equ. 14.8})$$

The kernel appears to work by adding together a kernel that computes the amount by which the central pixel deviates from its surrounding neighborhood to the original image. If the pixels in the neighborhood have the same value, the deviation is zero; but if the central pixel is higher or lower than the surrounding ones, the difference is multiplied by 8 and added to the original image. As a result, small deviations in the central pixel's deviation from the norm are enhanced.

Alternatively, you can argue that the kernel works by *subtracting* the average of the central pixel's neighborhood from a large multiple of the pixel:

## Section 14.2: Convolution in Two Dimensions



**Figure 14.9** At large scale, you can see how sharpening changes individual pixels. Those brighter than their neighbors become brighter, and pixels darker than their neighbors become darker; but the overall image brightness remains the same. As a result, low-contrast features become easier to see.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (\text{Equ. 14.9})$$

If the pixels in the neighborhood have the same value, the result of the convolution is a new pixel with the same value as the source. However, if the pixel is even slightly higher or lower than those in the neighborhood, the small difference is multiplied by a factor of 9—it becomes a large difference in the destination image. In either case, the result is the same.

Crispening kernels:  $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$  and  $\begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$

provide a milder degree of contrast enhancement, hence the name. Unless sharpening and crispening kernels are used intelligently, they can leave a dark ring around bright objects like stars, the result of the ring of negative elements in the kernel.

Because most astronomical images are fairly “soft,” a large kernel is usually better for sharpening because the comparison pixels are drawn from a wider area:

## Chapter 14: Linear Operators

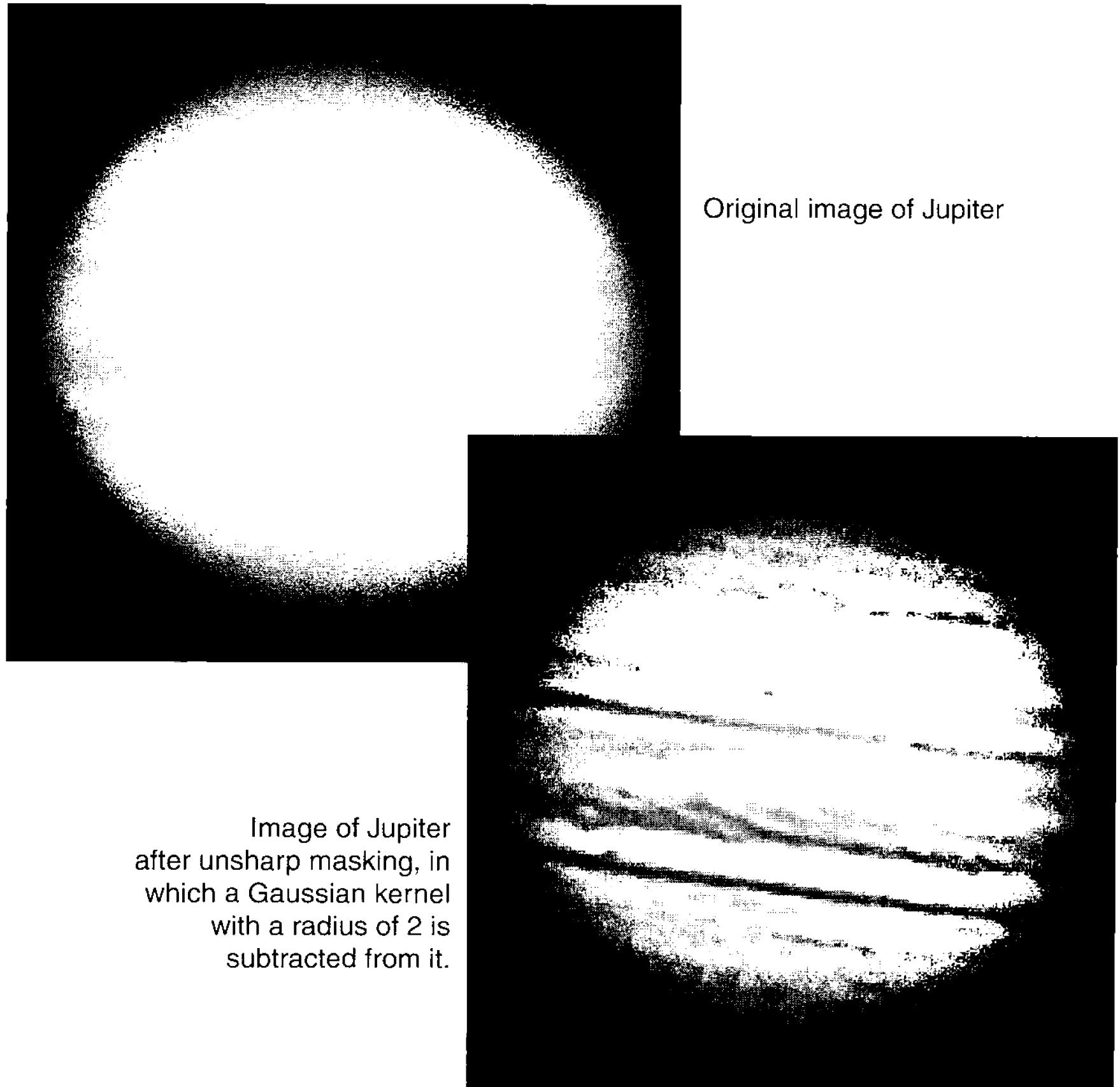


Figure 14.10 The original image of Jupiter is full of detail but lacks contrast between the light and dark tones. Subtracting a soft Gaussian mask increased the contrast difference between the features and their surroundings, producing an image in which more detail is visible. Jupiter image by Donald Parker.

$$\begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -4 & -16 & -24 & -16 & -4 \\ -6 & -24 & \mathbf{220} & -24 & -6 \\ -4 & -16 & -24 & -16 & -4 \\ -1 & -4 & -6 & -4 & -1 \end{bmatrix}.$$

Sharpening kernels and unsharp masking are essentially the same, and they produce the same results. Both methods are capable of producing remarkable enhancements in planetary and lunar images. However, as you will see below, unsharp masking is not only more efficient and conceptually simpler, but it is more flexible because it allows the user direct control over the degree of contrast enhancement.

### 14.2.4 Edge-Detection and Gradient Kernels

In many image-processing applications, it is important to detect boundaries and edges. Although astronomical images have few well-defined edges—nebulae and galaxies blend smoothly into the sky background—it is nonetheless useful to understand the classical methods used in image processing software for detecting and highlighting boundaries, edges, lines, and borders.

#### 14.2.4.1 Bas-Relief Operators

One of simplest image operators is the bas-relief kernel. (The word “bas” in **bas-relief** is pronounced “bah”; it means “low” in French because the output looks like the image has been carved in low relief.) Its action is to make a copy of the image, shift it by one pixel, and subtract it from the original. Because the sum of the kernel is zero, a constant is usually added to the destination of the convolution to produce an image with positive pixel values.

$$\text{Bas-relief operators are: } \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and: } \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and: } \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

These kernels produce interesting three-dimensional effects. The output of the first kernel looks as if the light came from above; the second like the light came from the left. (In the literature, the first is sometimes called the north operator; the second, the west operator; and the third, the northwest operator.) The  $-1$  element can take any position in the eight-pixel neighborhood, with a corresponding change in the apparent position of the light source.

A close relative of the bas-relief operator is the embossing operator. Its output resembles a photograph pasted to a wall carved in relief and illuminated from a low angle. Instead of one  $-1$  element, there are three, as well as three  $+1$  elements. By themselves, these elements would produce a bas-relief image, but the linearity property allows adding a central element that copies the image into output.

$$\text{Embossing kernels are: } \begin{bmatrix} -1 & -1 & -1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & 0 & 1 \\ -1 & 1 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

The first operator makes an image that appears to be illuminated from above (north), the second from the left (west), and the third from the upper left (northwest). Eight basic embossing operators exist.

For special-purpose image displays, it's easy to create linear combinations and kernels in larger sizes:

## Chapter 14: Linear Operators

$$\frac{1}{32} \times \begin{bmatrix} -2 & -5 & -4 & -1 & 0 \\ -5 & -10 & -4 & 2 & 1 \\ -4 & -4 & 8 & 12 & 4 \\ -1 & 2 & 12 & 14 & 5 \\ 0 & 1 & 4 & 5 & 2 \end{bmatrix}.$$

Larger sizes offer greater immunity to noise than small kernels, and may be worth pursuing on those special occasions when an unusual type of detail must be teased from an image.

It is important to remember that the initial output for a bas-relief or embossing operation may have a very small range of pixel values, especially if the source image is low in contrast. Do not be surprised if you must perform a linear brightness scaling before you can see the result.

### 14.2.4.2 Sobel, Kirsch, and Prewitt Operators

A number of engineers and scientists have described schemes for detecting features in images, most notably the sudden brightness changes that occur at their boundaries and edges. Among the three best known edge operators are those of Sobel, Kirsch, and Prewitt. Their kernels are:

$$\text{Sobel operators: } H = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ and } V = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix};$$

$$\text{Kirsch operators: } H = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \text{ and } V = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix};$$

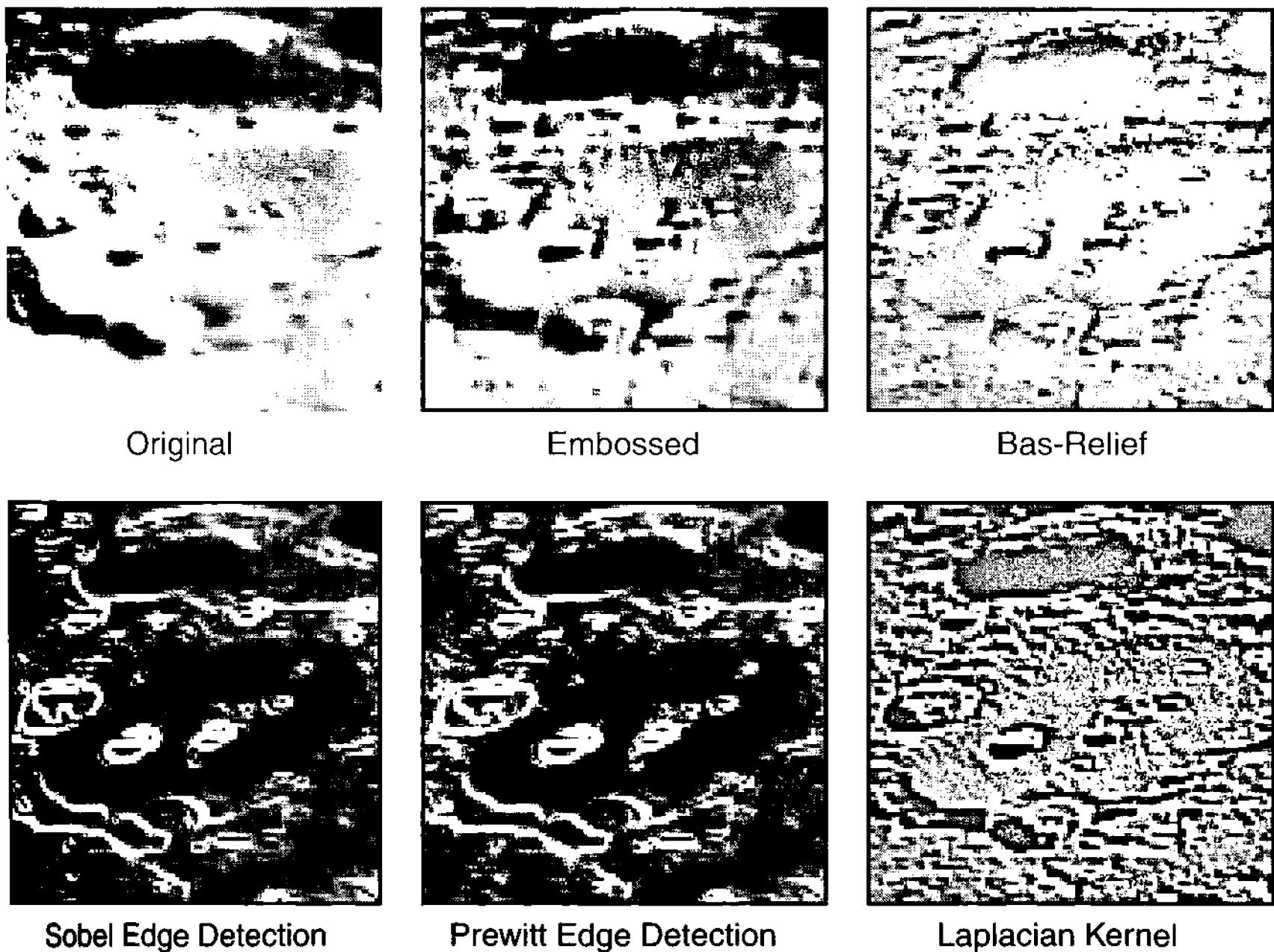
$$\text{Prewitt operators: } H = \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ and } V = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}.$$

These should be instantly recognizable as variants on the embossing operator. In each case,  $H$  detects horizontal edges and  $V$  detects vertical edges. To detect edges at all orientations,  $H$  and  $V$  are computed separately and then combined quadratically:  $I = \sqrt{H^2 + V^2}$ . The output,  $I$ , is always positive because  $H$  and  $V$  are squared when they are combined.

Two of the eight orientations possible for each kernel are shown. In the remaining six orientations, the outer row of elements is shifted around the kernel. To detect edges with specific desired orientations, you must select the kernel that most closely matches the desired orientation of the features sought.

The images produced by edge detection algorithms usually require further

## Section 14.2: Convolution in Two Dimensions



**Figure 14.11** Linear operators perform many different jobs. Embossing and bas-relief use local differences to create the appearance of light and shadow. Sobel and Prewitt kernels respond to neighborhoods with edge-like patterns. The Laplacian operator detects any difference between a pixel and its neighbors.

processing. The range of values in the output image may be small, calling for a linear stretch; or the values may be crowded into a narrow part of the histogram, requiring a nonlinear brightness scaling.

### 14.2.4.3 Line-Detection Operators

The Sobel, Kirsch, and Prewitt operators detect edges in images—they are sensitive to curves, points of light, and a variety of other abrupt changes in pixel value. To perform a specific task like detecting lines in an image, the kernel should be shaped like the feature sought. This kernel responds to horizontal lines:

$$\text{Horizontal line detection: } \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

To detect lines with another orientation, the long axis of the kernel must be parallel with the line. An exhaustive search for lines in an image would require applying a series of kernels rotated through 180 degrees, followed by an algorithm designed to search for the highest output values in the resulting images.

## Chapter 14: Linear Operators

### 14.2.4.4 The Laplacian Operator

The Laplacian operator detects any rapid changes in pixel values in an image, regardless of the orientation of the change. Technically, the Laplacian is the sum of the second partial derivatives translated into the spatially quantized realm of image processing.

$$\text{The Laplacian operator: } \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

bears a strong family resemblance to the sharpening kernel because, in fact, the sharpening kernel is the linear sum of the Laplacian and the unity operator:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (\text{Equ. 14.10})$$

If you use the Laplacian operator to create a temporary image and add that one to the source image, what you get is the same as convolving the source image with the classic sharpening filter.

## 14.3 Convolution by Unsharp Mask

Unsharp masking is a convolution technique that, at first sight, appears to be something else. It works because the linearity property of convolution allows you to break a kernel into additive component kernels. First, the image is convolved with a kernel that averages pixel values over a region:

$$\frac{1}{256} \times \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}.$$

The soft image is called the unsharp mask, after the masking process used during the last century by photographers and photolithographers. Next, the image is convolved with an expanded version of the unity kernel:

## Section 14.3: Convolution by Unsharp Mask

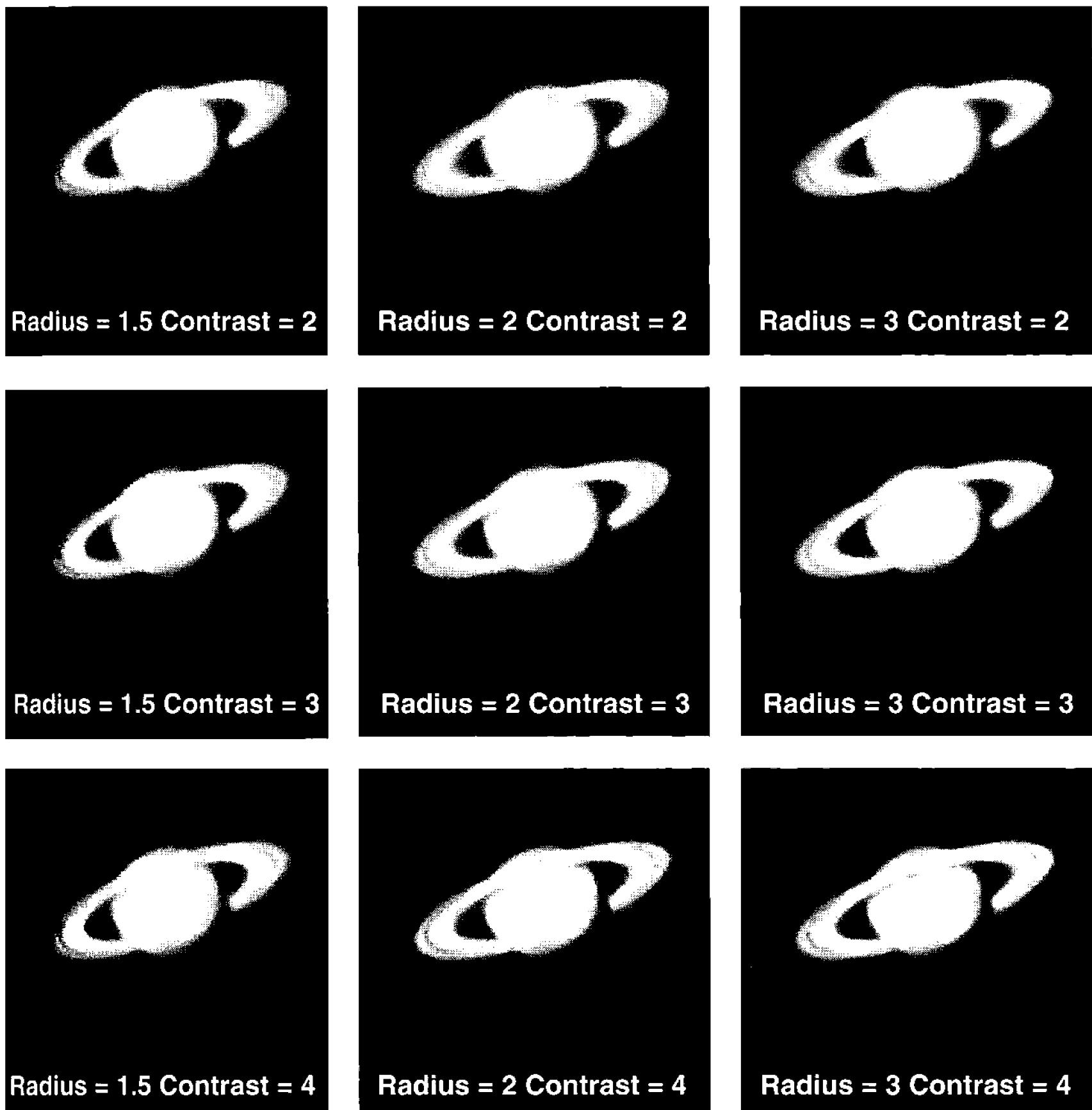


Figure 14.12 Unsharp masking depends on the radius of the mask and the contrast enhancement. A Gaussian mask with a radius of 1.5 pixels is too small to be effective, and a radius of 3 is somewhat too large. Good things happen when the kernel matches the size of the atmospheric point-spread function.

$$\frac{1}{256} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{256} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Finally, the two images are summed. Since convolution is linear, it is legitimate to multiply the values in each image by a constant. Suppose that we multiply the values in the image generated by the unity kernel by a contrast factor,  $c$ , (such as 3); multiply the unsharp mask image by  $-(c - 1)$  (i.e., -2); add the two images; and normalize the image by the sum of the elements in the mask.

This operation produces the same destination image as the mask:

## Chapter 14: Linear Operators

$$\frac{1}{256} \times \begin{bmatrix} -2 & -8 & -12 & -8 & -2 \\ -8 & -32 & -48 & -32 & -8 \\ -12 & -48 & \mathbf{696} & -48 & -12 \\ -8 & -32 & -48 & -32 & -8 \\ -2 & -8 & -12 & -8 & -2 \end{bmatrix}.$$

To understand how it works, consider how unsharp masking alters the low and high frequency content of an image. The unity image contains both high and low frequencies, but the unsharp mask has been stripped of high frequencies. When the unity image is multiplied by the contrast factor  $c$ , the strengths of the high and low frequencies are both multiplied by that factor. When the mask, containing only the low frequencies, is multiplied by  $-(c - 1)$ , the low frequencies become negative. When the unity image and the mask are summed, the resulting image has the same low-frequency content as the original; but its high-frequency content has been multiplied by the contrast factor.

Looking ahead briefly to the spatial frequency realm (see Chapter 17), unsharp masking acts as a high-frequency enhancement filter. It retains the low spatial frequencies from the original image, but multiplies the detail-filled high spatial frequencies by the contrast factor.

### 14.4 Generated Kernels for Unsharp Masking

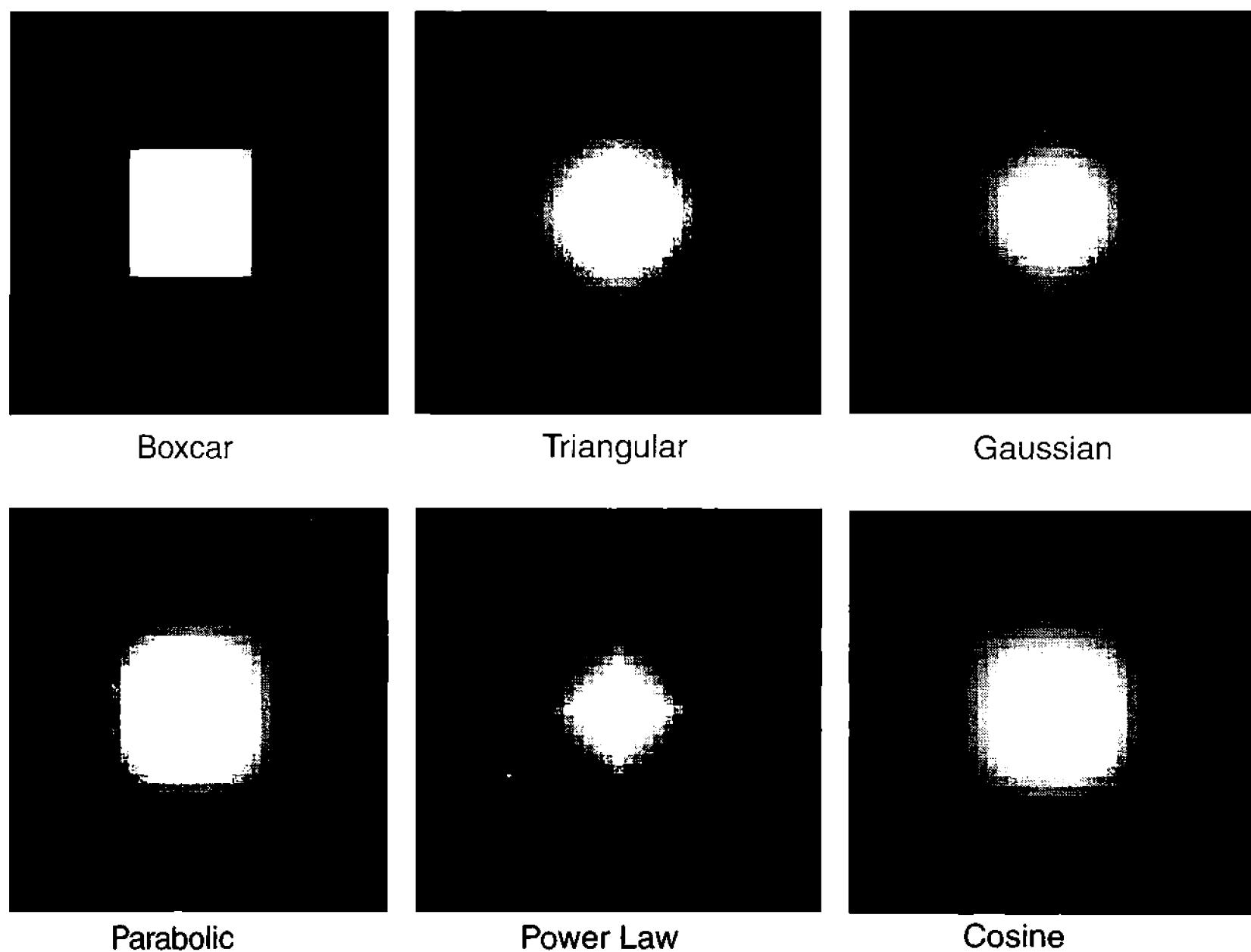
The kernels discussed above tend to have small dimensions, but they can easily be made as large as needed. Large kernels are used to enhance the medium-scale spatial features that 3x3, 5x5, and 7x7 ones scarcely touch. Because of their separability, large kernels can be synthesized by performing successive passes with appropriate one-dimensional kernels. Furthermore, the one-dimensional kernels can be generated algorithmically, to match the size and shape of the particular feature or features slated for enhancement.

The generated kernels described below—boxcar, triangular, Gaussian, and power-law—have properties that depend on the shape of the mask. Of the four, the Gaussian mask is the most useful because it tapers evenly at the edges, smoothly enhancing the high spatial frequencies. Each mask is assigned an effective radius that is designed so that half the sum of the elements in the kernel lies inside the mask and half outside it; however, because the distribution of element values is so different, the size of the generated kernels varies greatly.

#### 14.4.1 The Boxcar Unsharp Mask

The egalitarian “boxcar” kernel assigns equal weight to all pixels when the radius is an integer, and gives a fractional weight to the outermost elements when the radius is non-integer. For half the sum of the elements to mark the effective radius, its value must be 71% of the kernel size. To produce an effective radius of 4.5 pixels, elements 0 through 7 of the boxcar kernel are:

## Section 14.4: Generated Kernels for Unsharp Masking



**Figure 14.13** Unsharp mask kernels come in a variety of shapes. For good results, a mask must match the point-spread function in the original image. As a general rule, the Gaussian mask best matches the blur in planetary and lunar images, while triangular and power-law masks work for some deep-sky images.

1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 0.435.

Thus, a boxcar mask with an effective radius of 4.5 has a 15 x 15 kernel.

For astronomical images, the boxcar is a crude tool. It produces square shadow artifacts around bright star images and “rings” badly; i.e., it produces periodic banding parallel to horizontal and vertical edges in the image. In the frequency domain, it blocks some high frequencies but passes others. The high frequencies that are retained cause the ringing.

### 14.4.2 The Triangular Unsharp Mask

In the one-dimensional triangular kernel, element values decline linearly from the center to the edge. However, because two one-dimensional kernels cannot be convolved to produce a true two-dimensional triangular kernel, the separable triangular unsharp mask is necessarily an approximation. The effective radius is about half the kernel size. Here are elements 0 through 10 of a triangular kernel with an effective radius of 4.5 pixels:

1.000, 0.908, 0.817, 0.725, 0.633, 0.541, 0.450, 0.378, 0.266, 0.174, 0.083.

This triangular mask, with an effective radius of 4.5, employs a 21 x 21 kernel.

The triangular unsharp mask is useful with astronomical images. When its



Figure 14.14 Before unsharp masking, the stars in this image of the Horsehead Nebula originally looked a bit soft. Applying a very gentle Gaussian unsharp mask with a radius smaller than the star images helped make them look a little sharper, smaller, and brighter.

effective radius is chosen to match the size of the desired features, it produces a “good looking” gain at high frequencies. In the frequency domain, the triangular mask blocks some high frequencies and passes others; so there is some mild ringing; but the effect is not unpleasant in, for example, lunar images.

### 14.4.3 The Gaussian Unsharp Mask

Wherever random processes play, the universe generates Gaussian blurs. The Gaussian unsharp mask is superbly well suited for enhancing the high-frequency information that lies hidden beneath the Gaussian blur in soft star images, fuzzy planet pictures, and mushy Moon shots.

However, because the Gaussian function tapers off slowly, Gaussian kernels can be very large. Fortunately, a two-dimensional Gaussian kernel can be separated into two one-dimensional kernels; that is, convolution with two one-dimensional Gaussian kernels produces exactly the same result as convolution with one two-dimensional Gaussian kernel. Consider a Gaussian blur having an effective radius of 4.5 pixels:

$$1.000, 0.976, 0.901, 0.801, 0.674, 0.539, 0.411, 0.298, 0.206, 0.135, 0.085, \\ 0.050, 0.029, 0.015, 0.008, 0.004.$$

Convolution with this function as a two-dimensional  $31 \times 31$  kernel requires 961 multiplications and additions; but as two one-dimensional kernels, only 61 multiplications and additions are needed.



Figure 14.15 Applying an unsharp mask that is larger than the star images generates conspicuous shadows around bright objects, enhances noise in the sky background, and can cause the image to look artificial. However, choosing the “correct” unsharp mask is largely a matter of personal preference.

The Gaussian unsharp mask is an all-purpose tool for astronomy. It can be applied to sharpen any image blurred by random processes such as atmospheric turbulence. In the frequency domain, the Gaussian mask provides a smoothly increasing enhancement toward high frequencies.

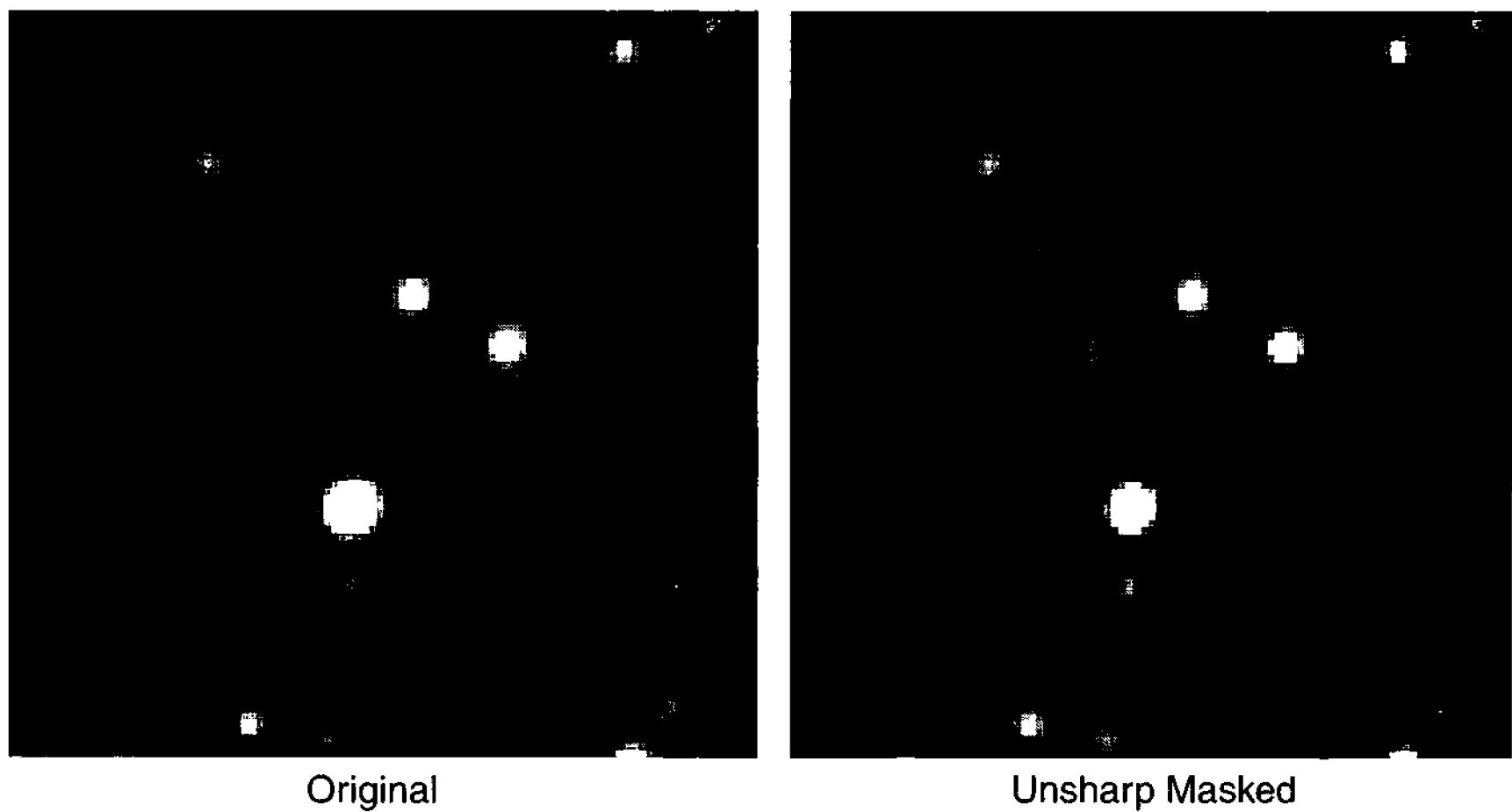
Selecting the effective radius is extremely important in obtaining full benefit with the Gaussian technique. If the effective radius of the mask lies between 1.4 and 1.7 times that of the atmospheric blur, the image is sharpened without obvious artifacts or ringing. If the effective radius of the mask is equal to or smaller than that of the blurring, enhancement is greatly reduced; and if the effective radius is larger than twice the radius of the blur, artifacts such as dark rings around bright stars may appear.

It is also important to select the contrast factor with care. Unsharp masking enhances image noise along with real image detail. Convolutions, no matter how they are done, have no way to distinguish between pixel value differences generated by random noise and those due to light falling on the detector. For most images, contrast enhancement by a factor between 1.5 and 5 reveals all the image detail possible with unsharp masking.

#### 14.4.4 The Power-Law Unsharp Mask

In the one-dimensional power-law (or “exponential”) kernel, element values decline exponentially from the center. In the two-dimensional kernel, element values

## Chapter 14: Linear Operators



**Figure 14.16** These enlargements show how proper unsharp masking changes the appearance of star images: the outer edges are slightly suppressed while the image core is made a little brighter. Overprocessing deep-sky images usually results in blocky-looking stars surrounded by dark rings.

decline exponentially only along the axes; between the axes, the drop is more rapid than exponential.

Because element values fall so rapidly from the center, the effective radius is roughly one-sixth the kernel size. For an effective radius of 4.5 pixels, the kernel must have a radius of 29:

1.000, 0.777, 0.604, 0.469, 0.364, 0.283, 0.220, 0.171, 0.133,  
0.103, 0.080, 0.062, 0.048, 0.036, 0.029, 0.023, 0.018, 0.014,  
0.011, 0.008, 0.006, 0.005, 0.004, 0.003, 0.002,....

Power-law unsharp masks have interesting properties when applied to astronomical images. The sharp core region enhances small detail while the long exponential tail enhances low-frequency structure, so an exponential mask with an effective radius of 20 pixels gives deep-sky images a pleasant “boost” at both low and high spatial frequencies.

### 14.4.5 Other Unsharp Masks

The best unsharp masking kernel to use depends on the characteristics of the point-spread function in your images. Fortunately, there is no limit to the number of unsharp mask profiles possible. A basic set should have enough variety to match different sorts of images. The indispensable one is the Gaussian profile; it mimics the random distribution of atmospheric motions, and is thus similar to well-sampled star image profiles.

The triangular and power-law profiles fall off more rapidly than the Gauss-

## Section 14.4: Generated Kernels for Unsharp Masking

ian one. They match the slightly aberrated images typical of refracting optical systems—telephoto lenses and some refractors—which often have a sharp core surrounded by a soft blur.

The boxcar profile is flat-topped, a characteristic of images that are slightly out of focus. Completing the suite of profiles are the parabolic and cosine profiles that fall off less rapidly than the Gaussian but more rapidly than the boxcar. These profiles match star images that are slightly enlarged from less-than-perfect focus and atmospheric turbulence.

## Chapter 14: Linear Operators

# 15 Non-Linear Operators

The previous chapter described neighborhood processes using linear operators. This chapter explores the properties of non-linear operators. These powerful and sometimes perplexing processes are extremely useful in astronomy.

Unlike linear operators, non-linear processes usually include a logical “if... then” operation. Because these built-in conditions can cause abrupt discontinuities, the output values cannot be traced back to unique input values.

The distinguishing feature of conditional operators is that they are not commutative; that is, the order of application matters. The linear filters discussed in the last chapter are commutative; that is, if you apply two linear operators, it doesn’t matter which is applied first—the result will be the same.

Consider two one-dimensional, non-linear operators,  $A$  and  $B$ :

$$A_i = \begin{cases} S_i \leq S_{i+1} \rightarrow 100 \\ \text{else} \rightarrow 0 \end{cases} \text{ and } B_i = \begin{cases} (S_i = S_{i+1}) \rightarrow 100 \\ \text{else} \rightarrow 0 \end{cases}.$$

In process  $A$ , the  $i$ -th pixel in the destination image is 100 if the  $i$ -th pixel in the source image is less than or equal to the  $(i+1)$ -st pixel; otherwise, the result is 0. In process  $B$ , the output is 100 if the  $i$ -th pixel in the source image is equal to the  $(i+1)$ -st pixel. These filters contain elementary logical operations.

Suppose that you apply these operations to a one-dimensional source image:

$$\dots 17 24 16 16 11 67 21 45 83 19 \dots.$$

The result of applying filter  $A$  first, followed by filter  $B$  is:

$$\dots 0 0 0 0 0 0 0 100 0 \dots \dots.$$

And the result of applying filter  $B$  first and then applying filter  $A$  is:

$$\dots 100 100 0 100 100 100 100 100 100 \dots \dots.$$

With non-linear filters, it is clear that—unlike linear operators—the result de-

## Chapter 15: Non-Linear Operators

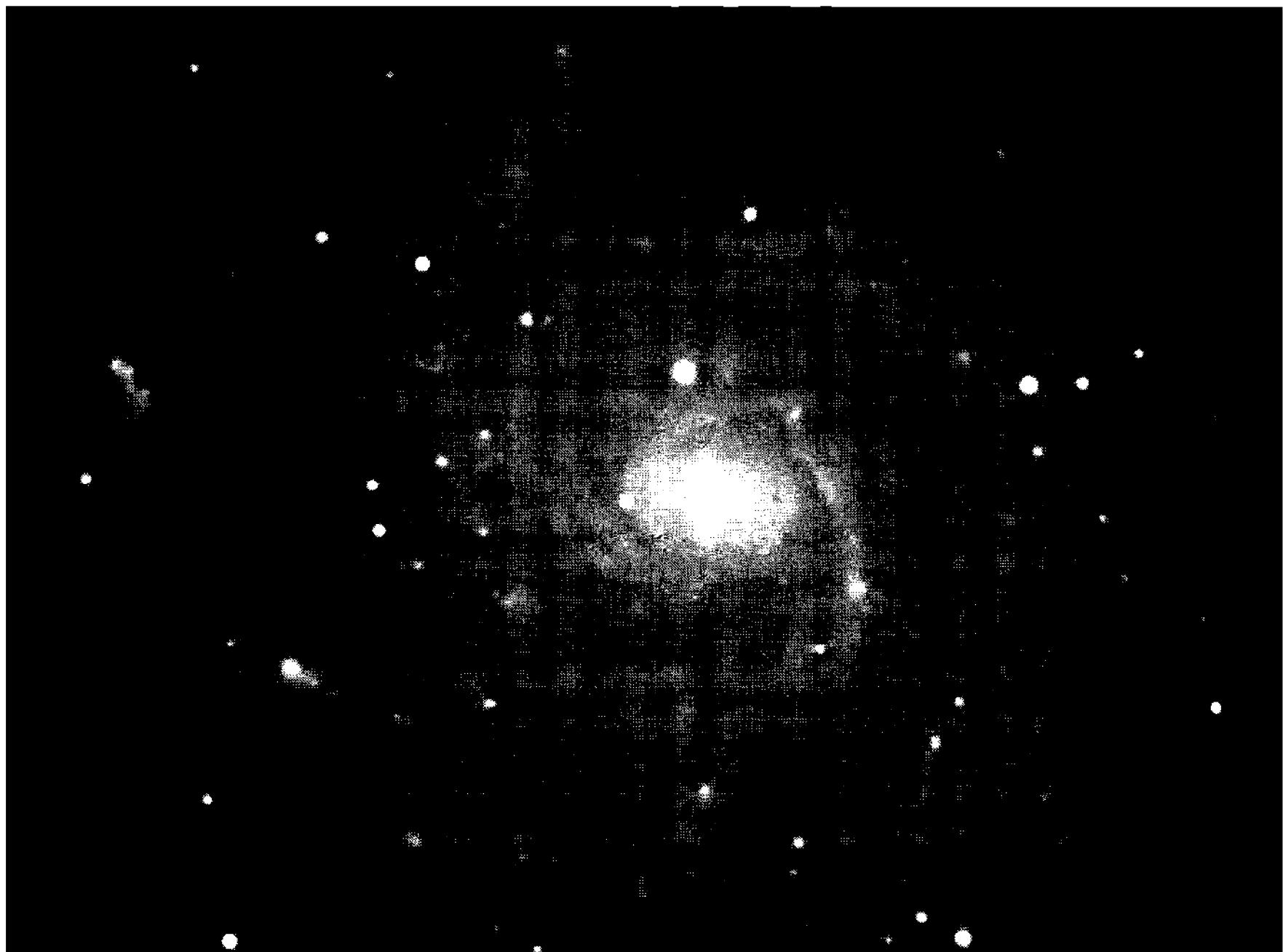


Figure 15.1 Case study: Conditional operators are powerful tools for enhancing deep-sky images. In the following pages, you will see variations on this unprocessed view of the spiral galaxy M101. The image appears smooth and grainless, the result of its high signal-to-noise ratio. Image by Rob West.

pends on the order in which the operations are applied.

A handy way to think about conditional operators is to imagine that the logical operation acts like a trap door. Image information passes through the operator in one direction only. To put it another way, when an image is processed by a conditional operator, the original image cannot be retrieved from the result. Linear operators usually leave some trace of the original image information, while conditional operators obliterate much of it.

### 15.1 Rank Operators

The pixel values in a neighborhood are usually spread over some range, from a lowest to a highest value. Rank operators create a new image by operating on the relative position, or rank, of the pixel in its neighborhood. Some pixels are the buck privates and others are four-star generals. These processes include the minimum, maximum, median, rank-order, and multiplicative rank operators.

#### 15.1.1 Minimum and Maximum

Minimum and maximum are the simplest non-linear operators. The former scans each pixel in an image and replaces it with the lowest ranking value in its neigh-

borhood. Maximum does the same, except that the replacing value is the highest ranking pixel in the neighborhood.

A computer pseudocode for the minimum procedure looks like this:

```

FOR x = 1 to xmax - radius
    FOR y = 1 to ymax - radius
        min = old(x,y)
        FOR i = -radius TO radius
            FOR j = -radius TO radius
                IF old(x+i,y+j) < min THEN
                    min = old(x+i,y+j)
                END IF
            NEXT j
        NEXT i
        new(x,y) = min
    NEXT y
NEXT x

```

where `radius` defines the size of the neighborhood, `old()` is the original image array, and `new()` is the new one. The parameters `xradius` and `yradius` are often set to 1, for a neighborhood consisting of one central pixel and eight neighbors.

The maximum procedure is the same except for two lines:

```

IF old(x+i,y+j) > max THEN
    max = old(x+i,y+j).

```

By themselves, the minimum and maximum operators are not particularly useful for astronomical images, but they are basic tools that should be in every image processing toolbox.

### 15.1.2 The Median Operator

The median in a set of numbers has an equal number of numbers less than itself and greater than itself. The median operator creates a new image in which each new pixel is the median of the old pixel's neighborhood. It differs from averaging the pixel values in the neighborhood because the numerical values do not matter—only the number of pixels of lower and higher rank (with lower and higher pixel value).

Computing the median operator is simple, but because the pixels in the neighborhood must be sorted into order, the procedure tends to be slow:

```

FOR x = 1 to xmax - radius
    FOR y = 1 to ymax - radius
        k = 0
        FOR i = -radius TO radius
            FOR j = -radius TO radius

```

## Chapter 15: Non-Linear Operators



Figure 15.2 A 15x15 median operator (radius = 7) has removed all of the star and fine detail from the original image, leaving only the large-scale structure. The median operator creates a new image from the pixel value at the middle of the ranked neighborhood—ignoring extreme low- and high-value pixels.

```
k = k + 1
median(k) = old(x+1,y+j)
NEXT j
NEXT i
SORT median()
new(x,y) = median(0.5*(k+1))
NEXT y
NEXT x
```

where `median()` is an array containing  $(2*xradius+1) * (2*yradius+1)$  elements, and the `SORT` function sorts the numbers in the `median()` array into descending order. When the radius is 1, the `median()` array contains nine elements, and the fifth element contains the median of the neighborhood (four elements above it and four below it).

The median operator is an exceptionally useful tool for removing star images and small features from images. Suppose that you have recently discovered a new comet, but you did not obtain a flat field, and the image shows severe vignetting. The median operator turns out to be very handy because you can make a “map” of the vignetting by getting rid of the comet and stars. Here’s how: Make a copy of the image, and then apply the median operator with a radius at least half the size

of the comet image. If the first pass does not remove all of the comet and star images, apply it again. After several passes, the median operator will have entirely removed both the star images and the comet, leaving only the sky background and its vignetting.

Next, measure the average value of the sky background in the center of the image. Add this amount to your original comet image, and then subtract the median filtered image. The stars and comet will be unchanged, but the vignetting will be gone.

### 15.1.3 The Rank-Order Operator

The rank-order process employs a logic that appears similar to the minimum, maximum, and median operators; but, because of a crucial difference, it produces radically different results. In this process, the new pixel value is computed from the rank of the old pixel relative to its neighbors.

Conceptually, the rank-process operator sorts the pixels in a neighborhood into ascending order. The new pixel value is then computed from the relative position, or rank order, of the old pixel within its neighborhood. For example, a neighborhood with a radius of 10 pixels is 21 x 21 pixels, and contains a total of 441 pixels. If the pixel is the 302nd from the lowest in the rank order, the new pixel value will be 302/441 of the maximum possible pixel value for the image. In a 16-bit image with pixel values from 0 to 65535, the 302nd out of 441 pixels will receive a value of 44879.

Here is a procedure for generating rank-order processed images:

```

FOR x = 1 TO xmax - radius
    FOR y = 1 TO ymax - radius
        under = 0
        equal = 0
        total = 0
        FOR i = -radius TO radius
            FOR j = -radius TO radius
                total = total + 1
                IF old(x+i,y+j) < old(x,y) then
                    under = under + 1
                END IF
                IF old(x+i,y+j) = old(x,y) then
                    equal = equal + 1
                END IF
            NEXT j
        NEXT i
        rank = (under + equal / 2) / total
        new(x,y) = rank * pvmax
    NEXT y
NEXT x

```

## Chapter 15: Non-Linear Operators

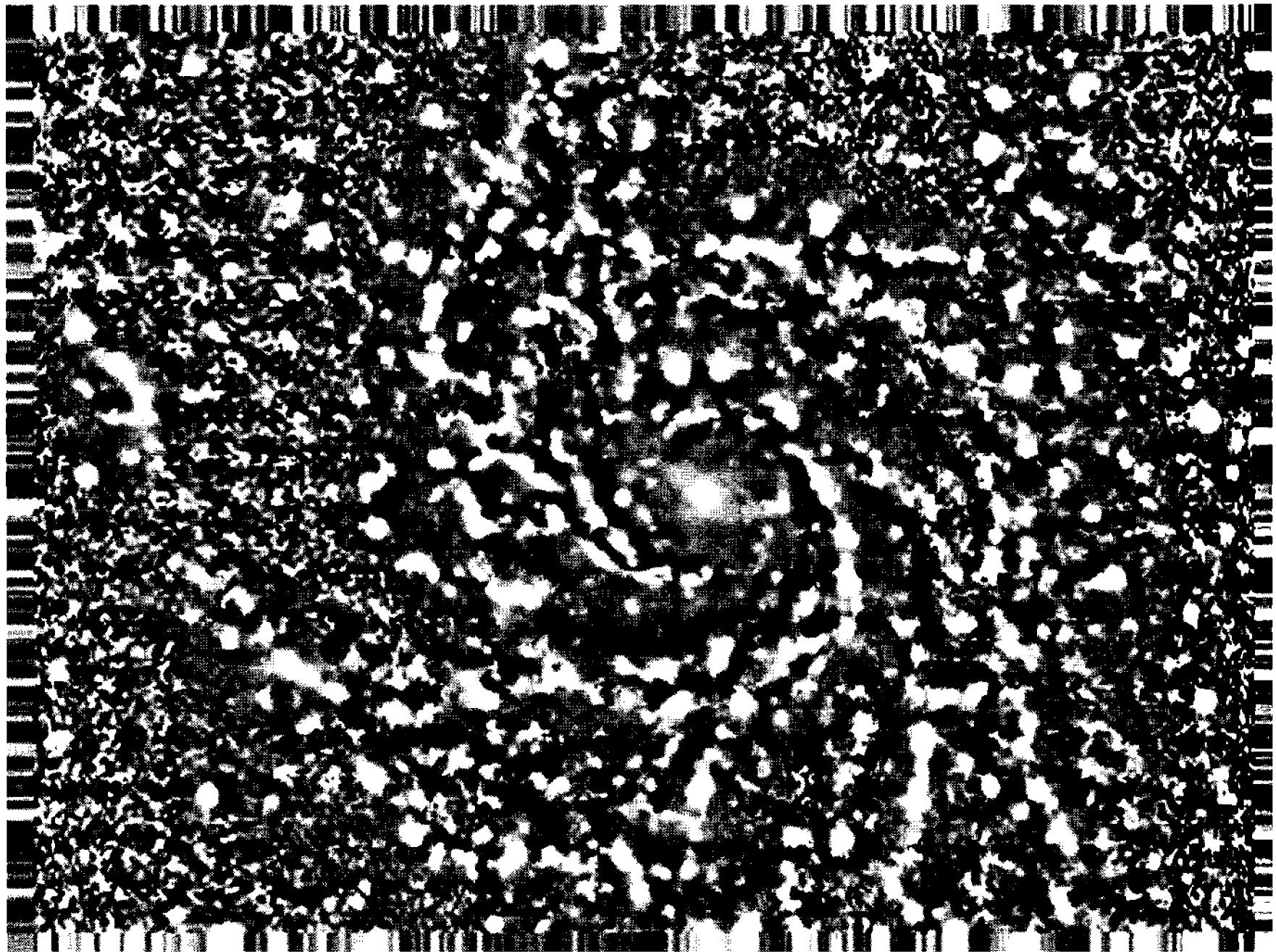


Figure 15.3 The rank-order generates a new image in which pixel values are proportional to the pixel's ranking within its neighborhood. Large-scale structure is entirely ignored. Rank-order images are informative but ugly. This image was made using a radius of 7 pixels ( $15 \times 15$  neighborhood).

This operator has several interesting properties that are extremely useful in astronomy. If the neighborhood contains only a small range of values, then a pixel only slightly brighter than its neighbors lies near the top ranking and becomes white, while a pixel that is only slightly below its neighbors becomes dark gray. If the neighborhood contains a full range of pixel values, the new pixel will probably receive a value not far from its original one.

Consider the results of applying the rank-order process to: a uniform region, regions with small positive variations, regions with small negative variations, and a region with large variations:

- Uniform image: All pixels have the same value. After the inner loop runs through the  $i$  and  $j$  loops,  $\text{under}$  will be 0 and  $\text{equal}$  will equal  $\text{total}$ . The rank will become 0.5, and the new pixel will be a 50% gray shade. In fact, any uniform area becomes 50% gray, whether it is initially black, white, or some shade of gray.
- Small positive variation: The pixels have the same value except the current pixel, which is just one ADU greater than the neighborhood. After the inner loop runs through the  $i$  and  $j$  loops, the value of  $\text{under}$  will equal  $\text{total}$  and  $\text{equal}$  will be 0.

The rank will become 1.0, so the new pixel will become `pvmax` and be displayed as white. The slightest departure above uniformity forces a pixel to pure white, regardless of the initial value of the region.

- Small negative variation: All pixels have the same value except the current pixel, whose value is one ADU less than its neighbors. After running through the `i` and `j` loops, the value of `under` is `total - 1` and `equal` is 0. The rank becomes 0.0, and the new pixel value is 0, or pure black. Even the slightest drop below uniformity and the pixel becomes black, regardless of the initial value of region.
- Wide variation: The pixels have a wide range of distribution. After counting pixels, the value of `under` depends on where in the scale of neighboring pixel values the current pixel falls. The `equal` count is most probably zero or a small number. Rank might have any value from 0.0 to 1.0, depending on the values of the current pixel and its neighborhood. The new pixel is some shade of gray.

The rank-order process is exceedingly sensitive to small local differences. Simultaneously, it is quite insensitive to large local differences. The original pixel values in the image mean very little, and the values in the new image will probably range all the way from white to black. In a bland image, the contrast may be stretched a thousand times; but in an image with a lot of tonal variation, the new one might look almost the same as the original.

Images created by the rank-order process depend strongly on the size of the local neighborhood. If the radius is smaller than the characteristic size of the image detail, the rank-order operator does not detect image structure and does little but enhance noise in the image. A rule-of-thumb for good results using this process is to select a radius at least five times the half-width half-maximum of the star images, and preferably greater. Although computationally demanding, large radii (20 pixels or more) enhance objects such as faint galaxies very nicely.

An image created by the rank-order process contains only as many gray levels as there are pixels in the neighborhood. Within a neighborhood of `radius = 1`, the output image will have only 9 gray levels; with `radius = 2`, it will have 25. For astronomical images, the radius should not only be large enough to include a reasonable amount of variation, but it should also be large enough to produce a few hundred gray levels.

#### 15.1.4 The Multiplicative Rank Processes

The rank-order operator is so potent that it can be difficult to use. However, blending the rank-order with the original image tames the process and makes it a useful tool for astronomical image processing.

Recall that, in the rank-order process, the new pixel value is computed from:

## Chapter 15: Non-Linear Operators

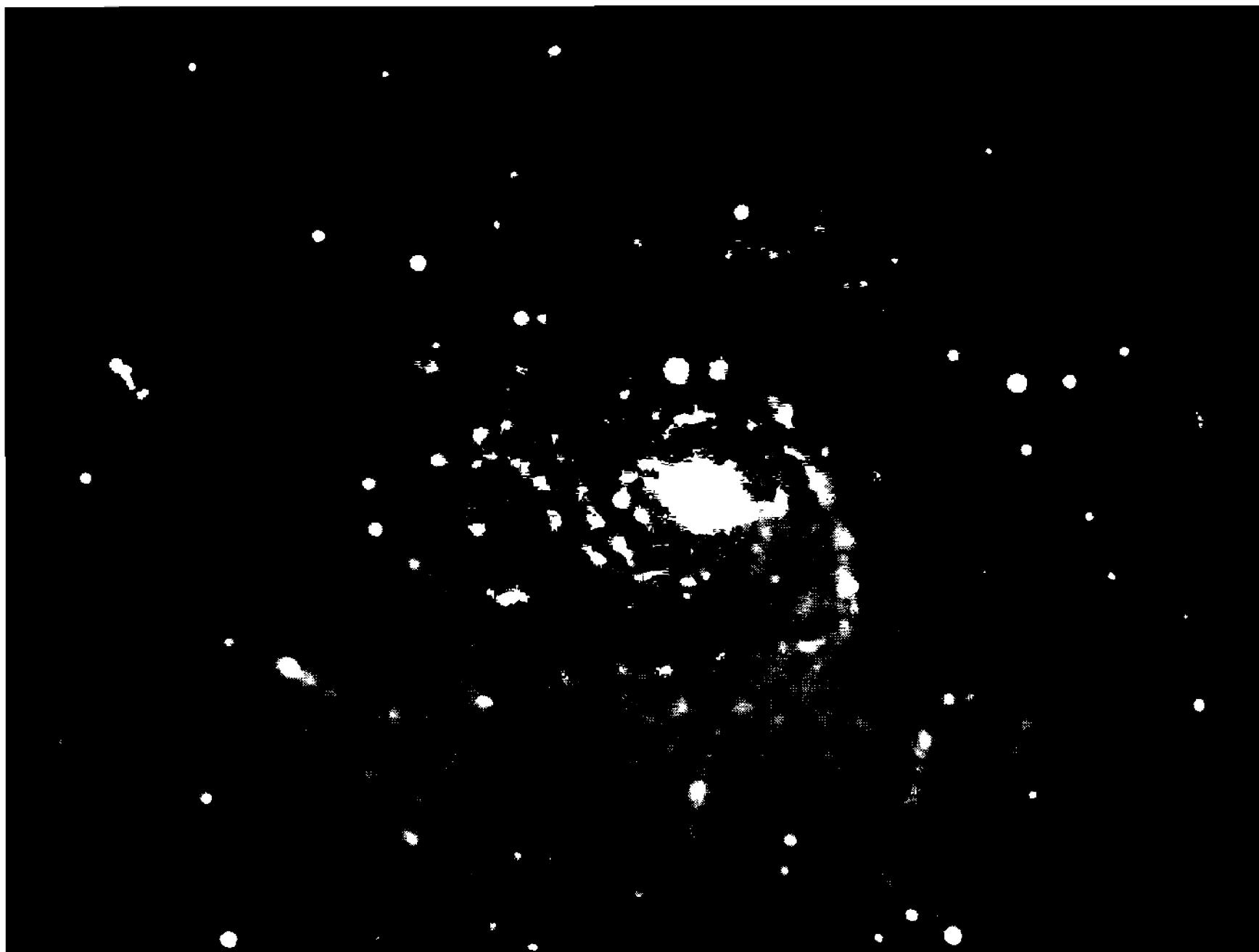


Figure 15.4 Like the rank-order process, the multiplicative rank process enhances fine detail—but the multiplicative version combines the detail enhancement with the large-scale brightness structure of the image. In this image, processing has enhanced the bright arm ridges and traced out the dust lanes.

```
new(x,y) = rank * pvmax
```

where `rank` is the pixel's ranking in its neighborhood. Note that the value of `rank` always lies in the range 0.0 to 1.0.

The additive rank process blends the rank pixel value with that of the original image:

```
oldpart = old(x,y) * (1 - mix) / pvmax
newpart = mix * rank
blend = (oldpart + newpart)
new(x,y) = pvmax * blend
```

where `mix` is the blending factor, a parameter in the range 0.0 to 1.0. The variable `oldpart` holds the normalized value of the old pixel, and `newpart` temporarily holds the new pixel value. For the most predictable results, pixel values in the original image are normalized to the maximum value in the old image, `pvmax`, before the old image and the rank value are blended. In the final line, the blend is scaled into the full range of new image pixel values, 0 to `pvmax`.

The additive rank process moderates the potency of the rank-order operator. If the blending is 50%, for example, a slightly higher pixel value in a dark part of the image cannot become pure white, and a few scattered low-value pixels in a sat-

## Section 15.2: Non-Linear Enhancement Operators

urated highlight region can't go totally black. Despite the relative moderation of blending, background skies become noisy from it unless the blending factor is too low to produce the desired enhancement.

The multiplicative rank process brings the rank-order under control. The rank value is multiplied by the original image as follows:

```
oldpixel = (old(x,y) / pvmax)
result = SQRT(oldpixel * blend)
new(x,y) = pvmax * result
```

where `SQRT` is the square-root function. The variable `oldpixel` holds the normalized old pixel value temporarily, and the variable `blend` is the result of an additive rank process as described above.

This formulation takes advantage of certain handy properties of numbers between 0 and 1. If either the normalized image pixel value, `oldpixel`, or the rank value, `rank`, is 0, their product is 0; black areas remain black, and the darkest pixels in each neighborhood become black. If both values are 1, their product is 1, and the pixel is white. Taking the square root of their product maintains image brightness, so that if both values are 0.5, the new pixel becomes a 50% gray shade.

The multiplicative rank process enhances small, low-contrast image features without grossly distorting the gray-scale of the image. The additive blending factor controls the strength of the rank-order process, and the multiplicative component keeps the tonal scale of the image under control.

- **Tip:** With **AIP4Win**, multiplicative rank processing is the default operation carried out by the rank-process tool. Multiplicative rank processing is an exceptionally powerful tool for extracting low-contrast detail from images of comets, nebulae, and planets.

## 15.2 Non-Linear Enhancement Operators

The human eye and brain are such powerful image processing tools that it is difficult to find operators that perform functions that a human being cannot do more easily. However, two non-linear operators stand out from the crowd of candidates: the extreme value operator, and local adaptive sharpening.

The extreme value operator is important because it removes subjective factors from finding regions of like pixel value in images. Humans trace “features” with little regard to the actual brightness, relying on very capable edge-detection processors in the eye and brain. As a consequence, it is difficult for a human to trace a boundary based solely on intensity. The extreme value operator creates sharp, objectively defined boundaries between image features.

Local adaptive sharpening is a detail-enhancing operator that applies a stronger enhancement to low-contrast parts of an image than it does to high-contrast parts. This operator complements the human visual system, which tends to ignore detail in low-contrast areas. After local adaptive sharpening, images may not be beautiful, but the observer is rewarded with clear detail over its entirety.

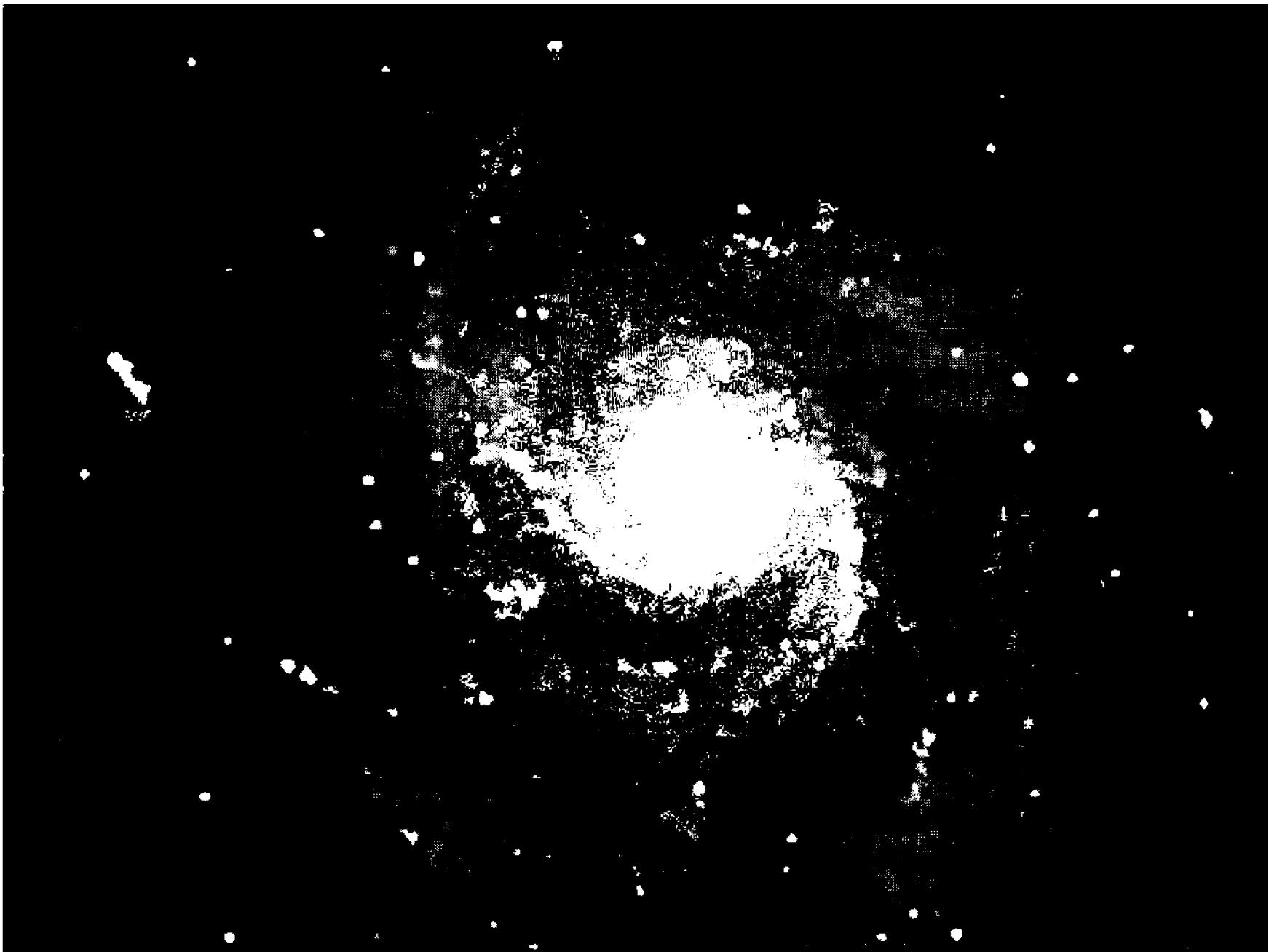


Figure 15.5 By amalgamating regions of similar pixel value, the extreme value operator sharpens the appearance of an image and creates distinct, well-defined steps in brightness that help delineate the “soft” features that define the spiral arms of this one. In this image, the neighborhood size is  $5 \times 5$  pixels.

### 15.2.1 The Extreme Value Operator

This operator amalgamates regions of similar pixel value, making patterns such as planetary cloud belts more readily visible. Its operation is remarkably simple: within the neighborhood, it first finds the lowest and highest pixel value. It then determines whether the current pixel is closer to the minimum or to the maximum value, and assigns whichever is closer to the value of the old pixel to the new pixel.

Here is pseudocode for the extreme value operator:

```
FOR x = 1 + xradius TO xmax - radius
    FOR y = 1 + yradius TO ymax - radius
        min = pmax
        max = 0
        FOR i = x - radius TO x + radius
            FOR j = y - radius TO y + radius
                IF min > old(i,j) THEN min = old(i,j)
                IF max < old(i,j) THEN max = old(i,j)
            NEXT j
        NEXT i
        IF old(x,y) - min <= max - old(x,y) THEN
            new(x,y) = min
```

## Section 15.2: Non-Linear Enhancement Operators

```
ELSE
    new(x,y) = max
END IF
NEXT y
NEXT x
```

In the procedure, `min` is the neighborhood minimum and `max` is the neighborhood maximum. Initially, they are set to the highest and lowest pixel values in the image, and then tested against each pixel in the neighborhood. Finally, we compute the difference between the current pixel and the minimum, and assign the closer value to a pixel in the new image.

What does this procedure accomplish? Consider what happens in a uniform region, a region with small variations, and a region with large variations.

- Uniform region: Here the current pixel, the minimum, and the maximum value are the same. The new image is exactly the same as the old one.
- Region with small variations: In each neighborhood, the extreme value operator assigns the current pixel the value of the lowest or highest neighborhood pixel. In a nearly uniform region with just a few low- or high-value pixels, the current pixel is forced to the local minimum or maximum, which is likely to represent low-contrast detail in the image. Any pixel that is on or near a border between two regions with a different average brightness is forced to join (that is, take on the characteristic pixel value of) one region or the other.
- Region with large variations: Similar to regions with small variations: local highs and lows are accentuated because the current pixel is forced to become part of a feature or part of the local background. Transitions between areas of different pixel value become abrupt.

The size of the neighborhood should be two to three times greater than the characteristic size of the image detail. For Nyquist-sampled images of the Moon and planets, a radius of 2 or 3 (i.e., 5 x 5 and 7 x 7 neighborhoods) can produce a dramatic edge enhancement.

The extreme value operator is useful for clarifying and delineating detail in planetary and lunar images, and produces interesting effects on those of faint deep-sky objects. It sharpens transitions and accentuates edges and boundaries. Images processed using it can reduce subjective factors in the measurement of sizes, shapes, and areas, such as the latitude extent of the Martian polar caps, or the size and orientation of elliptical galaxies.

- **Tip:** *In addition to the minimum/maximum extreme value operator, AIP4Win includes a three-level version in which the current pixel is forced to the minimum, the median, or the maximum of the*

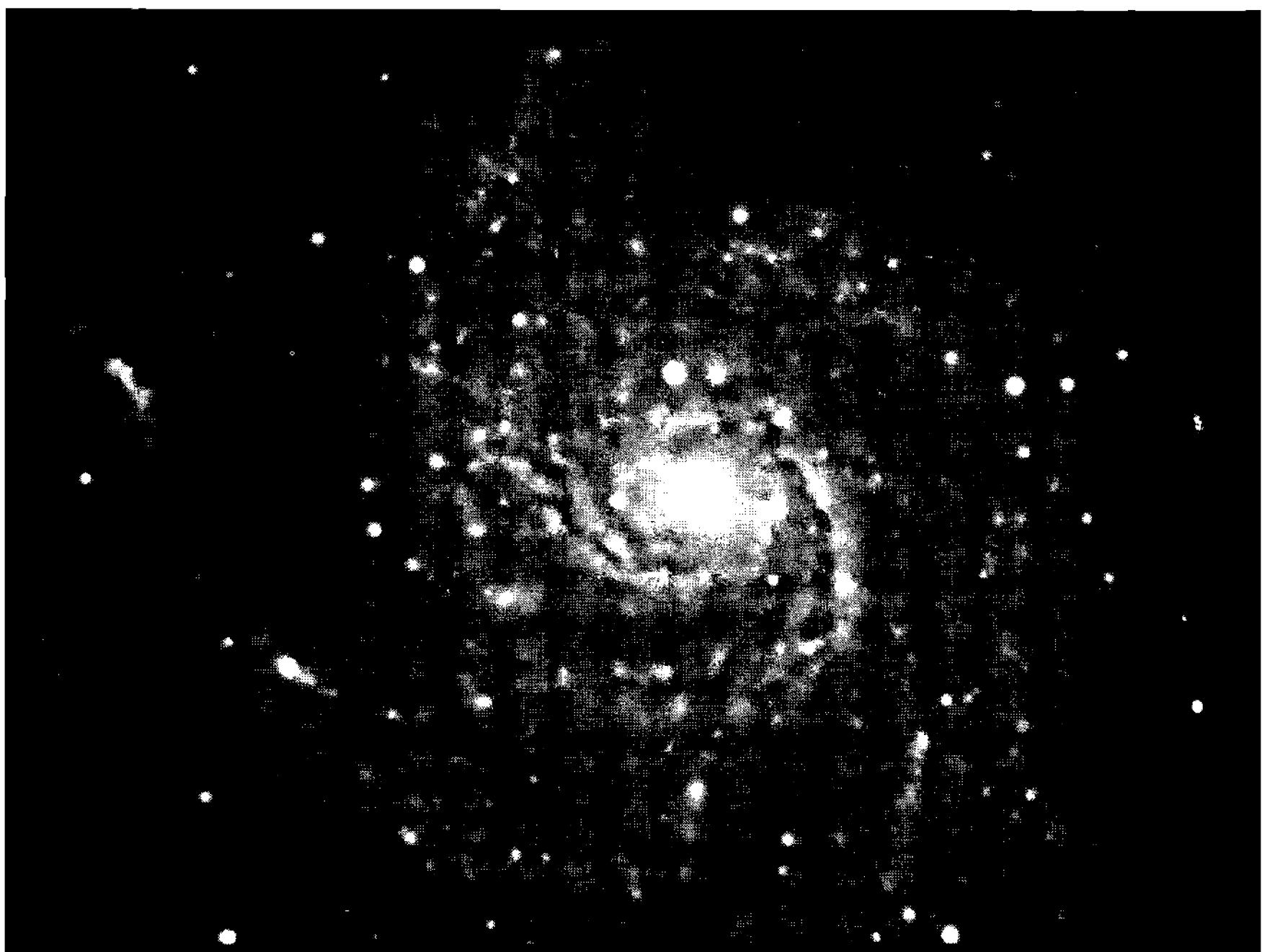


Figure 15.6 Local adaptive sharpening accentuates subtle small-scale features without looking unduly artificial. By applying an unsharp mask under the control of a routine that senses the contrast level in the original image, it increases sharpness where the existing contrast is too low to see easily.

*neighborhood. This variant does not sharpen as strongly, but the resulting images look smoother.*

### 15.2.2 Local Adaptive Sharpening

Local adaptive sharpening is an unsharp mask for which contrast is controlled by neighborhood statistics. Although the algorithm does not contain a conditional statement, it is strongly nonlinear because it alters the local statistics of the image.

In linear unsharp masking, the mask is formed as an average of the neighborhood. The difference between the current pixel and the mask is then amplified by a constant contrast factor and added back to the current pixel value. In this way deviations in pixel value, presumably representing image detail, are enhanced.

In local adaptive sharpening, however, rather than using a constant contrast factor, we compute the ratio of the average pixel value of the neighborhood to the standard deviation of the neighborhood. Where local image contrast is high, the resulting contrast factor is low. In regions where image contrast is low, the standard deviation is small, and the contrast factor becomes large.

The unsharp mask is formed by summing pixel values in the neighborhood in the variable `pvmean`, and then dividing by the number of pixels, `numpixels`, to obtain the mean pixel value, `pvmean`. The difference between the current pixel and the mean pixel value is stored in the variable `diff`.

## Section 15.2: Non-Linear Enhancement Operators

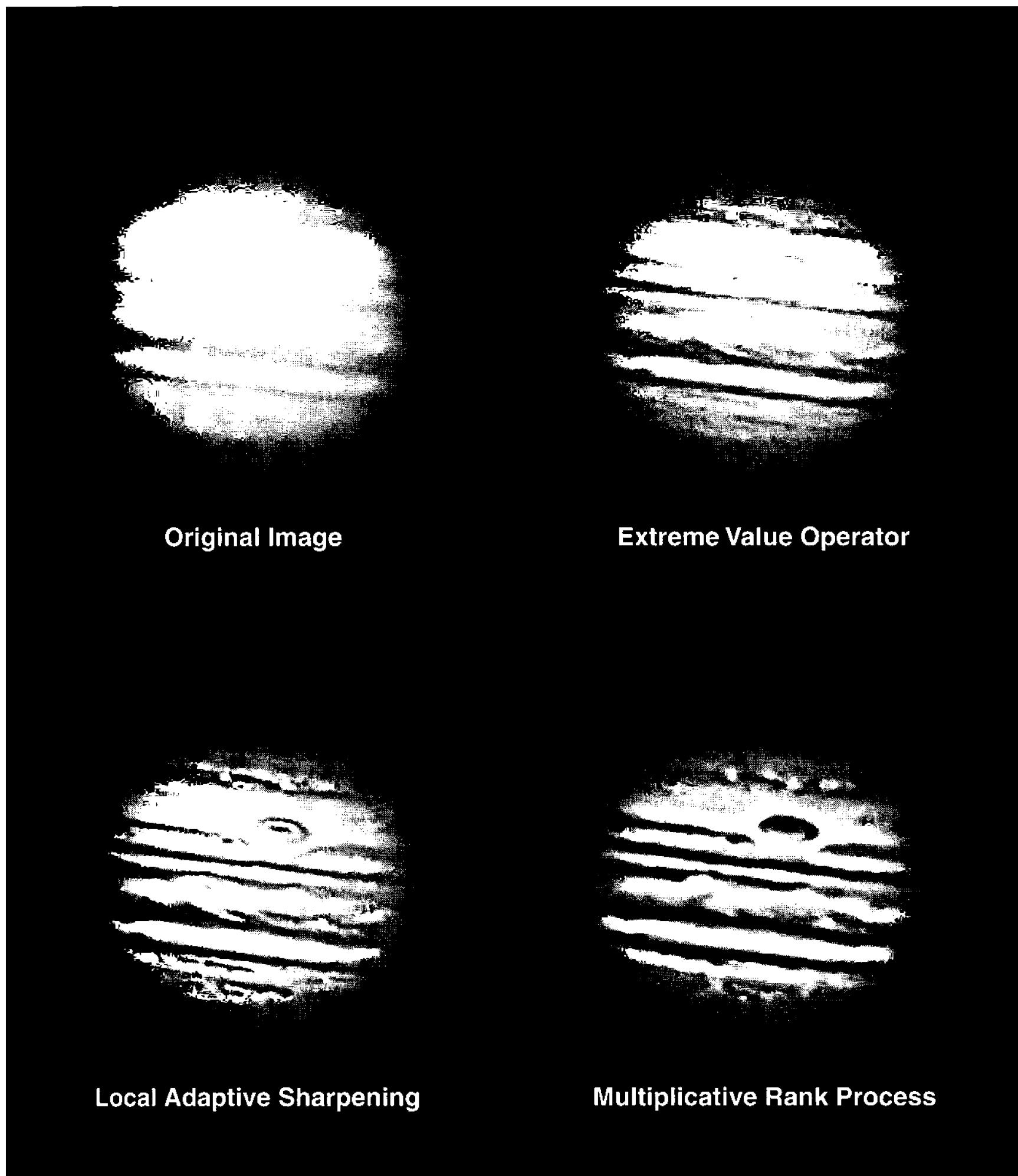


Figure 15.7 Although belts and zones are present in the original, they are soft and ill-defined. The extreme value operator adds definition to cloud and zone edges. Local adaptive sharpening does wonders to enhance the fine Jovian detail. The multiplicative rank process is overly harsh. Jupiter image by Donald Parker.

The standard deviation is:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_n (P_i - \bar{P})^2} \quad (\text{Equ. 15.1})$$

where  $n$  is the number of pixels in the neighborhood, and  $\bar{P}$  is the mean pixel value of the neighborhood. In the procedure below, the square of the deviation from the mean accumulates in the variable `sig`. Once the sum is computed, it is divided

## Chapter 15: Non-Linear Operators

by  $n - 1$  and the square root is extracted. The mask value, stored in diff, is then multiplied by the contrast factor, pvmean/sig, and added to the original pixel value.

Here is the complete algorithm:

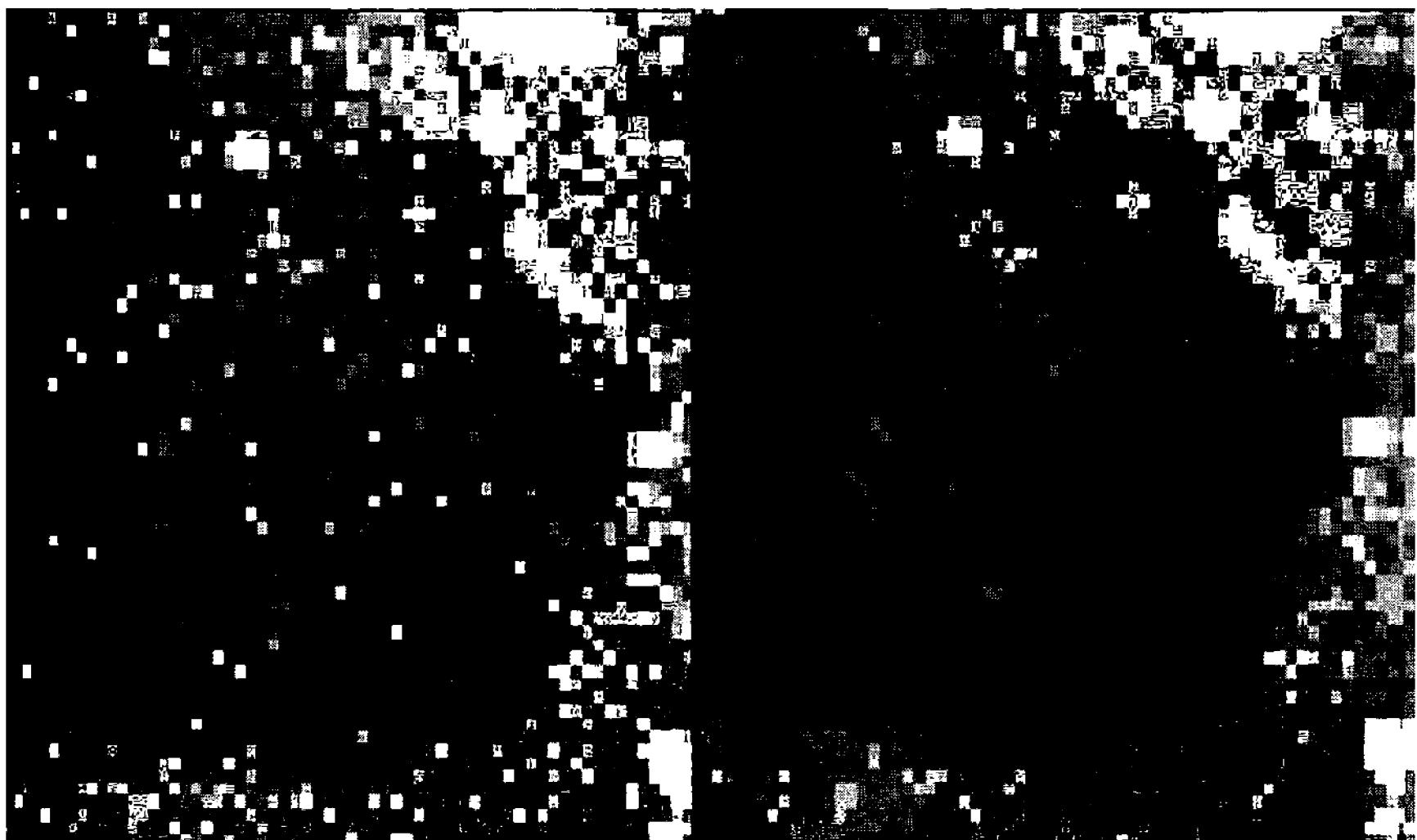
```
FOR x = xradius TO xmax - radius
    FOR y = yradius To ymax - radius
        numpixels = 0
        pvmean = 0
        FOR i = x - radius TO x + radius
            FOR j = y - radius To y + radius
                numpixels = numpixels + 1
                pvmean = pvmean + old(i,j)
            NEXT j
        NEXT i
        pvmean = pvmean / numpixels
        diff = old(x,y) - pvmean
        sig = 0
        FOR i = x - radius TO x + radius
            FOR j = y - radius TO y + radius
                sig = sig + (old(i,j) - pvmean) ^ 2
            NEXT j
        NEXT i
        sig = SQRT(sig / (numpixels - 1))
        diff = diff * pvmean / sig
        new(x,y) = old(x,y) + diff
    NEXT y
NEXT x
```

Local adaptive sharpening operates very strongly on images with low contrast—such as planetary disks—but has little effect on those that are already full of rich detail, such as the lunar surface near the terminator. It works well on images with high signal-to-noise ratios, but on noisy images it can enhance noise in uniform areas. Users should remain alert to this tendency, especially when relying on single images.

- **Tip:** In **AIP4Win**, you can select a separate contrast factor that further amplifies the contrast factor computed from the image statistics, giving you control over the total contrast enhancement applied to the image.

### 15.3 Noise Filters

In amateur CCD images, there are two kinds of noise: the random noise of photon and electron statistics, and the impulse noise of cosmic rays and hot pixels. Pixel values in random noise cling to some central value, whereas pixel values afflicted



**Figure 15.8** For whatever reason, you are sometimes stuck with a very noisy image. On the left is a highly enlarged image with hot pixels, on the right the same image after the application of a noise filter. The highly deviant hot pixels have been replaced with pixels that “belong” in the local neighborhood.

by impulse noise appear as large departures from a central value. The purpose of a noise filter is to analyze the statistical properties of the neighborhood around each pixel, determine whether the pixel is impulse noise, and if it is, replace it.

All noise filters that operate in the spatial realm rely on the fact that the numerical values of adjacent pixels in a properly sampled image are strongly correlated. However, if an image is severely undersampled, stars in it will appear as one pixel that deviates strongly from the sky background; and the noise filter will replace those pixels because, statistically, such stars *are* noise.

The rank median operator is sometimes used to remove noise, especially impulse noise, from faulty digital transmissions. Unfortunately, the median filter operates the same for all pixels, so that it often removes detail as well as noise. The standard noise filter described here is a median filter that activates only when a pixel exceeds the range of values characteristic of the image.

The following procedure replaces any pixel that deviates by a specified amount that is greater than the next-most-deviant pixel in the neighborhood. The procedure that follows removes noise from 3x3 neighborhoods:

```

FOR y = 1 to ymax - 1
    FOR x = 1 to xmax - 1
        k = 0
        FOR i = -1 TO 1
            FOR j = -1 TO 1
                k = k + 1
                neighbor(k) = old(x+i,y+j)
        NEXT j
    
```

## Chapter 15: Non-Linear Operators

```
    NEXT i
    SORT neighbor()
    new(x,y) = old(x,y)
    sigma = (neighbor(8) - neighbor(2)) / 2
    IF old(x,y) > neighbor(5) + sigs * sigma THEN
        new(x,y) = neighbor(5)
    END IF
    IF old(x,y) < neighbor(5) - sigs * sigma THEN
        new(x,y) = neighbor(5)
    END IF
    NEXT x
NEXT y
```

where `SORT` is a function that sorts the array `neighbor()` into ascending order, `sigma` is the standard deviation, and `sigs` is a user-set parameter specifying the number of standard deviations allowed.

The loops in `i` and `j` load neighborhood pixels into the array `neighbor` starting with `neighbor(1)` and ending with `neighbor(9)`. After sorting, `neighbor(9)` contains the pixel with the highest value in the neighborhood.

Rather than simply replacing the original pixel if it is an extreme, we want its statistical “fit” to determine whether we should keep it or replace it. In a normal distribution, 68% of the values lie within one standard deviation of the mean; in a sample of nine pixels, the closest we can come to this is to treat the inner seven pixels as falling within one standard deviation of the mean. This means that half the difference between `neighbor(8)` and `neighbor(2)` is roughly one standard deviation.

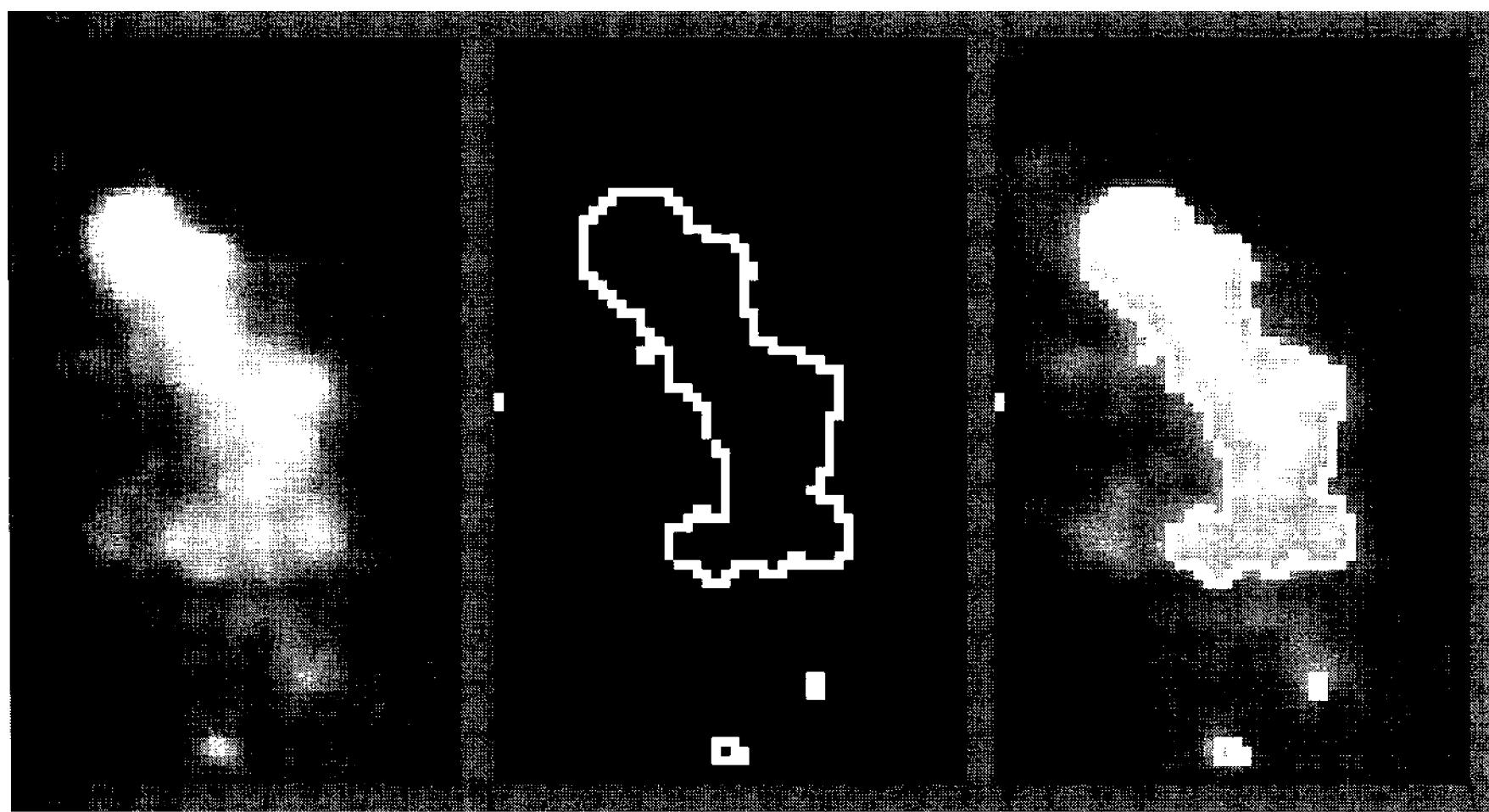
The parameter `sigs` is defined by the user as the allowable deviation from the median in units of `sigma`; typically it is given a value from 1.5 to 2. If the current pixel exceeds the allowable limits, the routine replaces the current pixel with the median of the neighborhood, `neighbor(5)`. The routine replaces the original pixel, if and only if, the original pixel has an extreme value relative to the neighborhood—otherwise it does nothing.

- **Tip:** *The noise filter in **AIP4Win** allows you to specify the trigger level in the noise filter, specifying that the deviant pixel must exceed the second-most-extreme by a factor that you have set. This gives better control over the strength of the filtering.*

In Chapter 18, we describe a more sophisticated class of noise filters that use wavelet technology to identify and filter out noise. Wavelet filters employ a hybrid approach that combines frequency analysis with spatial filtering.

## 15.4 Morphological Operators

Morphology deals with form and shape; in image processing, morphological operators are tools that aid the observer in defining, extracting, and manipulating the



**Figure 15.9** Isophote contour lines encircle a region having pixel values greater than the isophote pixel value. This highly enlarged section of the M101 image shows a knot in the spiral arm, an isophote contour line drawn around that knot, and the same isophote line superimposed on the image.

forms and shapes of objects in images. By themselves, the array of pixel values in a bitmapped image has nothing to say about objects. The task of the morphological operator is to associate groups of pixels—based on their pixel values—to reveal objects as isolated groups of pixels. In the process of extracting form and shape, morphological operators discard information about brightness in return for highlighting the shapes and forms of objects.

Morphological operators isolate contour lines of constant brightness, locate edges, locate shapes, shrink lines and shapes, broaden lines and shapes, and reduce fuzzy lines and blobs to one-pixel-wide skeletons. By themselves, these operators do not produce useful results. But used in a logical sequence, each process feeding into the next toward a specific goal, they are powerful analytic tools.

### 15.4.1 Isophote Lines

Just as the isobars on the weather map connect regions of equal barometric reading, isophote lines connect regions of equal brightness on an image. Isophotes show how the brightness in an image is really distributed, which is quite often surprising. Where the eye sees spiral arms in a galaxy, isophotes reveal radially declining contours of brightness within the spiral arms as relatively minor departures from the overall pattern of radial decline.

What exactly are isophote lines? You already know that a geographic elevation contour is the boundary between lower land and higher land. An isophote defines the boundary between a region with higher pixel values and another with lower values. To be an isophote pixel, a pixel must lie between one that has a higher pixel value than the isophote and another pixel that has a lower value than the

## Chapter 15: Non-Linear Operators

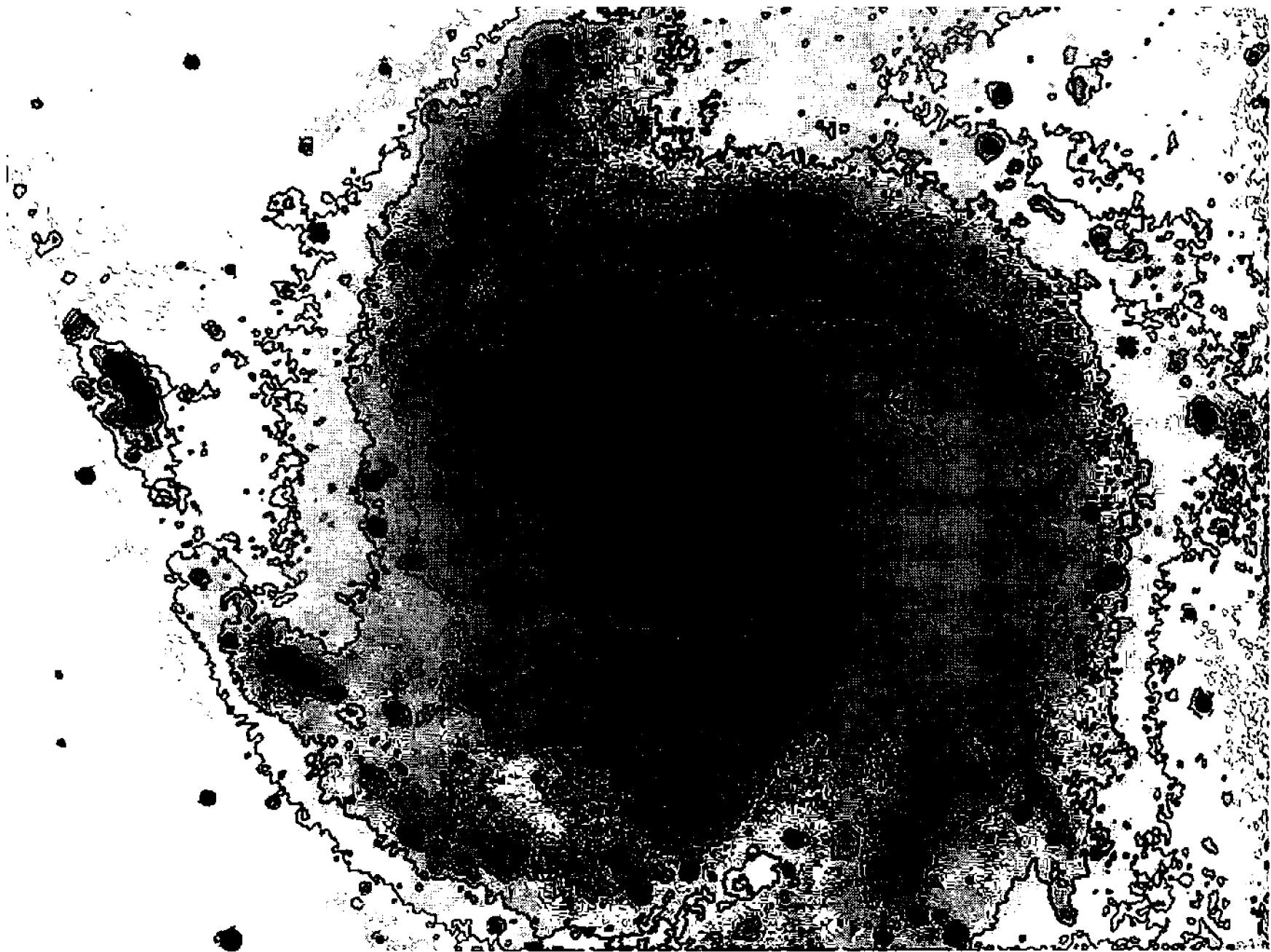


Figure 15.10 While structural features—the spiral arms—lead the eye, the isophotes follow distorted circles of surface brightness. For most observers, it comes as no surprise that galactic arms are very subtle features. The inverted gray scale makes it easier to see both the underlying galaxy and the contours.

isophote. There is just one exception: a pixel can also be an isophote by having a value equal to it.

From the definition, it is easy to craft an algorithm that draws contour lines by making pixels that do not lie on an isophote black, and by making those that do white. In the procedure below, the new pixel is first set to black, and then it is tested to determine whether it satisfies the conditions for being on an isophote.

```
FOR y = 1 TO ymax
    FOR x = 1 TO xmax
        new(x,y) = black
        IF old(x,y) = iso THEN new(x,y) = white
        IF old(x,y) < iso THEN
            IF old(x-1,y) > iso THEN new(x,y) = white
            IF old(x,y-1) > iso THEN new(x,y) = white
            IF old(x-1,y-1) > iso THEN new(x,y) = white
        END IF
        IF old(x,y) > iso THEN
            IF old(x-1,y) < iso THEN new(x,y) = white
            IF old(x,y-1) < iso THEN new(x,y) = white
```

## Section 15.4: Morphological Operators

```

    IF old(x-1,y-1) < iso THEN new(x,y) = white
    END IF
NEXT x
NEXT y

```

where `iso` is the pixel value for the isophote line, `black` is a pixel value that displays as black, and `white` is a pixel value that appears white on the screen.

The procedure begins by setting `new(x,y)` to `black`. The current pixel, `old(x,y)`, can be equal to, less than, or greater than the isophote. If the current pixel equals the isophote value, `new(x,y)` is set to `white`. If the current pixel is less than the isophote, and if an adjacent pixel must be greater than the isophote, `new(x,y)` is set to `white`. Finally, if the current pixel is greater than the isophote and an adjacent pixel is less than the isophote, `new(x,y)` is set to `white`. The same logic could be handled equally well using CASE statements.

Since it is better to draw the isophotes *between* the original pixels rather than *through* them, to draw the best contours, resample images on which you want isophotes drawn by 200% or more. Resampling places interpolated pixels between the original ones, and provides a clean, good-looking contour line.

- **Tip:** *AIP4Win provides the ability to draw a single contour at a fixed pixel value, as well as the ability to draw multiple isophotes that divide an image into regions having equal area.*

### 15.4.2 Frei and Chen Operators

The contour line operator draws boundary lines at a constant pixel value, but it is often necessary to detect boundaries marked by sudden changes in brightness rather than a constant brightness. Although Kirsch, Sobel, and Prewitt operators act as edge detectors, the suite of operators described by Frei and Chen are the best small-kernel operators for detecting edges, lines, points, and ripple.

There are nine Frei and Chen kernels. They are designed to cover the range of small-neighborhood morphologies, which includes edges (high-value pixels on one side and low-value pixels on the other), ripple (intermixed high- and low-value pixels), line (aligned high-value pixels against low-value pixels), and points (isolated high-value pixels). By combining the output of these kernels, you can build a variety of edge- and boundary-detecting operators.

The first pair of Frei and Chen operators, FCO1 and FCO2, detects edges:

$$\text{FCO1} = \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix}, \text{ and } \text{FCO2} = \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}. \quad (\text{Equ. 15.2})$$

Compute FCO1 and FCO2 as follows:

$$\begin{aligned} \text{FCO1} &= \text{old}(x-1,y-1) \\ &+ 1.414 * \text{old}(x,y-1) \end{aligned}$$

## Chapter 15: Non-Linear Operators

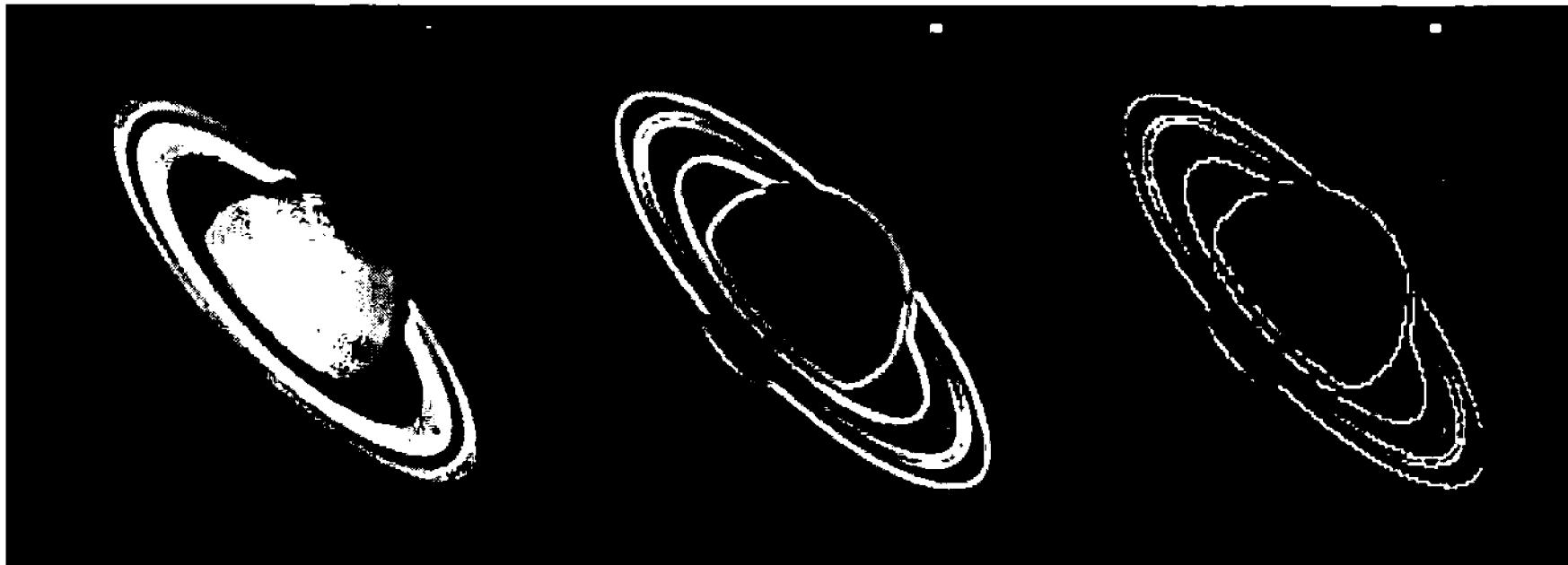


Figure 15.11 Morphological operators extract forms and shapes from images. The original image of Saturn (left) was processed using the Frei and Chen edge operator, to make the middle image. The final step was to skeletonize the edges into lines one pixel wide. Saturn image by Donald Parker.

```

+ old(x+1,y-1)
- old(x-1,y+1)
- 1.414 * old(x,y+1)
- old(x+1,y+1)

FCO2 = old(x-1,y-1)
- old(x+1,y-1)
+ 1.414 * old(x-1,y)
- 1.414 * old(x+1,y)
+ old(x-1,y+1)
- old(x+1,y+1)

```

The FCO1 and FCO2 kernels function as horizontal and vertical edge detectors, respectively. The  $\sqrt{2}$  coefficients compensate for the separation between the central pixel and corner pixel, so that the kernel's response to edges that are not horizontal or vertical is the same as it is to aligned edges.

The FCO3 and FCO4 operators are what Frei and Chen call “ripple” detectors, since they are insensitive to edges and produce the largest output with irregular patterns, or ripples:

$$FCO3 = \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix}, \text{ and } FCO4 = \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix}. \quad (\text{Equ. 15.3})$$

FCO5 and FCO6 detect linear groups of pixels that stand out from the local background. The crossed positive and negative coefficients produce small responses at edges and large responses at lines:

## Section 15.4: Morphological Operators

$$FCO5 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & \mathbf{0} & -1 \\ 0 & 1 & 0 \end{bmatrix}, \text{ and } FCO6 = \begin{bmatrix} -1 & 0 & 1 \\ 0 & \mathbf{0} & 0 \\ 1 & 0 & -1 \end{bmatrix}. \quad (\text{Equ. 15.4})$$

FCO7 and FCO8 are point detectors, responding most strongly to isolated pixels and small clusters of pixels:

$$FCO7 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & \mathbf{4} & -2 \\ 1 & -2 & 1 \end{bmatrix}, \text{ and } FCO8 = \begin{bmatrix} -2 & 1 & -2 \\ 1 & \mathbf{4} & 1 \\ -2 & 1 & -2 \end{bmatrix}. \quad (\text{Equ. 15.5})$$

Finally, FCO9 is the sum or average operator, useful in adjusting the output over a range of image intensity:

$$FCO9 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \mathbf{1} & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (\text{Equ. 15.6})$$

Frei and Chen operators can be combined to make kernels that detect specific morphological patterns in small neighborhoods. The following procedure creates a new() image that highlights edges in an old() image:

```

FOR y = 1 to ymax - 1
    FOR x = 1 to xmax - 1
        [compute FCO1, FCO2, and FCO9 here]
        result = SQRT(FCO1 * FCO1 + FCO2 * FCO2)
        new(x,y) = (result / (FCO9+1))
    NEXT x
NEXT y

```

where `result` is a variable that holds the output of combining the horizontal and vertical edges detectors and `SQRT` is the square root function. Dividing by the average value means that this operator finds edges with little regard to the image brightness.

- **Tip:** *AIP4Win includes the Frei and Chen operators for detecting edges, lines, ripples, and points. In astronomy, the edge detector and the line detector are by far the most useful of the Frei and Chen operators.*

The new image created by the Frei and Chen line operator is a grayscale image, and the boundary lines tend to be two to three pixels wide. To produce lines that are one pixel wide, first threshold the image and then run the skeletonize operator described in the following section.

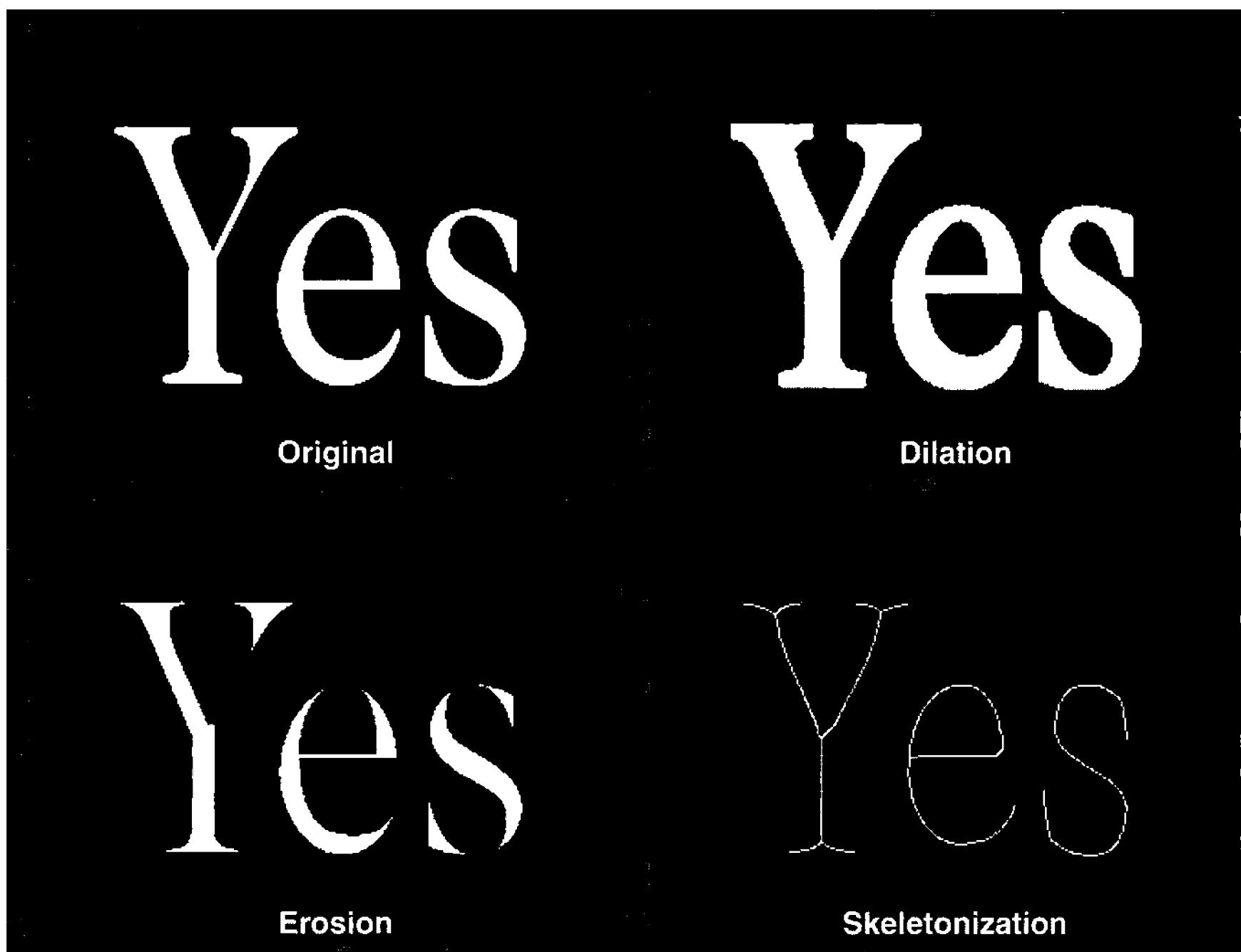


Figure 15.12 Morphological operators perform most reliably on binary images—that is, those containing only two pixel values. Dilation swells the white parts of the original image, whereas erosion shrinks them. Skeletonization reduces the thick strokes of the original to one-pixel-wide lines.

### 15.4.3 The Skeleton Operator

Skeletonizing is a morphological process that is probably most useful for reducing the fat lines created by edge detection operators into thin boundary lines. In fields such as document scanning, the skeleton operator carries out tasks such as cleaning up images of printed letters. In astronomy, it is a useful tool for cleaning up isophote, edge, boundary, and line images created by other operators.

Although skeletonizing can process grayscale images, it works best with binary images—that is, ones that have only two pixel values: black and white. Black is always 0, and white is usually 1, 15, 255, 4095, or 65535, depending on the bit depth of the image data.

Skeletonization works by scanning the image repeatedly setting any pixel that is anything other than the end of a line or a one-pixel-thick section of a line to zero. After some number of scans, the image is entirely black except for “skeleton” lines one pixel thick.

In astronomical image processing, skeletonizing is a handy way to convert the fuzzy images from edge- and boundary-detection operators into thin lines or contours. Because skeletons made from irregular shapes often have many short branches, after skeletonizing, use a skeleton pruning operator to remove unwanted side branches.

- **Tip:** Although **AIP4Win's** skeltonizing function works with grayscale images, it gives more intelligible results with binary ones. Use the threshold function to convert grayscale images into binary images.

#### 15.4.4 Dilation, Erosion, Opening, and Closing Operators

Dilation and erosion are operators best used only with binary images. On a binary image, dilation accretes new high-value pixels whereas erosion eats away high-value pixels. Opening is a dilation followed by an erosion, and closing is an erosion followed by a dilation. Dilation fattens an area, and erosion thins an area. Opening enlarges voids in otherwise solid areas, and closing closes the voids.

In astronomical image processing, these operators can be used to tidy up output from other operators. Edge operators, for example, produce messy-looking results. After thresholding to produce a binary image, you can apply erosion to eliminate small regions and thin out heavy lines.

Erosion is a very simple operator: for each high-value pixel in the image, the erosion operator tallies the neighborhood. If any of the surrounding pixels is a low-value one, the operator sets it to low value. Erosion shrinks regions of high-value pixels and expands regions of low-value ones. After erosion, a line that is two pixels wide will be gone.

Dilation is equally simple: for each pixel in the image, the operator checks for high-value pixels in the neighborhood. If any of its neighbors is a high-value pixel, then the operator sets it to high value. Dilation expands regions of high-value pixels and whittles away at regions containing low-value ones. After dilation, a line one pixel wide is tripled in width.

Because dilation and erosion are complementary—the former building and the latter destroying—their combined operation in opening and closing largely cancel out. The net result of an opening or closing is to round off sharp corners one pixel at a time, and to fill cracks and voids one pixel at a time. After multiple openings, a square group of pixels becomes round; after multiple closings, two touching circles become separate.

### 15.5 The Topographic Operator

This operator treats the pixel values in an astronomical image as elevation in a digital terrain map, to create a shaded relief image. The resulting bas-relief produces a more realistic illusion of a three-dimensional surface than the simple bas-relief kernel described in the previous chapter, because the new pixel value is based on the behavior of diffusely reflecting surfaces. What the topographic operator does is to figure out how the image would look if it were sculpted as a relief in plaster of Paris.

The main loop of the procedure examines each pixel in turn, and determines the elevation difference between the current pixel and an interpolated pixel computed from the solar azimuth angle (described below). The key to the operation is

## Chapter 15: Non-Linear Operators



Figure 15.13 The topographic operator produces a realistic simulation of a plaster relief by using a realistic model to describe how bright each pixel should appear. To produce this image, the artificial Sun shone from a  $30^\circ$  elevation at an azimuth of  $315^\circ$  on a mesh of pixels spaced 1000 units apart.

ILLUM, the illuminance (apparent brightness) of the pixel.

```
FOR y = 1 TO ymax
    FOR x = 1 TO xmax
        ct = old(x,y)
        hn = cos(saz) * (ct - old(x,y-1)) / sep
        hw = sin(saz) * (ct - old(x-1,y)) / sep
        diff = SQRT(hw * hw + hn * hn)
        new(x,y) = ILLUM(diff)
    NEXT x
NEXT y
```

where *saz* is the solar azimuth angle, *sep* is the separation between pixels in the units used for elevation, *hn* is the elevation difference relative to the pixel to the north, *hw* is the elevation difference relative to the pixel to the west, *SQRT* is the square root function, and *ILLUM* is the illuminance function that computes how bright the new pixel should be.

The illuminance model is based on the optical properties of a Lambertian reflector, which in the everyday world is a perfectly diffuse white surface such as plaster of Paris. The apparent brightness, or illuminance, depends on the angle of

## Section 15.5: The Topographic Operator

the illumination and on whether the surface tilts toward or away from the light source. The only thing missing from the model is that steep hills do not cast shadows, but this is beneficial because details are never hidden in shadows.

When there is no difference between adjacent pixels, that is, when  $d = 0$ , the ground tilt,  $e$ , is 0. Thus, the local angle of incidence,  $i$ , equals  $90^\circ - \vartheta_{\text{sun}}$ . The terrain illuminance,  $I$ , becomes:

$$I = \frac{1}{1 + \frac{1}{\cos i}}. \quad (\text{Equ. 15.7})$$

When the elevation difference is positive, the ground tilts toward the light,  $d$  is positive and the ground tilt angle is  $e = \tan(d/s)$ . Light falls on the terrain with an angle of incidence,  $i$ , equal to  $90^\circ - e - \vartheta_{\text{sun}}$ . The terrain illuminance is:

$$I = \frac{1}{1 + \left(\frac{\cos e}{\cos i}\right)}. \quad (\text{Equ. 15.8})$$

When the elevation difference is negative and the ground tilts away from the light, the ground tilt angle becomes negative. The angle of incidence becomes  $i = 90^\circ + e - \vartheta_{\text{sun}}$ . When the ground tilt is greater than the elevation of the Sun; that is, when  $e \geq \vartheta_{\text{sun}}$ , the slope is in shadow, the ground will be dark and the illuminance will be zero:

$$I = 0. \quad (\text{Equ. 15.9})$$

Where  $e < \vartheta_{\text{sun}}$ , that is, where Sunlight does not fall on the ground to illuminate it, the illuminance is:

$$I = \frac{1}{1 + \left(\frac{\cos e}{\cos i}\right)}. \quad (\text{Equ. 15.10})$$

In a computer program, the best strategy is to precompute a table of illuminance for all values of elevation difference. If this is done, then `ILLUM()` is an array containing the precomputed values.

To use the topography operator, the user must specify the pixel separation, the solar azimuth, and the solar elevation. The smaller the pixel separation, the steeper the slopes, and the more dramatic is the resulting image. However, overly steep slopes look unrealistic; so in practice, the pixel separation should be 2 or 3 times the “typical” range of pixel value found in the image.

The solar azimuth controls the apparent direction of illumination. At an azimuth of  $0^\circ$ , sunlight appears to come from above; that is, from due north. Azimuth  $90^\circ$  is due east,  $180^\circ$  due south, and  $270^\circ$  due west. The 3-D illusion is most convincing when the light comes from the upper left, at roughly  $315^\circ$  azimuth.

The solar elevation is a critical parameter. When the Sun is straight overhead at elevation  $90^\circ$ , all terrain contrast disappears. If the pixel separation is set to a

## Chapter 15: Non-Linear Operators

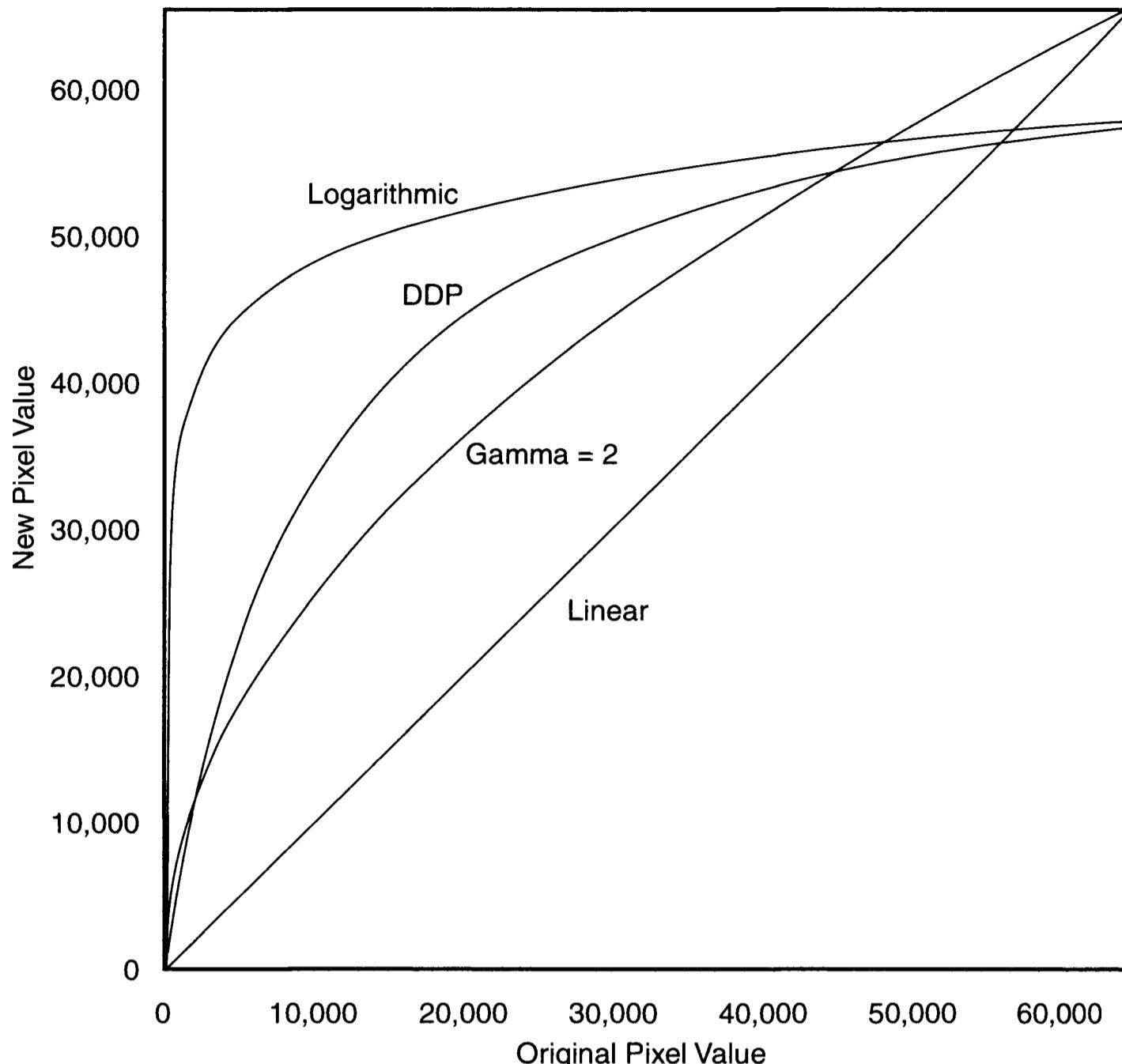


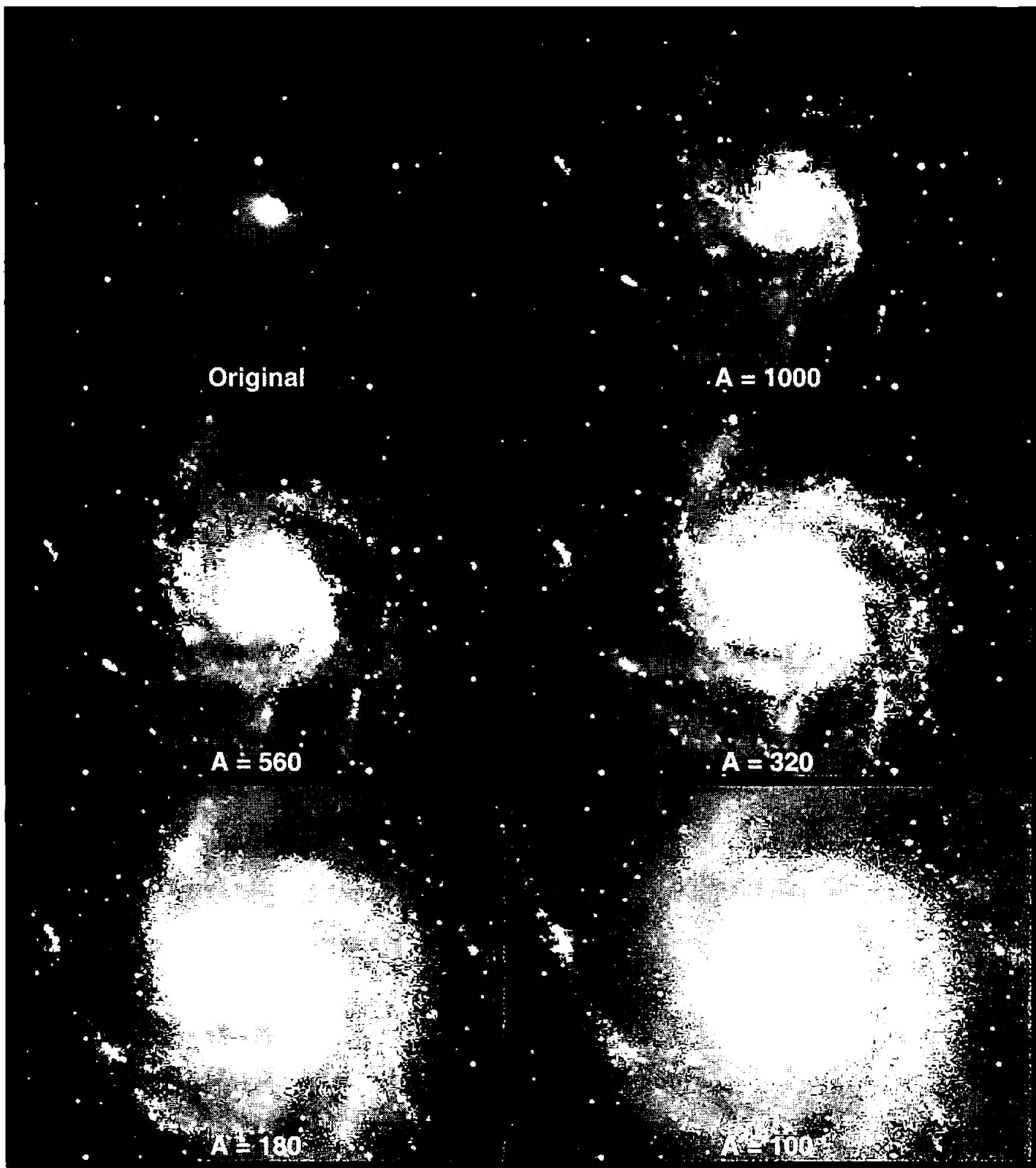
Figure 15.14 The hyperbolic transfer curve of digital development starts below the gamma curve, holding down the sky background, but then passes it to enhance the crucial middle-gray tones. It becomes nearly flat at high pixel values, compressing the dynamic range of astronomical images.

reasonable value, solar elevations between  $15^\circ$  and  $45^\circ$  generally give excellent results. At Sun angles less than  $15^\circ$ , the ground becomes dimmer even though the relief becomes more pronounced.

## 15.6 Digital Development

Digital development is a nonlinear procedure that simultaneously sharpens and compresses the brightness scale of the original image. Its inventor, Kunihiko Okano, coined the term “digital development process” (or “DDP”). Digital development mimics the response of photographic materials, both in its nonlinear brightness scaling and by simulating chemical adjacency effects that gently sharpen photographic images during development.

Two key features make digital development useful: it can squeeze the large dynamic range of a CCD image into a relatively small dynamic range that a computer can display and hard copy can easily handle, and it simultaneously enhances small-scale structure in the image. It is hard to overstate the importance of digital development as a one-step enhancement for deep-sky images.



**Figure 15.15** The shoulder parameter  $A$  controls the appearance of the new image. For consistently good results, set the shoulder parameter to the pixel value of the most significant features in the image; in this example, the spiral arms. Note how subtle edge enhancement makes the image look sharp.

The nonlinear scaling characteristic of digital development is:

$$P_{\text{new}} = \left( \frac{\overline{P}_{\text{usm}} P_{\text{old}}}{\overline{P}_{\text{usm}} + A} \right) + B \quad (\text{Equ. 15.11})$$

where  $P_{\text{old}}$  is a pixel in the old image,  $\overline{P}_{\text{usm}}$  is the average pixel value of an unsharp mask made from the original image,  $P_{\text{usm}}$  is the pixel in the unsharp mask that corresponds in position to  $P_{\text{old}}$ ,  $A$  is a constant chosen by the user equal to the pixel value of greatest interest in the image, and  $B$  is an arbitrary constant (called the toe parameter) that determines the lowest pixel value in the new image.

## Chapter 15: Non-Linear Operators

In the equation above, the term  $P_{\text{old}}/(P_{\text{usm}} + A)$  generates the strongly non-linear hyperbolic scaling. Since  $P_{\text{old}} \approx P_{\text{usm}}$ , the ratio is characterized by a slow rise from zero, a rapid rise to 0.5 as  $P_{\text{old}}$  approaches and then passes  $A$ , and then a slow approach to 1. This term acts as a transfer curve, scaling pixel values from the original image into the new one. With the full equation, the minimum output value is  $B$  and the maximum value is  $\overline{P}_{\text{usm}} + B$ .

The appearance of the new image is controlled by choice of  $A$ . If  $A$  is higher than the sky background, then pixel values in the new sky background will remain low. As pixel values approach  $A$ , they will rise steeply to reach half their maximum value,  $\overline{P}_{\text{usm}}$ , when  $P_{\text{old}} \approx A$ . For pixel values greater than  $A$ , the curve becomes shallow, and the output pixel value approaches its maximum gradually.

The DDP transfer function is shallower than the gamma curve at low pixel value, steeper in the midrange values near  $A$ , and shallower at high pixel values. In sharp contrast to DDP, the logarithmic transfer function is steepest at low pixel values and becomes very shallow at high pixel values. With the right choice of parameters, gammalog scaling is similar to the scaling function of DDP.

The sharpening effect in DDP results from using an unsharp mask to compute the transfer curve rather than the single current pixel. If  $P_{\text{old}}$  is greater than  $\overline{P}_{\text{usm}}$ , then the new pixel will be disproportionately bright compared to its neighbors, and if  $P_{\text{old}}$  is less than  $\overline{P}_{\text{usm}}$ , then the new pixel will be disproportionately dark. If the radius used to create the unsharp mask is set too large, then the mask will create dark circles around bright objects such as stars. For optimum sharpening, the radius of the unsharp mask should be equal to or slightly greater than the half-width half-maximum of the star images.

- **Tip:** *AIP4Win has a DDP tool that enables users to obtain consistent and reproducible results from digital development. The tool asks the user to enter the radius of the unsharp mask, the sky background, the shoulder parameter  $A$ , and the toe parameter  $B$ . The best value for the shoulder parameter is the pixel value in the original image that should display as half the maximum value in the new image.  $B$  can be set to any convenient value, such as 100.*

---

# 16 Image Operations

---

Image operations are procedures carried out with entire images. They include mathematical operations (addition, subtraction, multiplication, division, merge), rank operations (minimum, median, maximum), calibration (dark subtraction and flat-fielding), plus methods used to compare (blinking) and combine (track-and-stack averaging) images. Their value stems from allowing the user to create even more powerful procedures by combining point, linear, and nonlinear operators. This chapter details how those image operations are carried out.

Before going on, let us again clarify the words “image” and “frame.” Both words refer to image data. A set of data that shows the bias offset, dark current, or the flat-field of the CCD is sometimes called a “frame,” while a set of data that shows an object is almost always called an image. The difference is purely one of terminology: there is no intrinsic difference in how the image data are stored or processed.

## 16.1 Image Math

Image math includes adding, subtracting, multiplying, and dividing the pixels in two (or more) images. These operations are carried out between pixels at corresponding  $(x, y)$  locations in the images. The resulting pixel value can be placed into a new image array, or it can replace the values in one of the original images. For clarity in the following sections, results are always placed in a new image.

### 16.1.1 Addition

Image addition means adding the pixel values in two or more images.

$$N(x, y) = O_1(x, y) + O_2(x, y) \quad (\text{Equ. 16.1})$$

where  $O_1(x, y)$  and  $O_2(x, y)$  are pixels in the images to be added, and  $N(x, y)$  is the new pixel resulting from their addition. The addition is carried out for all pixels from 0 to  $x_{\max}$  and 0 to  $y_{\max}$ .

Computer pseudocode for image addition looks like this:

```
FOR y = 0 to ymax  
    FOR x = 0 to xmax
```

## Chapter 16: Image Operations

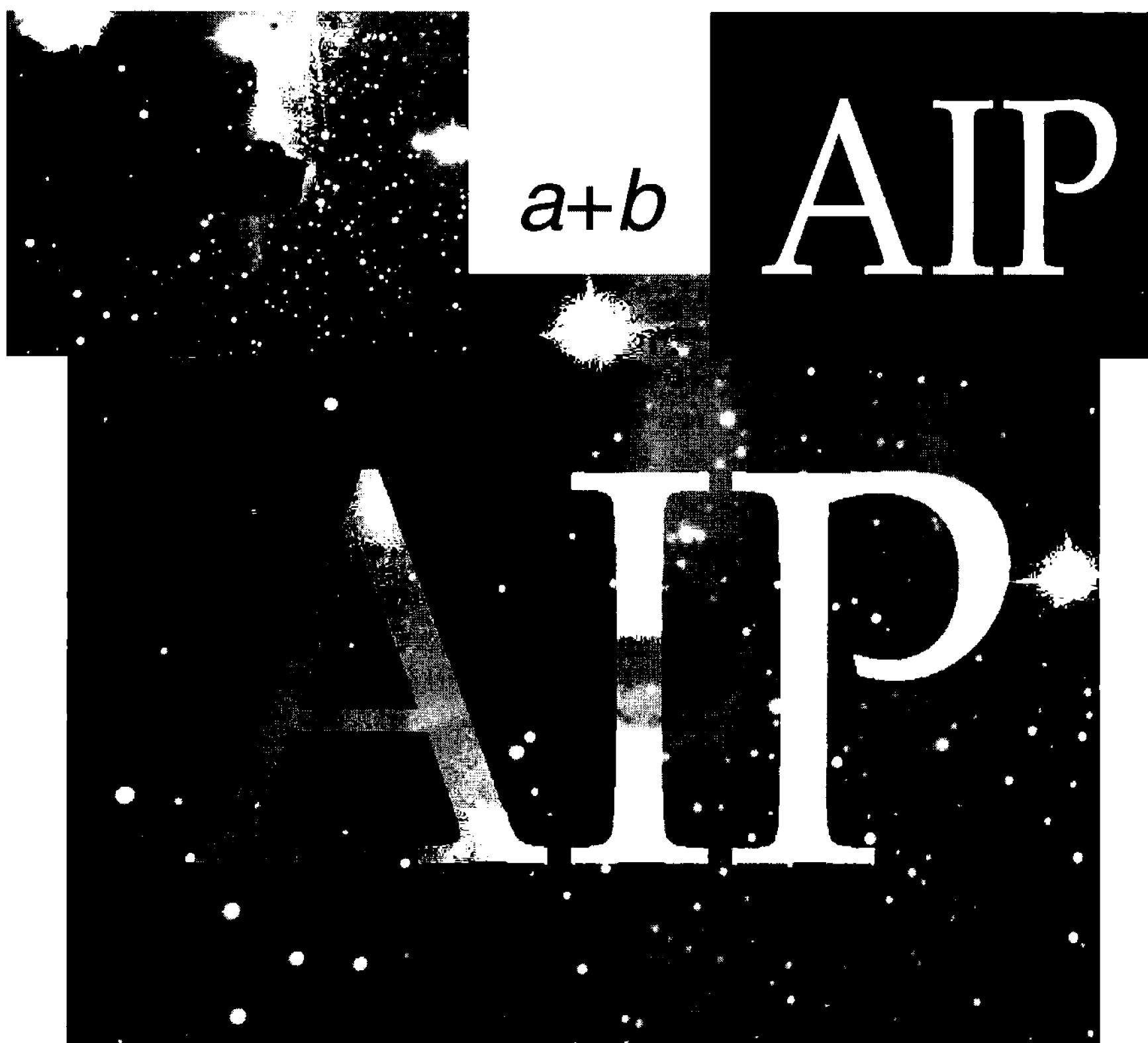


Figure 16.1 Image math is nothing more than a pixel-by-pixel application of a mathematical operation to two or more images. In the images shown above, the value of each pixel in the corresponding locations of the two images is added together; the result is the sum of the two images.

```
new(x, y) = old1(x, y) + old2(x, y)  
NEXT x  
NEXT y
```

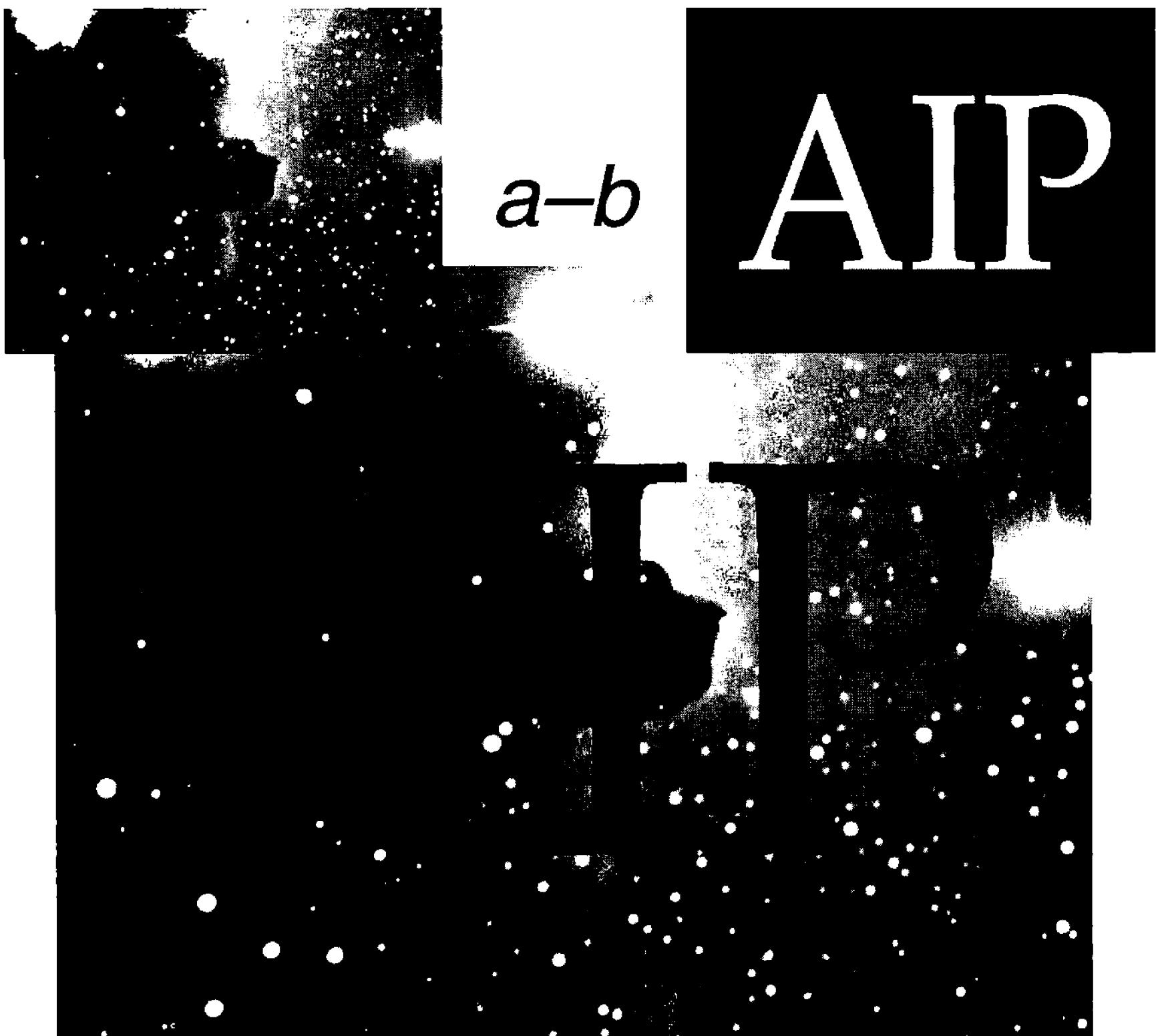
where `old1()` and `old2()` are the source images to be added, and `new()` is the sum image created.

- **Tip:** *AIP4Win does not truncate the result of image addition. It is possible to generate pixel values that are well outside the original range of pixel values found in either image.*

### 16.1.2 Subtraction

Image subtraction means subtracting the pixel values in one image from those in another. Here is computer code to carry out image subtraction:

```
FOR y = 0 to ymax  
    FOR x = 0 to xmax
```



**Figure 16.2** Subtracting one image from another produces a difference image. In this case, pixels in the white letter areas reduce pixel values in the result more than pixels in the black background do. The difference image of two nearly identical images highlights differences that might otherwise be invisible.

```

new(x, y) = old1(x, y) - old2(x, y)
NEXT x
NEXT y

```

where `old1()` is the minuend image (the one subtracted from), `old2()` is the subtrahend image (the one that is subtracted), and `new()` is the difference image created by subtracting them.

Subtraction produces negative values when the subtrahend is greater than the minuend, which may not be an acceptable result. If the pixel value represents light that struck the detector, negative values are physically unrealistic. However, in an application such as a digital terrain map, negative values are physically meaningful (*i.e.*, the elevation of California's Death Valley) and necessary.

- **Tip:** *Negative values are allowed in **AIP4Win** 2.0. If it is necessary to clip pixel values at zero, use the Min/Max Tool to eliminate negative pixel values.*

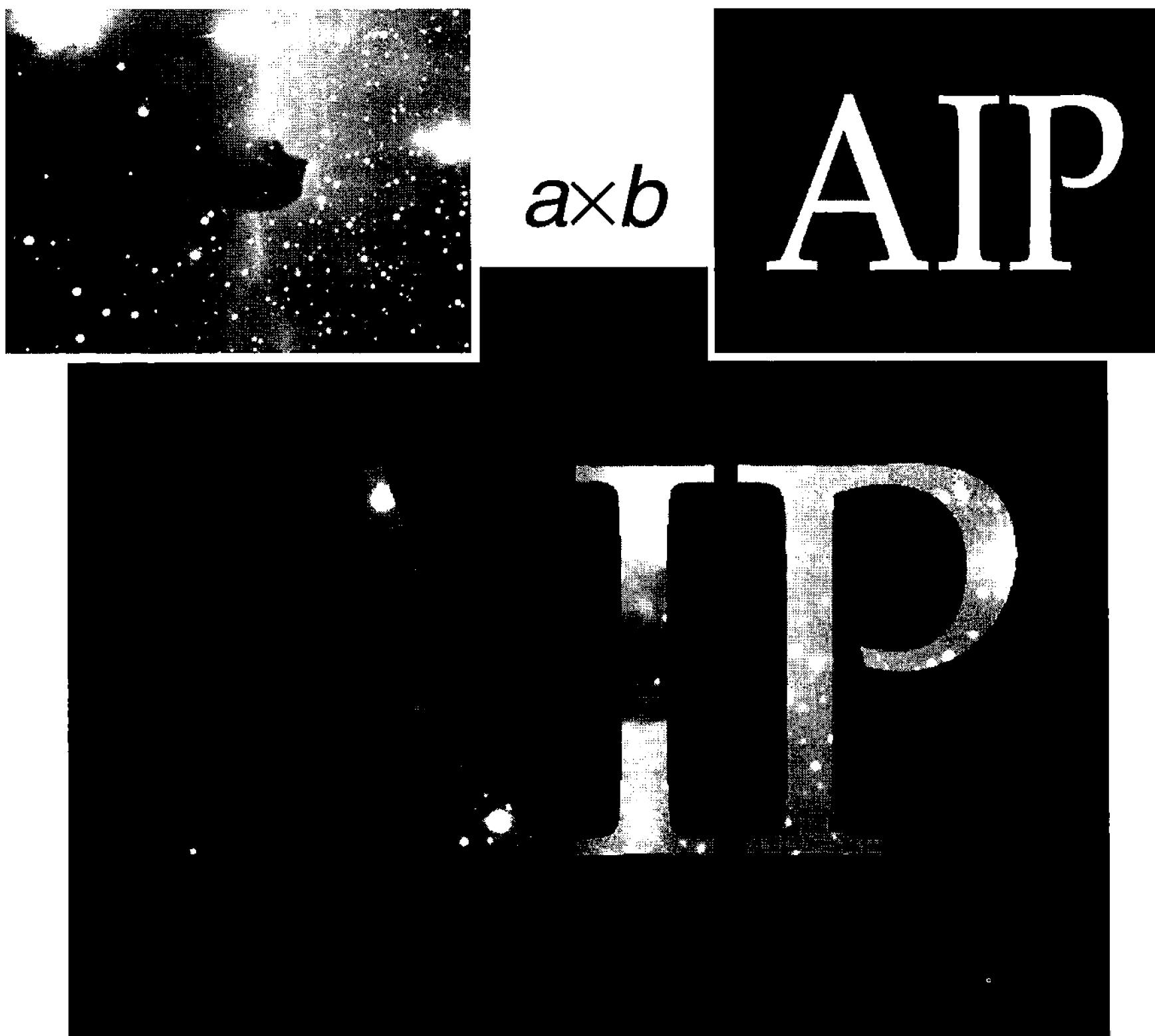


Figure 16.3 Multiplication can produce a wide range of results. In this example, the pixel value of the black background is zero, while that of the white lettered areas is 1.0. When the Horsehead is multiplied by the letter image, background areas become zero while the lettered areas retain their original values.

### 16.1.3 Multiplication

In image processing using integer pixel values, multiplication must be normalized to produce results in the range of values found in the images. However, with floating-point pixel values, multiplication proceeds normally:

```
FOR y = 0 to ymax
    FOR x = 0 to xmax
        new(x, y) = old1(x, y) * old2(x, y)
    NEXT x
NEXT y
```

- **Tip:** *Multiplication is a versatile and powerful image operation. In AIP4Win use it in a sequence of operations to generate discontinuous transfer functions, to highlight ranges of brightness, and to create masks that modify the Fourier transform.*

### 16.1.4 Division

Division is a difficult operation because the results of division have a wide dynamic range, and also because division by zero is an undefined operation.

```

FOR y = 0 to ymax
    FOR x = 0 to xmax
        IF old2(x, y) = 0 then
            new(x, y) = infinity
        else
            new(x, y) = old1(x, y) / old2(x, y)
        END IF
    NEXT x
NEXT y

```

where `old1()` is the dividend image, `old2()` is the divisor source image, and `new()` is the quotient image created by the division. The constant `infinity` is a special value chosen to represent the indeterminate result of division by zero.

The most frequent use for image division in astronomical image processing is flat-fielding, used in image calibration. In flat-fielding, the flat-field image is normalized to a mean value of 1.00. As a result, the quotient has very nearly the same range of pixel values as the original (dividend) image.

### 16.1.5 Absolute Difference

Subtraction gives the difference between two images, but the result may have a negative sign and thus be lost. The function that finds how different two images are—regardless of the arithmetic sign—is the absolute difference:

$$N(x, y) = |O_1(x, y) - O_2(x, y)| \quad (\text{Eq. 16.2})$$

where  $O_1(x, y)$  and  $O_2(x, y)$  are pixels in the original images,  $|x|$  is the absolute difference operator, and  $N(x, y)$  is the resultant new pixel. The absolute difference operator returns  $+x$  whether the argument is  $-x$  or  $+x$ .

In computer code, the absolute difference works like this:

```

FOR y = 0 to ymax
    FOR x = 0 to xmax
        IF old1(x, y) > old2(x, y) THEN
            new(x, y) = old1(x, y) - old2(x, y)
        ELSE
            new(x, y) = old2(x, y) - old1(x, y)
        END IF
    NEXT x
NEXT y

```

All pixels in the resulting `new()` image will be positive.

## Chapter 16: Image Operations

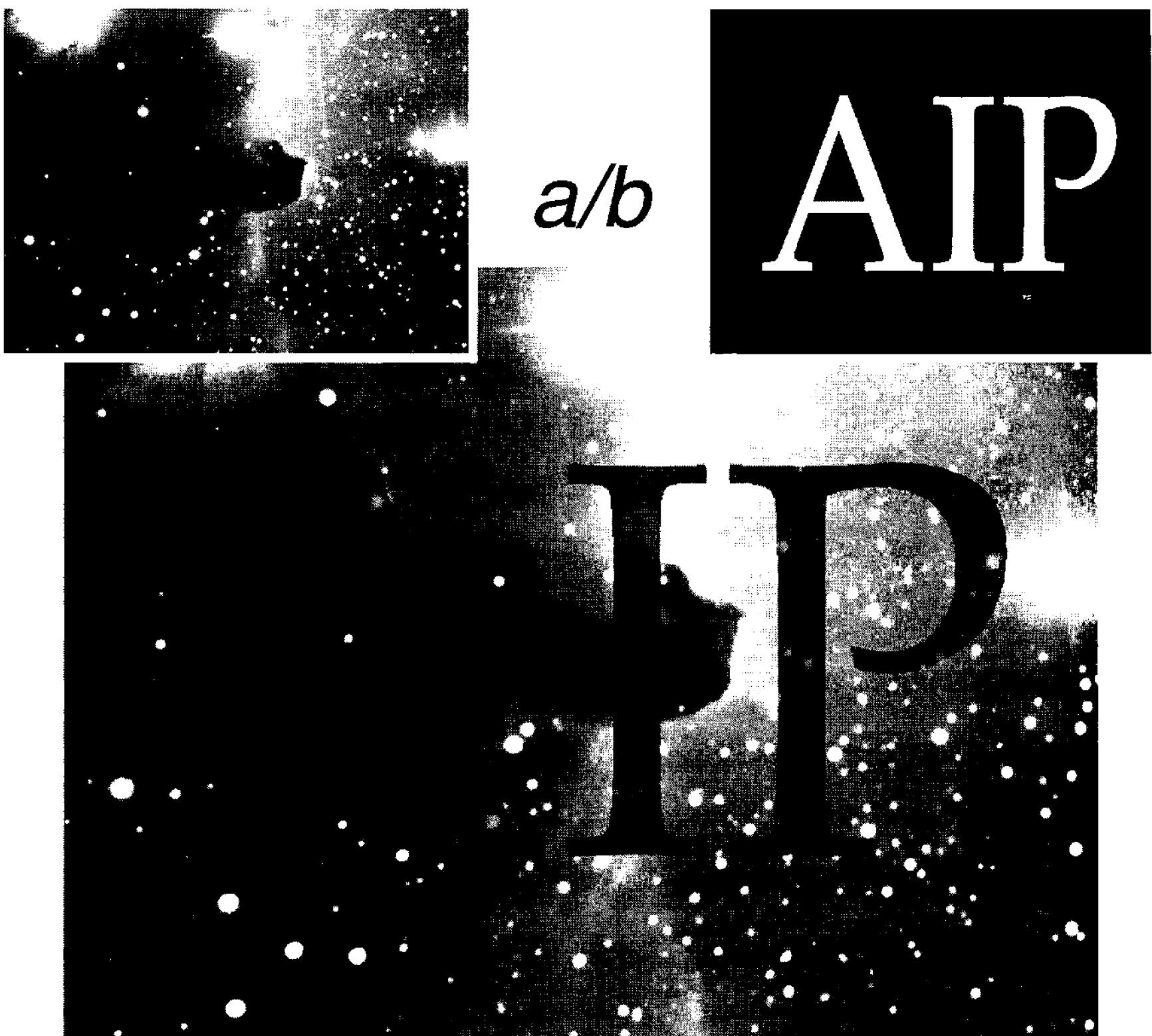


Figure 16.4 Division is another versatile image operation. In this example, the black background areas have a pixel value of 1.0 and the white lettered areas have a value of 2.0. When the Horsehead is divided by the letter images, the lettered regions take on lower values.

- **Tip:** Use absolute difference to find subtle differences between images. The absolute difference between two identical integrations is zero except for the readout noise; but if any feature has changed in either image, it will stand out immediately.

### 16.1.6 Merge

Merging is a more versatile image operation than addition or subtraction. In a merge, the user specifies a multiplier for each image, and the products are summed:

$$N(x, y) = \alpha_1 O_1(x, y) + \alpha_2 O_2(x, y) \quad (\text{Equ. 16.3})$$

where  $\alpha_1$  and  $\alpha_2$  are the coefficients,  $O_1(x, y)$  and  $O_2(x, y)$  are pixels in the images to be added, and  $N(x, y)$  is the new pixel resulting from the merge. Since the coefficients can be positive or negative, merging allows images to be added or subtracted in any relative mixture.

A computer pseudocode for image merging looks like this:

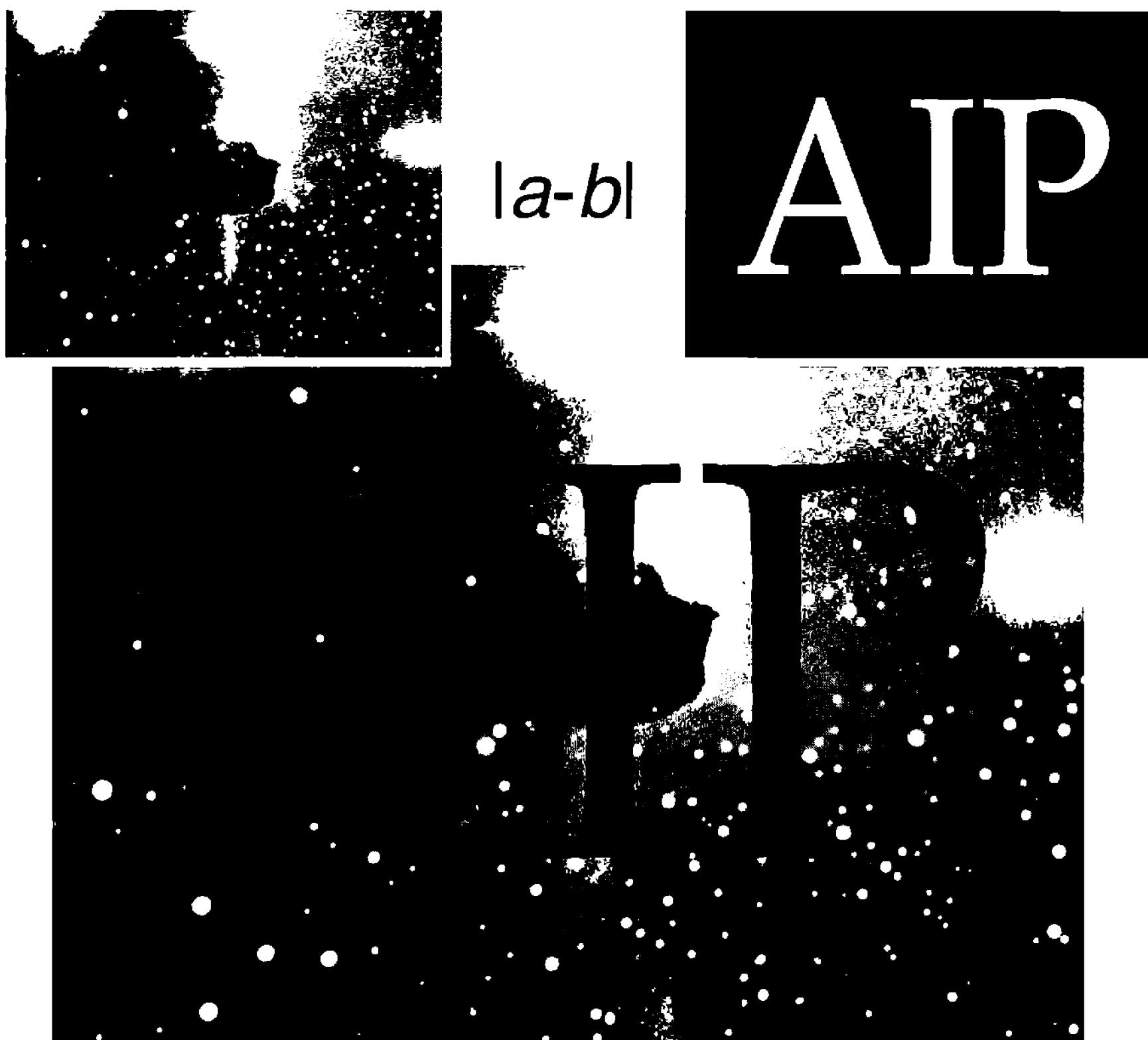


Figure 16.5 The absolute difference means that any difference between the images will be positive. In this example, subtracting the lettered areas produced negative values, but the absolute value became positive, reversing the tone scale in dark parts of the Horsehead image.

```

FOR y = 0 to ymax
    FOR x = 0 to xmax
        new(x, y) = a1 * old1(x, y) + a2 * old2(x, y)
    NEXT x
NEXT y

```

where  $a_1$  and  $a_2$  are coefficients for the two source images. This operation can produce negative values, which are not a problem with floating-point pixel values.

- **Tip:** *Merging is a useful operation. In AIP4Win, you can enhance an image in two different ways, and then use the merge operation to blend the results. For example, process copies of a deep-sky image two ways: one with a strong unsharp mask, the other with an aggressive histogram shaping. By themselves, neither is satisfactory, but merging them obtains the best mix of characteristics from the two images.*

## Chapter 16: Image Operations

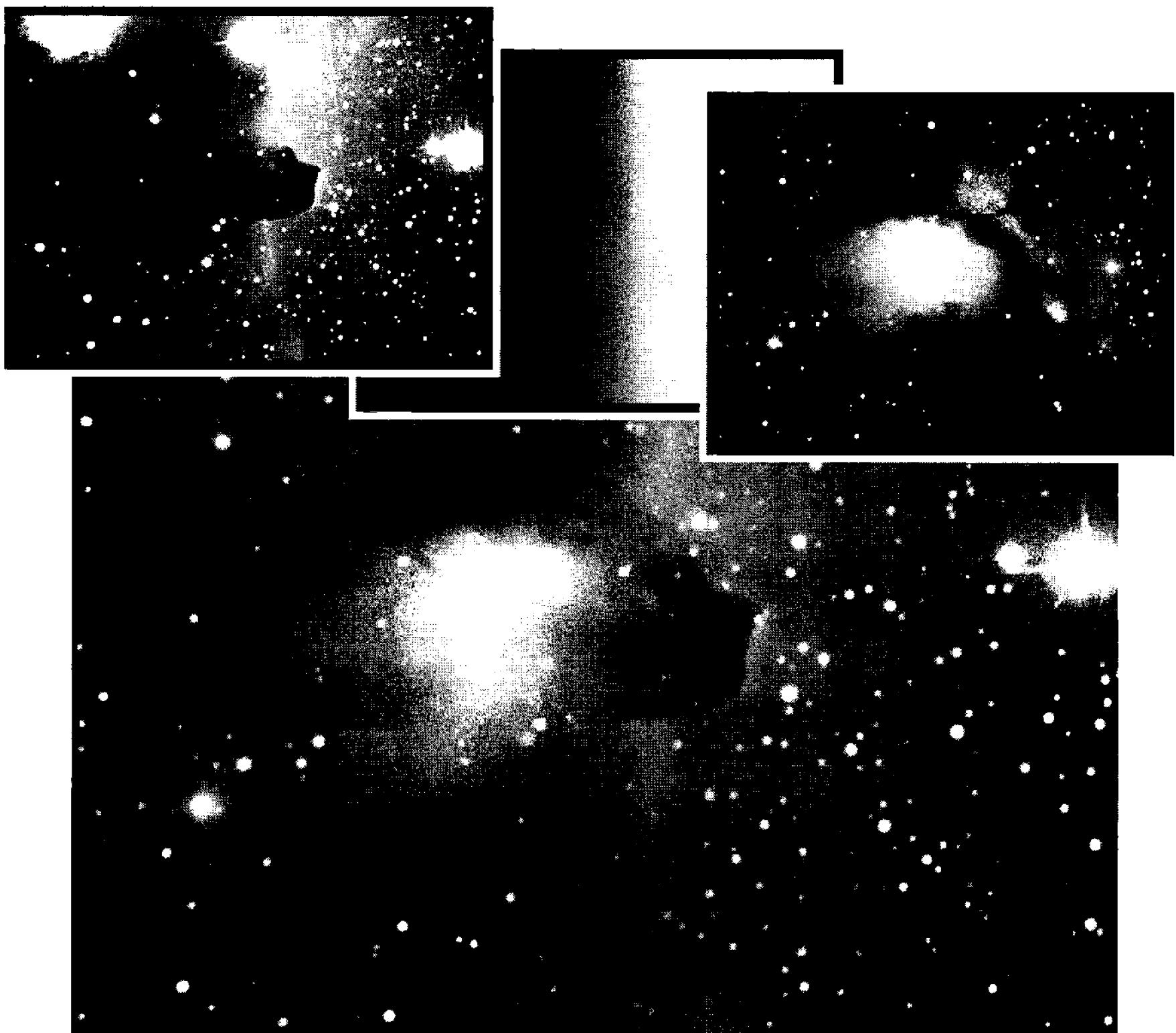


Figure 16.6 An image mask allows you to combine two images using the pixel values in a third image to control the process. In this example, the Horsehead and M78 images are mixed according to values in the masking image, in this case a smooth black-to-white gradient. The result is a blend of the two originals.

### 16.1.7 Average

Adding a set of images and then dividing by the total number of images to find the mean pixel value is called averaging. Averaging images containing random noise reduces the amplitude of the noise by the square root of the number of images, and is therefore a crucial operation for obtaining high-quality images. Averaging is also used to produce master bias frames, master dark frames, and master flat-fields in high-quality image calibration.

Formally, averaging is written:

$$N(x, y) = \frac{1}{n} \sum_{i=1}^n O_i(x, y) \quad (\text{Equ. 16.4})$$

where  $O_i$  is the  $i$ -th image in a set of  $n$  images, and the mean image,  $N(x, y)$ , is understood to be evaluated for every pixel  $(x, y)$  in it.

To average an arbitrary number of images in a computer, each must in turn

be added to an accumulator array. The pseudocode below suggests one way to handle this operation. The `COPYIMAGE` function places an image into the `old()` array, from which it is added to the `sum()` array:

```

FOR ImageNumber = 1 to NumberOfImages
    old() = COPYIMAGE(ImageNumber)
    FOR y = 0 to ymax
        FOR x = 0 to xmax
            sum(x, y) = sum(x, y) + old(x, y)
        NEXT x
    NEXT y
NEXT ImageNumber
FOR y = 0 to ymax
    FOR x = 0 to xmax
        new(x, y) = sum(x, y) / NumberOfImages
    NEXT x
NEXT y

```

After all of the images have been added to `sum()` array, the accumulated total is divided by `NumberOfImages` and placed in the `new()` image.

- **Tip:** *AIP4Win can average open images from the desktop or image files on the hard disk using the Multi-Image | Average tool. AIP4Win's calibration routines can automatically average a set of images to make master bias, dark, and flat-field frames.*

## 16.2 Image Ranking

Rank operations among sets of images provide ways to find extreme values and also to eliminate extreme values. The most useful rank operation is the median, which finds the pixel value for which an equal number of images have a lower pixel value and a higher pixel value. The resulting median image will be free of abnormal pixel values caused by cosmic ray hits and local electronic noise.

Although the minimum and maximum image operations have rather limited and specialized uses, they are included to complete the suite of rank operations.

### 16.2.1 Median Ranking

The median of a set of numbers is the middle value when the set is sorted. Extreme values float to the top of the sorted set or sink to the bottom. The middle part of the sorted set contains values that probably occur frequently, although not necessarily the most frequent value. The median of a series is the value for which an equal number of values are less than or equal, or greater than or equal.

Computing a median image from a set of images proceeds as follows:

```

FOR ImageNumber = 1 to NumberOfImages
    old() = COPYIMAGE(ImageNumber)

```

## Chapter 16: Image Operations

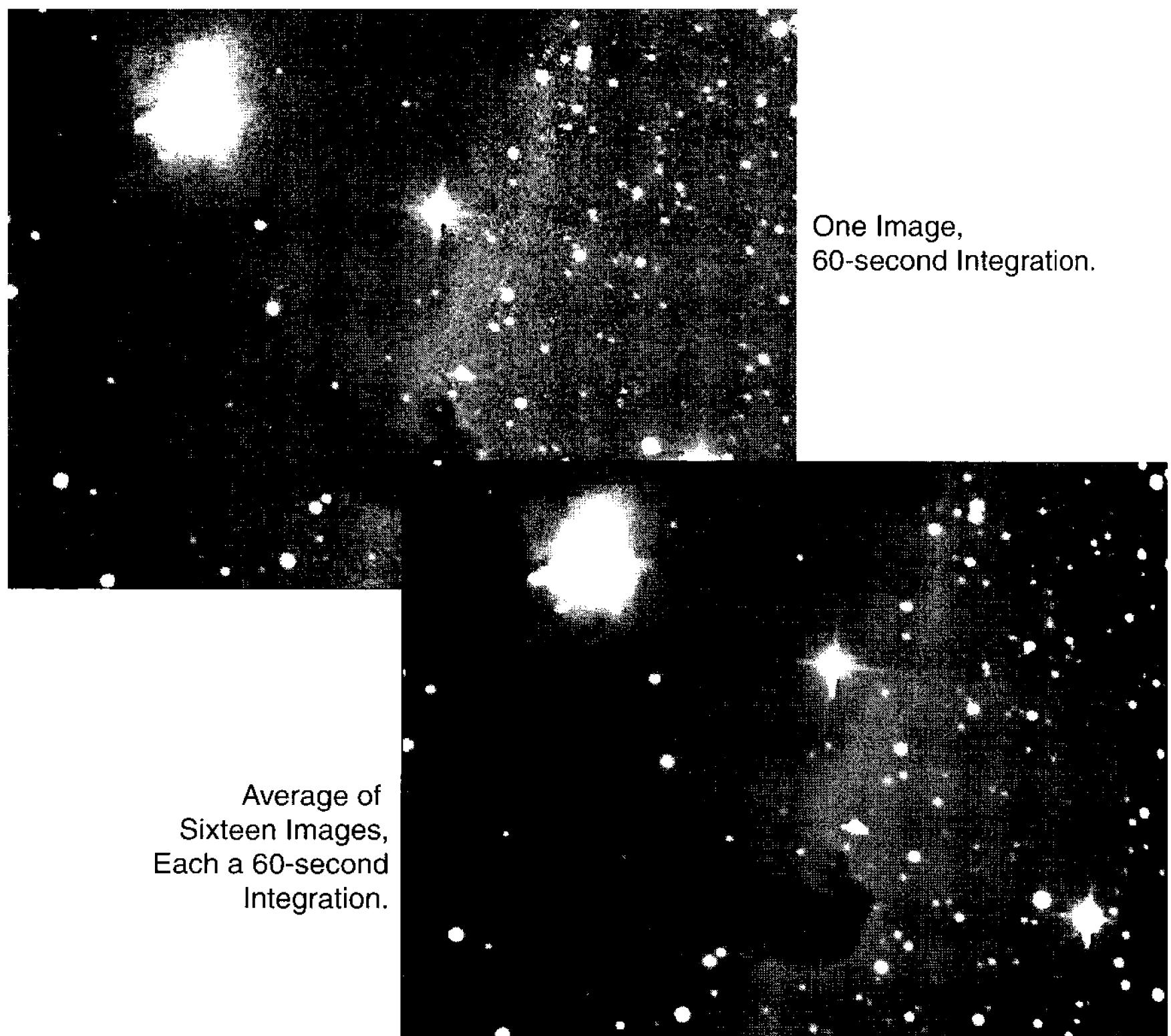


Figure 16.7 Averaging beats down image noise. Of course, each image must be an independent sample of photons—averaging one with multiple copies of itself introduces no new information to the average. In the example, the average of sixteen CCD images has a four-times-better signal-to-noise ratio.

```
FOR y = 0 to ymax
    FOR x = 0 to xmax
        cube(x, y, ImageNumber) = old(x, y)
    NEXT x
    NEXT y
NEXT ImageNumber
```

After the algorithm above has run, the three-dimensional array `cube()` now contains all of the images. To find the median for each pixel ( $x, y$ ), the procedure below copies all of the pixel values at location  $x, y$  into the temporary array, `temp()`. The function `MEDIAN` returns the median value found in `temp()`:

```
FOR y = 0 to ymax
    FOR x = 0 to xmax
        FOR ImageNumber = 1 to NumberOfImages
            temp(ImageNumber) = cube(x, y, ImageNumber)
        NEXT ImageNumber
```

```

new(x, y) = MEDIAN(temp())
NEXT x
NEXT y

```

While conceptually sound, the algorithm outlined above is not a practical way to generate the median of a large number of large images. The size of the cube() array can exceed the memory available.

In a set of CCD images, the median value is unlikely to be influenced by unusual or disturbing events that raise or lower a pixel from its “correct” value; such as cosmic ray hits, airplane, satellite, and meteor trails, moments of bad guiding, and electrical interference. For noise-free bias frames, dark frames, or flat-field frames, median combinations are a good way to go.

- **Tip:** *AIP4Win can generate a median image from open images on the desktop or from image files on the hard disk. Use the Multi-Image | Median tool to select the images. The Calibration tools have the option to create median bias, dark, and normalized flat-field images.*

### 16.2.2 Minimum Ranking

The minimum image operation finds the lowest pixel value for each image location in a set of images. The minimum can be found using the same method as the median, except that instead of the MEDIAN function, the software calls the MINIMUM function:

```
new(x, y) = MINIMUM(temp())
```

Because most noise sources add to the signal, the minimum in a set of astronomical images is likely to be the lowest value produced by the normal statistical variation of readout noise and photon statistics, or if the clouds passed over, an image that received less light than other images in the set. As a consequence, the minimum image operation is little used.

### 16.2.3 Maximum Ranking

The maximum image operation finds the highest value for each pixel location in a set of images. Computing the maximum is done exactly like the median and minimum, except that the MAXIMUM function is used to find pixel values for the new() image array:

```
new(x, y) = MAXIMUM(temp())
```

The maximum operation has some interesting uses, such as determining the average flux of cosmic rays from dark frames and the experimental creation of high-resolution images through the atmosphere. To count cosmic rays, make a set of dark frames. Create a median dark frame; then subtract the median dark frame from each of the dark frames in the set. The resulting images contain random noise (with small pixel values) and cosmic ray hits (with large pixel values). To find the

## Chapter 16: Image Operations

cosmic rays, run a maximum frame operation, threshold the image to eliminate the random noise, and then count the number of cosmic ray events in the result.

Maximum rank sharpening has an experimental application. The idea is to take a series of short-integration images of an object with a small angular diameter, such as Jupiter’s satellite Ganymede, Uranus, Neptune, or the nucleus of a bright comet. The integration time must be short enough to “capture” atmospheric seeing—perhaps 1 second or less. The images brought into registration (a process covered in Section 16.4), and then processed to create a maximum rank image. The idea is that bad seeing enlarges the image, making it dimmer. During moments of good seeing, small details are in sharp focus, making them bright. Extracting the maximum pixel values from a registered set of short-integration images identifies the moments of best focus and assembles the sharpest images.

### 16.3 Calibrating Images

In an earlier chapter, we detailed the theory of image calibration. This section treats calibration as an image operation, as indeed it is. The first two subsections deal with dark subtraction and flat-fielding, and the following three examine the operations used to apply the basic, standard, and advanced calibration protocols.

CCD images consist of multiple layers of signal and noise. The simplest raw image type is a bias frame, an integration of zero duration made in total darkness. Bias frames contain a zero-point offset and readout noise. Dark frames are time integrations made in total darkness. In addition to bias and readout noise, a dark frame contains a dark current signal and the noise in that signal.

Raw images have everything that dark frames do, plus they contain photoelectrons generated by light falling on the CCD and the noise in the signal. Flat-field frames are specialized raw images of a uniform field of view, made to determine the relative sensitivity of the individual pixels on the CCD, and to map the distribution of light falling on it.

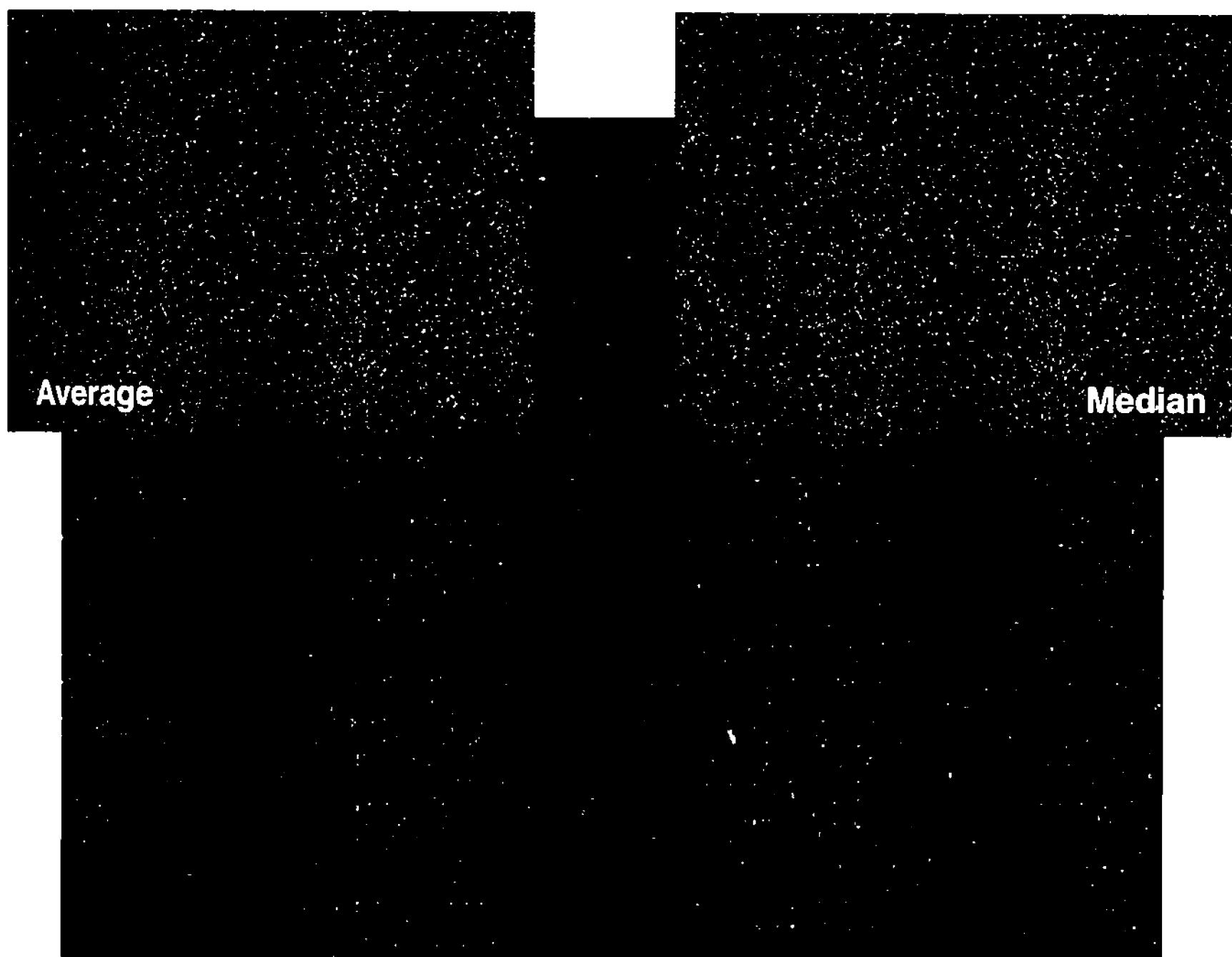
The goal of image calibration is to extract the signal generated by light. By taking bias, dark, and flat frames, it is possible to remove the bias, subtract the dark current, and divide out the nonuniformity in the CCD.

- **Tip:** *AIP4Win supports bias correction, dark frame subtraction with matching, and flat-fielding, using the basic, standard, and advanced calibration protocols.*

#### 16.3.1 Dark Subtraction

Dark subtraction is image subtraction applied to the specific task of removing bias and dark current. Since a dark frame contains bias plus dark current, and a raw image contains bias, dark current, and photoelectrons, the difference contains only photoelectrons. Here is pseudocode for dark frame subtraction:

```
FOR y = 0 to ymax  
    FOR x = 0 to xmax
```



**Figure 16.8** The average and the median of a group of images—ten 300-second dark frames here—may look similar, but there are crucial differences. The large image is the average *minus* median difference. Single pixels show extreme values absent from the median, and pixel clusters are cosmic ray hits.

```

difference = raw(x, y) - dark(x, y)
IF difference <= 0 THEN
    rawdarksub(x, y) = 0
ELSE
    rawdarksub(x, y) = difference
END IF
NEXT x
NEXT y

```

where `raw()` is the raw image, `dark()` is the dark frame, and `rawdarksub()` is the resulting dark-subtracted image.

For dark subtraction to work properly, the integration times for the raw image and dark frame must be the same, and the bias offset must not have drifted during the interval between the taking of the two frames. If the integration times differ, subtraction will not remove all of the dark current and will leave a scattering of hot pixels; or it will remove too much, leaving too-dark pixels sprinkled around the image. If the bias level has drifted, the resulting image may have a falsely elevated background; or the background may be depressed below zero, resulting in a falsely black sky background and loss of information.

## Chapter 16: Image Operations



**Figure 16.9** Unsightly dust donuts and vignetted corners conspire to spoil this image of M78. Even though the darkest dust spot blocks only 7% of the light reaching the CCD, histogram shaping greatly enhances the visibility of the blemishes. Fortunately, good flat-field frames were available (see Figure 16.10).

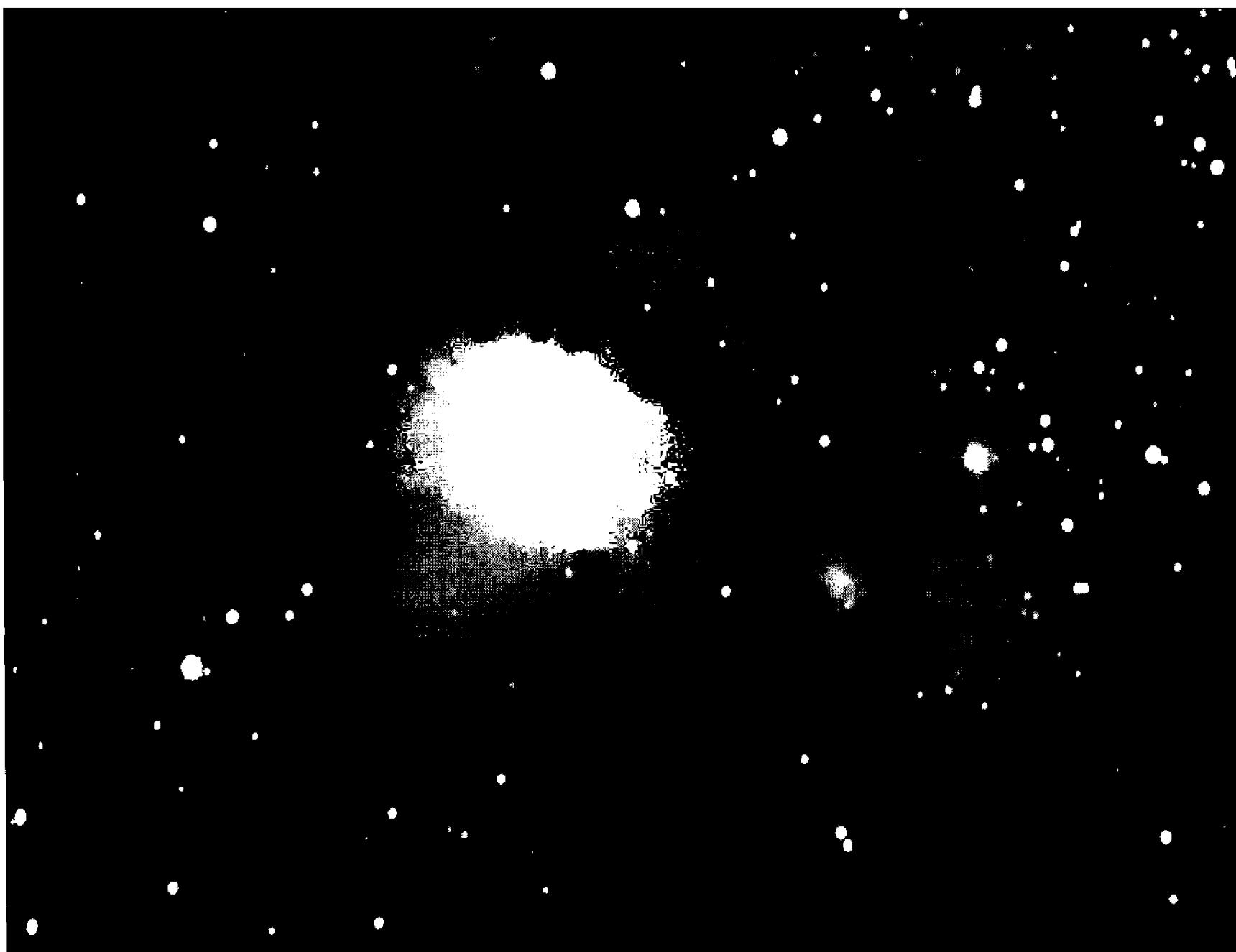
### 16.3.2 Flat-Fielding

Flat-fielding is a specialized type of image division. The dividend is a dark-subtracted image, and the divisor is a flat-field image purposely made using a uniform source of illumination. Because a raw flat-field frame is an image, it is necessary to make dark frames (“flat darks”) to recover the photoelectron signal.

Flat-fielding consists of four steps. Initially, the flat must be created from the raw flat and the dark. Next, you must find the mean pixel value of the flat frame. Finally, the dark-subtracted image is divided by the flat field.

The first step is to create the array `flat(x, y)`, the flat-field frame:

```
FOR y = 0 to ymax
    FOR x = 0 to xmax
        difference = rawflat(x, y) - darkflat(x, y)
        IF difference <= 0 THEN
            PRINT "Bad data in flat frame!"
        ELSE
            flat(x, y) = difference
        END IF
    NEXT x
```



**Figure 16.10** With the dust donuts removed and the dark corners corrected, M78 now looks great. Flat-fielding is crucial for making good deep-sky image because these defects prevent you from applying the strong enhancements needed to reveal detail and faint outer parts of nebulae and galaxies.

NEXT y

where `rawflat()` is the raw flat-field image, `darkflat()` is the dark frame for the raw flat, and `flat()` is a dark-subtracted image. Because a properly exposed flat frame should have pixel values well above zero, the algorithm prints a warning message to alert the user to potentially bad data.

The second step is to compute the mean pixel value of the flat-field frame. This value allows us to create a new image in which the range of pixel values closely matches that in the dark-subtracted image. The mean is computed by summing the pixel values and counting the number of pixels in a defined region around the center of `flat()`. It is a good idea to avoid the margins of the image because vignetting there may be severe, and also because defects are most common away from the center of a CCD. In the code below, the outer 20% of the flat—64% of the area of the CCD—is excluded:

```
pvtotal = 0
pixels = 0
FOR y = 0.2 * ymax to 0.8 * ymax
    FOR x = 0.2 * xmax to 0.8 * xmax
        pvtotal = pvtotal + flat(x, y)
        pixels = pixels + 1
```

## Chapter 16: Image Operations

```
NEXT x  
NEXT y  
pvmean = pvtotal / pixels
```

Next, the `flat()` array is normalized to 1.00:

```
FOR y = ymax to ymax  
    FOR x = xmax to xmax  
        flat(x, y) = flat(x, y) / pvm  
    NEXT x  
NEXT y
```

The final step is flat-fielding. If the value in `flat()` is greater than 1, and the new image will be darker at that location; if it is less than 1, the new image will be adjusted upward to compensate for receiving too little light:

```
FOR y = 0 to ymax  
    FOR x = 0 to xmax  
        new(x, y) = rawdarksub(x, y) / flat(x, y)  
    NEXT x  
NEXT y
```

In an image processing program, `flat()` can be computed and stored for use with all images taken with the camera setup used to make the flat.

### 16.3.3 Basic Calibration

The basic protocol is the simplest form of calibration, but often it's all that is necessary. It specifies taking images and multiple dark frames having the same integration time.

Before calibration, it is necessary to create a master dark frame by combining as many dark frames as were taken under the same conditions as the raw images. Combining dark frames reduces their noise content. The master dark may be the average of the dark frames (see Section 16.1.7), or the median of the dark frames (see Section 16.2.1). Taking an average results in lower noise but leaves a small number of extreme values, whereas the median is a little noisier but is free of cosmic ray hits and electronic noise. The master dark frame is stored in memory for use with many different raw images.

Dark-frame subtraction is carried out using the master dark frame, as described in Section 16.3.1.

Basic calibration relies on the stability of the CCD camera's temperature and amplifier electronics. If the bias level changes, if the temperature drifts, or if raw images have a different integration time than the master dark frame, basic calibration will show hot pixels, dark pixels, or a sky background level that is either too low or too high.

### 16.3.4 Standard Calibration

This is the method of calibration used by most amateurs. It consists of simple dark-frame subtraction followed by flat-fielding, exactly as described in Sections 16.3.1 and 16.3.2. Dark frames for standard calibration have the same constraints as those for the basic protocol: the integration time for raw images and the dark frame must be the same.

Standard calibration also relies on the thermal and electronic stability of the CCD camera. The bias level must not change, and the temperature must not drift more than a very small amount. Flat-fielding relies on the optical and mechanical stability of the telescope and CCD camera between the time that the observer takes raw images and the time the flats and flat darks are taken.

### 16.3.5 Advanced Calibration

This is a sophisticated calibration procedure. It requires the observer to make bias frames, and from them make a master bias frame. Subtracting the bias from a master dark frame leaves a thermal frame, which can be linearly scaled to match the dark current in raw images taken with any integration time. This represents a significant gain in observational flexibility.

Given master bias, dark, and flat frames, here is a pseudocode for the advanced procedure:

```

FOR y = 0 to ymax
    FOR x = 0 to xmax
        boffset = bias(x, y)
        thermal = alpha * (dark(x, y) - boffset)
        new(x, y) = raw(x, y) - boffset - thermal
        new(x, y) = new(x, y) / flat(x, y)
    NEXT x
NEXT y

```

where `raw()` is the raw image, `bias()` is a master bias frame, `dark()` is a master dark frame, `flat()` is a master flat-field frame, and `new()` is the resultant calibrated image. The tricky part is to determine `alpha`, the scaling coefficient. To an excellent approximation, `alpha` is the ratio of integration times in the raw image and dark frame. If the camera's bias frames are uniform and stable, it is advisable to replace the bias frame with a single bias offset value, represented by the variable `boffset` in the algorithm above.

Advanced calibration allows the observer more freedom in selecting integration times. However, the bias level must not change, and the temperature of the CCD should not change more than a few degrees. Flat-fielding relies on the light in raw images and the flat-field integrations taking the same path. Optical or mechanical changes between the times that raw images and the flats and flat-darks are taken can cause calibration errors.

Calibration is simply a set of image operations tailored to the specific tasks

## Chapter 16: Image Operations

of removing the bias offset, subtracting dark current, and correcting the nonuniformities in the CCD and its illumination. For unusual calibrations, observers can use standard image operations to accomplish the same task manually.

- **Tip:** *AIP4Win allows observers to carry out calibration either manually or using automated software tools that make Basic, Standard, and Advanced calibrations both fast and flexible.*

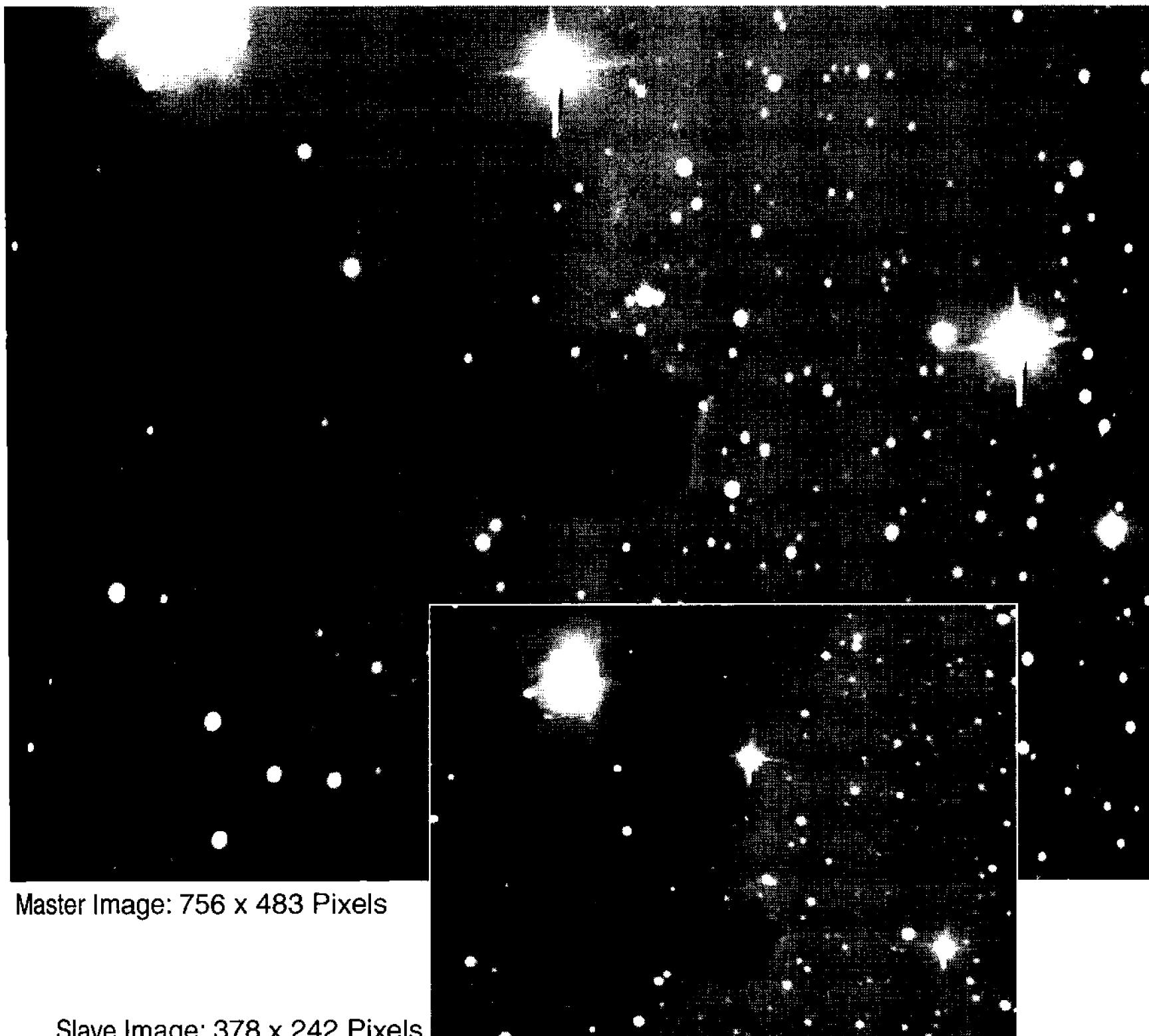
### 16.4 Image Registration

Astronomers often compare images taken at different times to search for comets, asteroids, variable stars, optical gamma-ray bursts, and supernovae. They also combine images to improve the signal-to-noise ratio over what they can obtain in a single integration, or to composite images into mosaics or color images. However, because telescope pointing is never *exactly* the same in two images, they must be brought into coincidence, or registered, before the images can be compared. Registration thus serves as a basic step in a variety of high-level processes.

Two images are said to be “in register” when corresponding features appear at the same pixel coordinates. Stars are ideal reference points for images, because they are both plentiful and fixed in space; but virtually any well-defined image feature can be used as a reference point. For precise registration, the image processing software must be able to compute a centroid for each reference point, under good conditions, to better than 0.05 of a pixel.

Registration requires a “master” image whose features can serve as reference points. In the “slave” image or images, the same reference points must also be visible. Images can be out of register in translation, rotation, scaling, or any combination.

- **Translation** means the reference points have shifted side to side or up and down because of tracking error and declination drift. If images are translated only, then one reference point is needed to register the images.
- **Rotation** usually occurs when a CCD camera is taken off the telescope, and with some focusing units, when the telescope is refocused. Severe field rotation occurs for CCD cameras mounted on alt-azimuth telescopes, except for a brief period when the object crosses the celestial meridian. Images taken with equatorially-mounted telescopes over a time span of hours are usually only translated, but those taken three or four hours apart or on different nights may be slightly rotated if the telescope is refocused. Polar alignment errors in an equatorial mounting also produce field rotation. It is rare to see field rotation without translation. To correct for rotation, it is necessary to measure two reference points.



**Figure 16.11** These two images differ in translation, rotation, and scaling because they were taken with a Cookbook camera with different operating modes—yet after image registration, they will be aligned. To register images like these, it is necessary to mark the same two stars as reference points in each one.

- **Scaling** is usually a small effect for images taken with one telescope, but images taken with different instruments show large scaling differences. Images taken with different telescopes, filters, or CCD cameras are invariably also translated and rotated relative to one another. Two reference points are needed to correct for scaling differences.

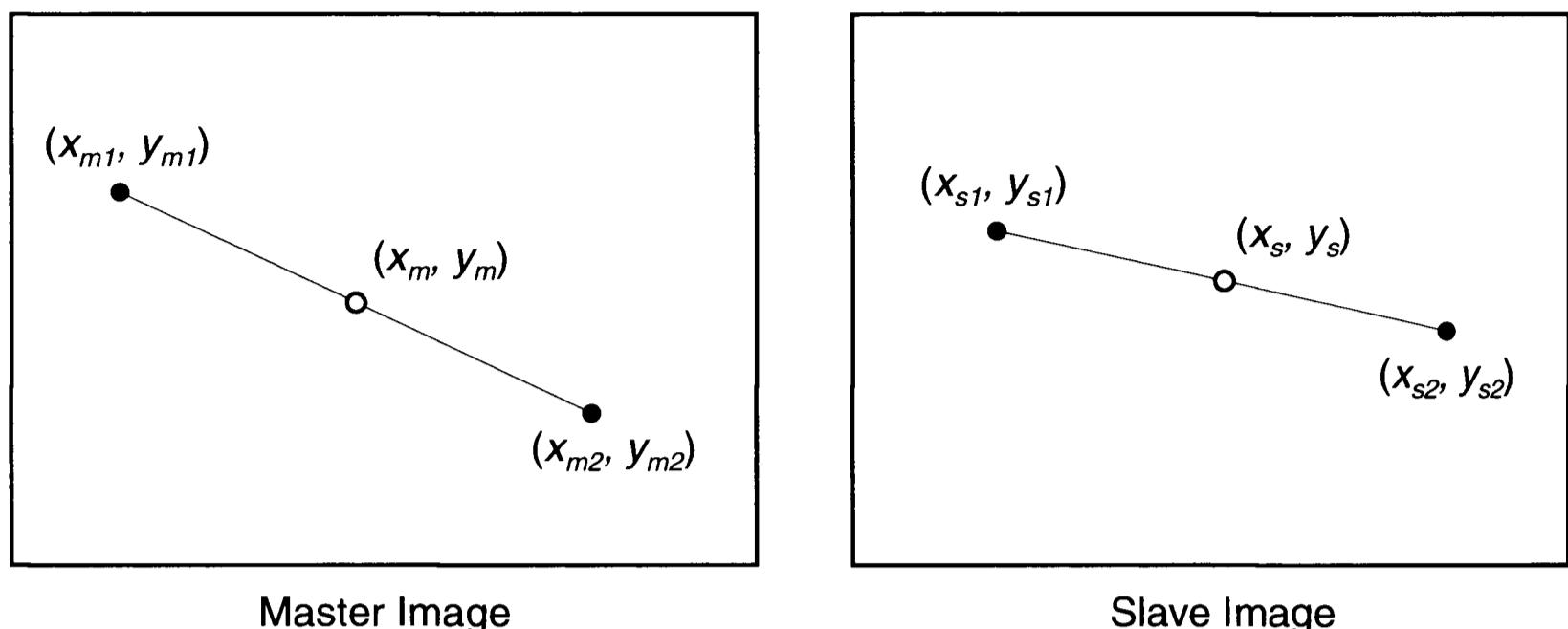
#### 16.4.1 Registration with Translation Only

Registration in translation only requires measuring one reference point in the master image, and measuring the same reference point in each slave image. If possible the reference points are isolated star images exposed to roughly half the full-well capacity of the CCD.

If the coordinates of the reference point on the master image are measured as  $(x_m, y_m)$ , and the coordinates of the corresponding reference point on the slave image are measured as  $(x_s, y_s)$ , then the translations,  $\Delta x$  and  $\Delta y$ , are:

## Chapter 16: Image Operations

## Reference Coordinates for Registration



**Figure 16.12** Solid dots are the reference points in these images, and the hollow dots, midway between each pair of reference points, are the centers for rotation and scaling. Astronomical images are easy to register because their star images make excellent reference points.

$$\begin{aligned}\Delta x &= x_m - x_s \\ \Delta y &= y_m - y_s.\end{aligned}\quad (\text{Equ. 16.5})$$

$\Delta x$  and  $\Delta y$  describe the translation of the slave image relative to the master, in units of pixels. Positive values of  $x$  mean the reference point has moved to the right; negative values mean the reference point in the slave image is left of that in the master image. For registration, the slave image should be translated by  $-\Delta x$  and  $-\Delta y$ . Section 12.1 describes the mechanics of image translation.

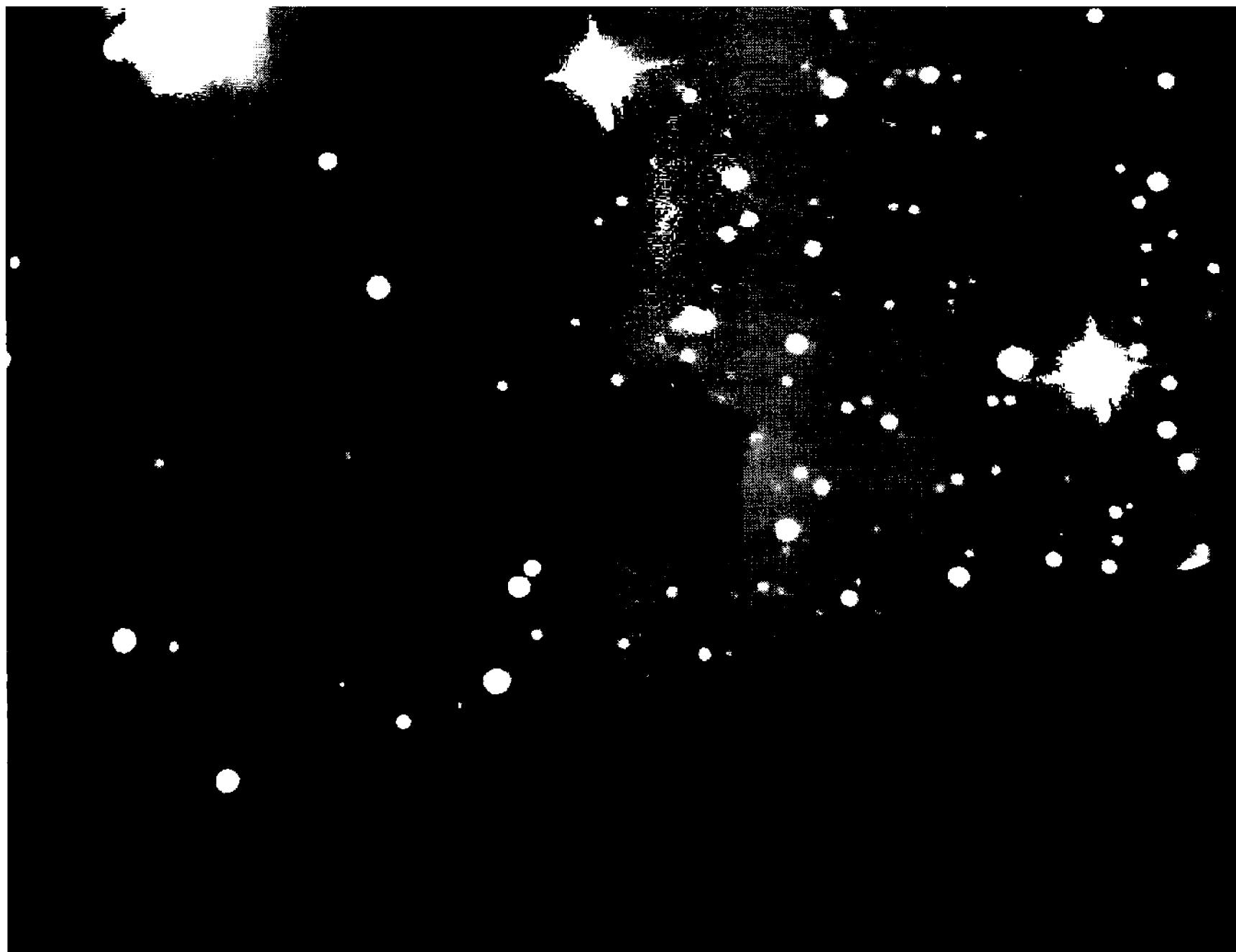
## 16.4.2 Registration with Translation, Rotation, and Scaling

Two reference points are required to register images that are translated, rotated, and scaled relative to one another, plus you must have some way to select the center for rotating and scaling each slave image. In addition, if the CCD does not have square pixels, you must supply the pixel aspect ratio of the image (or images).

Registration begins by measuring coordinates for two points on the master image,  $(x_{m1}, y_{m1})$  and  $(x_{m2}, y_{m2})$ , and two corresponding reference points on the slave image,  $(x_{s1}, y_{s1})$  and  $(x_{s2}, y_{s2})$ . For good accuracy, the reference points should be located in diametrically opposite corners; failing that, as far apart as possible.

Although the centers for rotation and scaling could be either reference point, it is convenient to perform rotation and scaling at the midpoint of the two reference points. This location is the same regardless of the rotation or scaling of the original images, and is well suited as the reference point for translation. Accordingly, the equations for this “center” of the master are:

$$x_m = \frac{(x_{m1} + x_{m2})}{2} \quad (\text{Equ. 16.6})$$



**Figure 16.13** The slave image from Figure 16.11 is now registered to the master image. Note that areas outside the original are black—the software cannot create data where there are none. Just for kicks, flip the page rapidly back and forth to see image blinking in action.

$$y_m = \frac{(y_{m1} + y_{m2})}{2},$$

and the corresponding equations for the center of the slave image are:

$$\begin{aligned} x_s &= \frac{(x_{s1} + x_{s2})}{2} \\ y_s &= \frac{(y_{s1} + y_{s2})}{2}, \end{aligned} \tag{Equ. 16.7}$$

and the image translations relative to  $(x_m, y_m)$  in the master image are:

$$\begin{aligned} \Delta x &= x_m - x_s \\ \Delta y &= y_m - y_s. \end{aligned} \tag{Equ. 16.8}$$

We next find the rotational orientation between each pair of reference points in the master and slave images. Although their actual orientations are arbitrary, the difference between the orientations is the rotation of the slave image. The rotational orientation of the master image is:

$$\vartheta_m = \arctan\left(\frac{x_{m1} - x_{m2}}{y_{m1} - y_{m2}}\right), \tag{Equ. 16.9}$$

## Chapter 16: Image Operations

and the rotational orientation of the slave image is:

$$\vartheta_s = \arctan\left(\frac{x_{s1} - x_{s2}}{y_{s1} - y_{s2}}\right), \quad (\text{Equ. 16.10})$$

so the rotational difference is:

$$\Delta\vartheta = \vartheta_m - \vartheta_s. \quad (\text{Equ. 16.11})$$

Finally, we determine the image scales. The image scale is proportional to the distance between the pairs of reference points. In the master image, this distance is:

$$d_m = \sqrt{(x_{m1} - x_{m2})^2 + (y_{m1} - y_{m2})^2}, \quad (\text{Equ. 16.12})$$

and in the slave image, the distance is:

$$d_s = \sqrt{(x_{s1} - x_{s2})^2 + (y_{s1} - y_{s2})^2} \quad (\text{Equ. 16.13})$$

so the scaling of the slave image relative to the master image is:

$$s = \frac{d_m}{d_s}. \quad (\text{Equ. 16.14})$$

The final step is to translate, rotate, and scale the slave image to match the master, for which the translations are  $-\Delta x$  and  $-\Delta y$ , the rotation is  $-\Delta\vartheta$ , and the scaling factor is  $1/s$ . Algorithms for translation, rotation, and scaling are covered in Section 12.4.

For precise comparison between two images, they should be shifted equal amounts in opposite directions. For the master image, translations are  $\Delta x/2$  and  $\Delta y/2$ , the rotation is  $\Delta\vartheta/2$ , and the scaling factor is  $\sqrt{s}$ . For the slave frame, the translations are  $-\Delta x/2$  and  $-\Delta y/2$ , the rotation is  $-\Delta\vartheta/2$ , and the scaling factor is  $\sqrt{1/s}$ .

Astronomical CCD images are extremely well behaved in registration. They offer lots of reference points, and the resulting centroids are accurate to around 0.05 pixels. This means that the errors in translation due to errors in the centroids are roughly  $\pm 0.1$  pixels. If the reference points are well spaced, the rotation errors are roughly  $\pm 0.01^\circ$ , and scaling errors are roughly  $\pm 0.02\%$ . Two-point registration guarantees that any mismatch between images will be considerably smaller than half a pixel at all points on the image. When large errors occur, they are almost always due to mis-identified reference points.

- **Tip:** In **AIP4Win**, access image registration through the *Multi-Image | Registration tool*. Registration is also built into several other tools, where image registration is an essential step.

## 16.5 Blinking Images

“Blinking” gets its name from the blink microscope, a tool that astronomers use to compare photographic plates. In this instrument, relay optics allow the observer to view two photographic plates simultaneously at low magnification. Each plate is lit by a lamp; with both lamps turned on, the observer moves the two plates until the images coincide. By alternating which lamp is turned on, the observer’s view switches between the two plates. Where none of the stars has changed, the two plates look nearly alike, and the observer sees a single flickering view. But if a comet, asteroid, or variable star has changed, the object that varied appears to hop back and forth or pulsate in size. At Lowell Observatory, for example, Clyde Tombaugh carried out the search for Pluto by blinking 12-inch square photographic plates. Blinking is the computer-age equivalent of Tombaugh’s blink microscope.

To blink-compare two images, they must be registered and should be adjusted to have the same brightness and contrast. The computer copies both into blink-buffer memory so that they can be alternately displayed on the computer screen.

Since differences between the images draw the eye, it is important to make search images as alike as possible. Integration times, focus, and filters should be the same. In well-matched images, the atmospheric seeing and atmospheric air-mass will be the same, too. With well-matched images, the background “grain” of the image due to readout noise is the only thing that changes.

Despite the desirability of well-matched images, most observers can successfully blink images that are very different. The human eye/brain system seems to be capable of comparing sharp and bloated images, smeared and round ones, and images with different graininess quite well. However, images that have differing magnitude limits are difficult to blink because every star that appears in one but not the other is a potential discovery.

Observers vary greatly in their preferred blink rate. At the fast end, each image is displayed for about 150 milliseconds; at the slow end, for a full second. Observers may wish to stop the blinking when they suspect an object is changing, then manually switch slowly between the images.

- **Tip:** *AIP4Win has two different ways to blink images. The Multi-Image | Blinking | Images tool compares images that are already loaded. In this tool, the slave image is shifted to match the master. The Multi-Image | Blinking | Files is designed for serious searches involving many image files. This tool shifts both images to an average location, resampling both for greater overlap and easier comparison.*

## 16.6 Track-and-Stack Image Summing and Averaging

Track-and-stack averaging is an operation used widely by astronomers. The Hubble Ultra-Deep Field image, for example, was built from hundreds of individual integrations and assembled by sophisticated track-and-stacking to merge the indi-

## Chapter 16: Image Operations

vidual subimages into a single image.

For amateur astronomers, track-and-stack averaging is a way to collect more photons than is possible in a single integration, and thus to create an image with a very high signal-to-noise ratio. Although shooting a single long integration yields a higher signal-to-noise ratio than does summing multiple images, track and stack imaging offers other significant advantages to the observer:

- Individual integrations can be short so that the telescope need not be guided, although they should be as long as possible.
- Individual images can be checked. Images with aircraft or satellite trails, cosmic rays, and poor tracking need not be used.
- Short individual integrations will be free of blooming.
- Images of faint asteroids or comets can be registered on the moving object itself, to build up a strong image.

Almost all telescope drives track well enough to permit integration times of 15 to 60 seconds. The longer the integrations are, however, the better the final signal-to-noise ratio. With typical amateur instruments, an integration totaling 60 minutes with a CCD camera easily matches or beats the Palomar Observatory Sky Survey taken with the 48-inch Oschin Schmidt. Although there is no theoretical limit to the number of images that can be stacked, a practical one is six to ten total hours of integration; that is, one very deep image per night. For comparison, the Hubble Space Telescope obtained two weeks of data on a region near the north galactic pole.

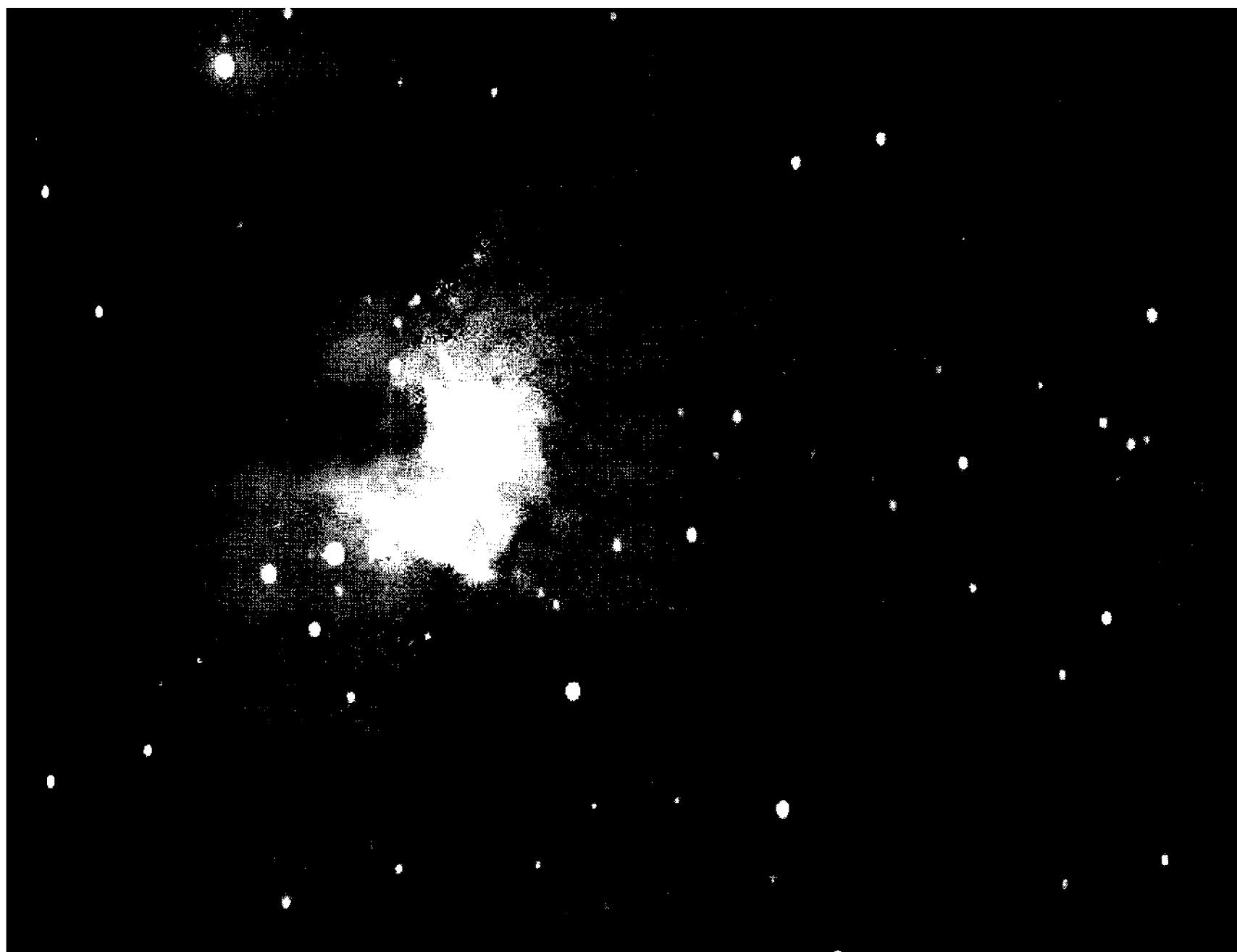
Obtaining good images for track-and-stack imposes several requirements on the observations:

- The total integration time of the master dark frame should exceed the total image integration time.
- Flat-fielding is necessary because the high quality of the resulting image reveals otherwise inconspicuous vignetting.
- The telescope must be constructed so that focus does not shift during the extended imaging time.
- The observer must guard against the formation of dew on the optics during long image series.
- The control computer must be able to store many megabytes of images and dark frames.

After a night of imaging, the observer may have hundreds of raw images of many different objects, plus a large complement of dark frames and flat fields. Although it is possible to register and combine images by hand, it is far more efficient to let computer software do the bulk of the work.

The basic track-and-stacking consists of loading an image, identifying one or two reference points, tracking (or registering) it, and stacking (that is, adding) the registered image to a stacking array. When all of the images have been stacked, the stacking array can be divided by the number of images to give an average val-

## Section 16.6: Track-and-Stack Image Summing and Averaging



**Figure 16.14** Track-and-stack averaging has enabled an observer to capture the wide range of brightness between the bright Trapezium stars and the Orion Nebula. This image combines 150 integrations of two seconds each. The artifacts at left are due to a dark column at the edge of the CCD chip.

ue, or simply saved as the sum.

Below is the pseudocode for the basic algorithm. Each image is copied into the `old()` array, which is then registered. When the registration procedure evaluates a pixel that is outside the limits of the old image, it gives it a unique value called `misscode`. This could be  $-1$  or any pixel value not used in the image itself. The array `stack()` holds the sum of the registered original images. Before a pixel is added to the stack array, its value is checked. If the pixel value is `misscode`, the procedure skips the pixel; otherwise it adds the pixel to the stack array and increments the array `hits()` by one.

```
FOR ImageNumber = 1 to NumberOfImages
    old() = COPYIMAGE(ImageNumber)
    old() = REGISTER(ImageNumber)
    FOR y = 0 to ymax
        FOR x = 0 to xmax
            IF old(x, y) <> misscode THEN
                stack(x, y) = stack(x, y) + old(x, y)
                hits(x, y) = hits(x, y) + 1
            END IF
    NEXT x
```

## Chapter 16: Image Operations

```
NEXT y  
NEXT ImageNumber
```

At this point, `stack()` holds the sum of pixel values registered to that particular location, and `hits()` holds the count of images that have “hit,” or contributed to, that location in the stack array. The final step is to divide the `stack()` array by the `hits()` array to obtain the mean value for each pixel in the contributing images:

```
FOR y = 0 to ymax  
    FOR x = 0 to xmax  
        IF hits(x, y) > 0 THEN  
            new(x, y) = stack(x, y) / hits(x, y)  
        ELSE  
            new(x, y) = 0  
        END IF  
    NEXT x  
NEXT y
```

Note that when images are stored and processed with floating-point pixel values, the average value and the sum value are equally precise.

For astronomical image processing, it is useful to incorporate some additional functions to allow the observer to build finished track-and-stack images directly from raw images. These functions include:

- **Calibration.** As each image in a sequence is loaded from the hard disk, it is dark-subtracted and flat-fielded.
- **Defect Correction.** For images taken with CCDs having point, cluster, and column defects, a defect map can be applied to correct a cosmetic defect automatically.
- **Noise Filtering.** For noisy or poorly dark-subtracted images, a standard noise filter bars hot pixels and noise spikes from the new image.
- **2x Resampling.** Registration involves resampling the image, entailing some loss of detail. By resampling the original image to twice its original size before registration, the loss of detail is held to a minimum.
- **Enhancement.** Nonlinear point operations such as gammalog scaling or Gaussian histogram shaping enhance the visibility of faint objects. As prescaling does, nonlinear enhancement increases the number of gray levels available to represent the higher-quality data. Nonlinearly enhanced images should never be used for astrometry or photometry.

Track-and-stack averaging is one of the most effective tools amateurs can use to extend the basic capabilities of a CCD camera. With telescope drives that do not track more than 30 seconds, track-and-stacking permits an observer rou-

## Section 16.6: Track-and-Stack Image Summing and Averaging



**Figure 16.15** Twenty integrations of 60 seconds each were track-and-stacked to make this image of the central region of M31, the Andromeda Galaxy. The smooth appearance of the nuclear region and visibility of the dust lanes result from the high signal-to-noise ratio achieved in 1200 seconds of integration time.

tinely to reach effective integration times of 10 to 15 minutes. For observers who guide their cameras for single 10 to 15 minute integrations, combining sixteen high-quality deep integrations offers a factor-of-four improvement in the signal-to-noise ratio, and reveals stars more than a magnitude fainter.

## Chapter 16: Image Operations

---

# 17 Images in Frequency Space

---

This chapter deals with the uses of the Fourier transform for altering the spatial characteristics of an image. Up to this point, we have treated images as functions in space—yet there is an entirely different and richly productive way to think of them: as the superimposition of numerous frequencies. Using a mathematical technique called the Fourier transform; we can transform images into an array of frequencies in frequency space, process the image data as frequencies, and then transform the processed array of frequencies back into image space.

In this chapter, you will explore the concepts behind images in frequency space and learn how you can use image processing in the frequency realm to enhance your astronomical images. Frequency space is a mind-bending concept—but it is not necessary to have a complete understanding of the theory to be able to use the concepts behind it. As you work with images in frequency space, you will develop an intuitive feel for Fourier transforms.

## 17.1 Exploring Frequency Space

The fundamental idea behind frequency-domain image processing is that every image can be decomposed into a unique spectrum of spatial frequencies. We can apply this concept by altering the spectrum and then transforming it back to a new image that has changed characteristics. With appropriate alterations to the spectrum, we can enhance image detail, alter the energy distribution in the image, and remove noise—just to name a few of the many applications.

### 17.1.1 Spatial Frequency

When we speak of frequency, we normally mean a signal that varies with time, such as sound waves carried in air or the varying voltages in electrical signals. However, suppose that you map the intensity of light across an image—you see a signal that changes in space. Just as a sound wave varies in time, the intensity of the image varies in space. And just as the sound can be characterized by a frequency in cycles per second, a spatial frequency can be expressed as a frequency in cycles per pixel.

The fundamental similarity between functions that vary in time and those

## Chapter 17: Images in Frequency Space

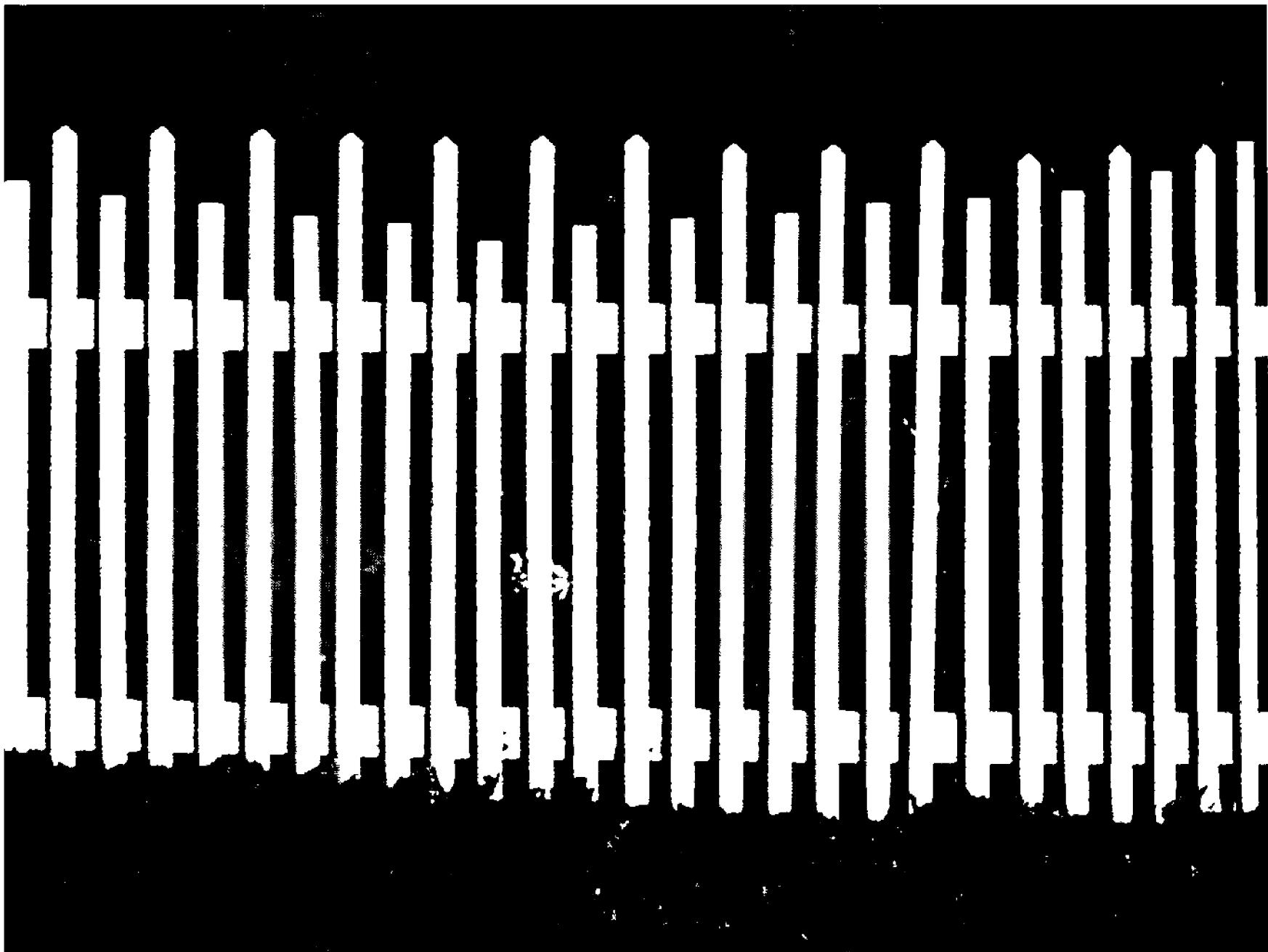


Figure 17.1 Variations in image brightness have a spatial frequency spectrum—although few images so strongly resemble a square-wave pattern as this one of a fence. In this image, 27 cycles span 114 millimeters—giving a fundamental frequency of 0.237 cycles per millimeter (with lots of overtones).

that vary in space is the key to understanding the meaning of spatial frequency. We know that the frequency of a sound wave is the number of times that the air pressure peaks in a specific interval of time. When we hear the musical note A struck on a tuning fork, for example, we know that the sound pressure on our eardrum is rising and falling 440 times per second. Spatial frequencies are analogously defined as the number of cycles in a specific interval in space. A spatial frequency of 64 cycles per millimeter means that the function rises and falls 64 times across the distance of 1 millimeter.

Visually, single spatial frequencies in an image look like stripes or bars. It is easy to imagine assigning a spatial frequency to a picket fence, a Venetian blind, a window screen, or the stripes on a zebra. You can easily take a ruler and count how many times a feature repeats in a given interval—roughly 5 cycles per meter for a picket fence, 60 cycles per meter for Venetian blinds, and 1200 cycles per meter for a window screen. The irregular stripes on a zebra hint at the real complexities found in frequency space.

### 17.1.2 The Frequency Spectrum

Most of us feel comfortable dealing with musical frequencies because we have experienced them all of our lives. When we hear a pure tone, we recognize that

## Section 17.1: Exploring Frequency Space

someone has struck a harmonic oscillator such as a tuning fork. We know that chords are several notes sounded at the same time, so we intuitively grasp that many different frequencies can be present simultaneously. When we look at a graph of the frequencies in a piano note, it comes as no surprise that there is one high peak for the fundamental and many smaller peaks for the overtones and harmonics.

For a single tone, the graph of air pressure at the eardrum is obvious: it is a simple sine wave rising and falling smoothly. Less obvious is the graph of the air pressure for a note with overtones and harmonics. Clearly it must be the sum of the air pressures of the different frequencies, but the shape of the resulting wave does not immediately resemble its constituents: the sum of multiple sine waves is a new function of considerably greater complexity.

Just as sounds are rarely a single pure tone, and therefore composed of a single frequency, a pattern of light in an image rarely consists of a single spatial frequency. Instead, the signal contains many frequencies, from the low ones that make up the gross features of the image to high frequency overtones and harmonics that comprise the fine detail.

Of course, time-based frequencies are plotted on a single axis (time), whereas those in an image exist in the two spatial dimensions of the image. By their very nature, images are two dimensional. To simplify the initial stages of understanding the Fourier transform, however, we shall consider a one-dimensional slice through a two-dimensional image. When we graph this slice, we treat it as a function that varies along a single spatial dimension. The graph rises and falls as the intensity of the light along the slice varies. The graph of an image slice is similar to that of a sound wave that varies along the single axis of time; indeed, as far as the math is concerned, there is no difference at all. If the graph were a sound wave, it would represent air pressure rising and falling as we progress through time; as a slice from an image, it represents light intensity rising and falling as we traverse the image in space.

The mathematical basis of frequency analysis was formulated by the French mathematician Jean Baptiste Joseph Fourier in 1822. What he did was to postulate and prove mathematically that it is possible to break down any periodic function into simple sinusoids; and that by combining simple sine waves, it is possible to recreate virtually any periodic function. This applies to music, speech, electronic signals, and of course, astronomical images.

The power of Fourier's theorem is not immediately obvious. It is clear that combining a few sinusoids can produce a few generic wiggly functions, but the imagination usually initially balks at the idea of breaking down or producing any periodic function using sinusoids. As we shall see, though, with enough sinusoids you can synthesize square waves, triangular waves, and pictures of Jupiter.

### 17.1.3 Sinusoid Basics

To understand how versatile sinusoids really are, recall that three parameters are necessary to describe a sinusoid: period, amplitude, and phase. On a graph, we

## Chapter 17: Images in Frequency Space

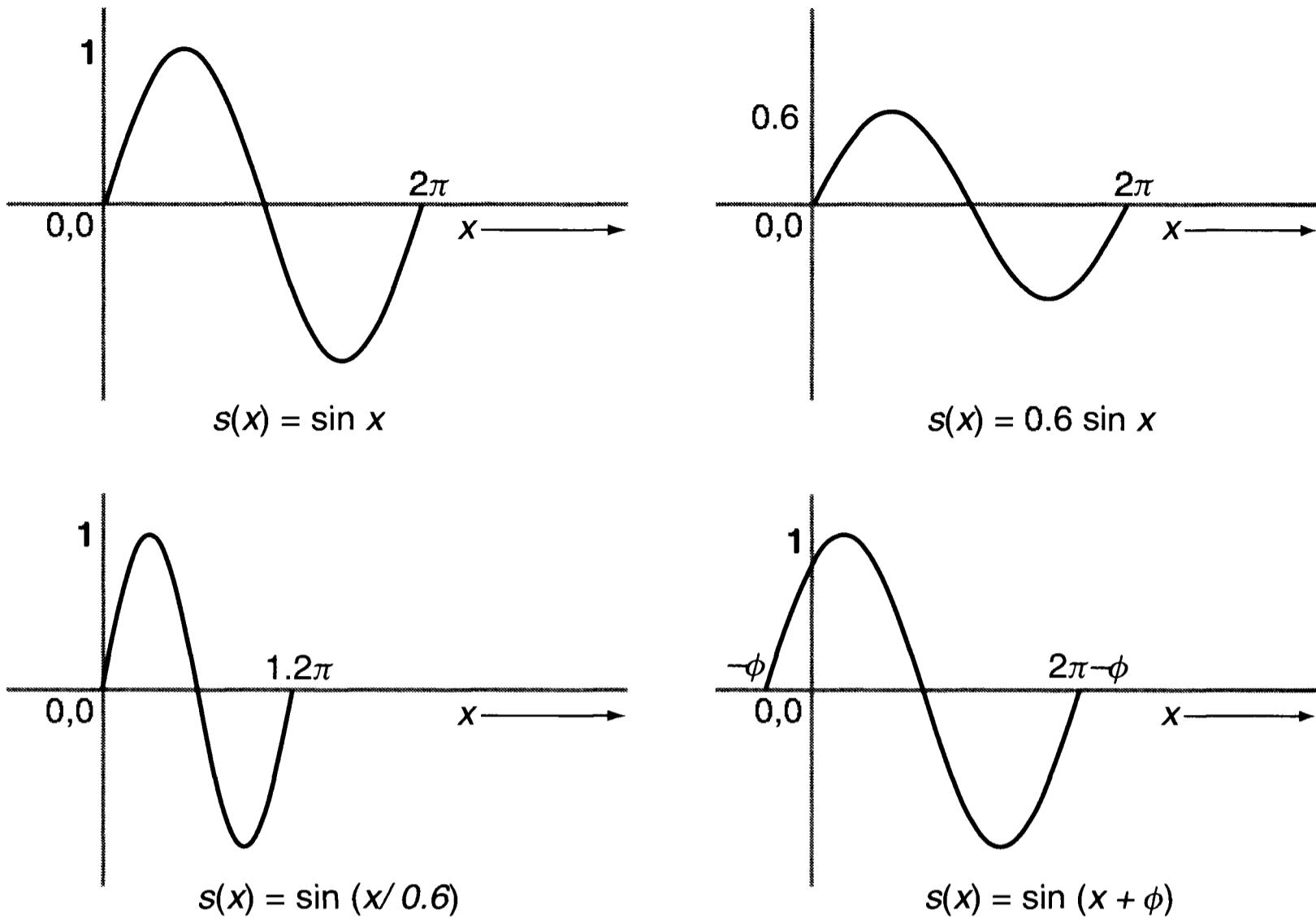


Figure 17.2 Sinusoids have three associated properties: period, amplitude, and phase. The sine function at upper left has a period of  $2\pi$ , an amplitude of 1, and a phase of 0. At upper right, the amplitude is changed to 0.6; at lower left, the period is changed to  $1.2\pi$ , and at the lower right, the phase is shifted by  $-\phi$ .

represent the frequency by the number of peaks in some interval of time or space. The amplitude is the height of the sinusoid from top to bottom, and the phase is the starting point of the sinusoid. Figure 17.2 shows one period for four sinusoids with different frequencies, amplitudes, and phases.

## 17.2 Fourier Theory

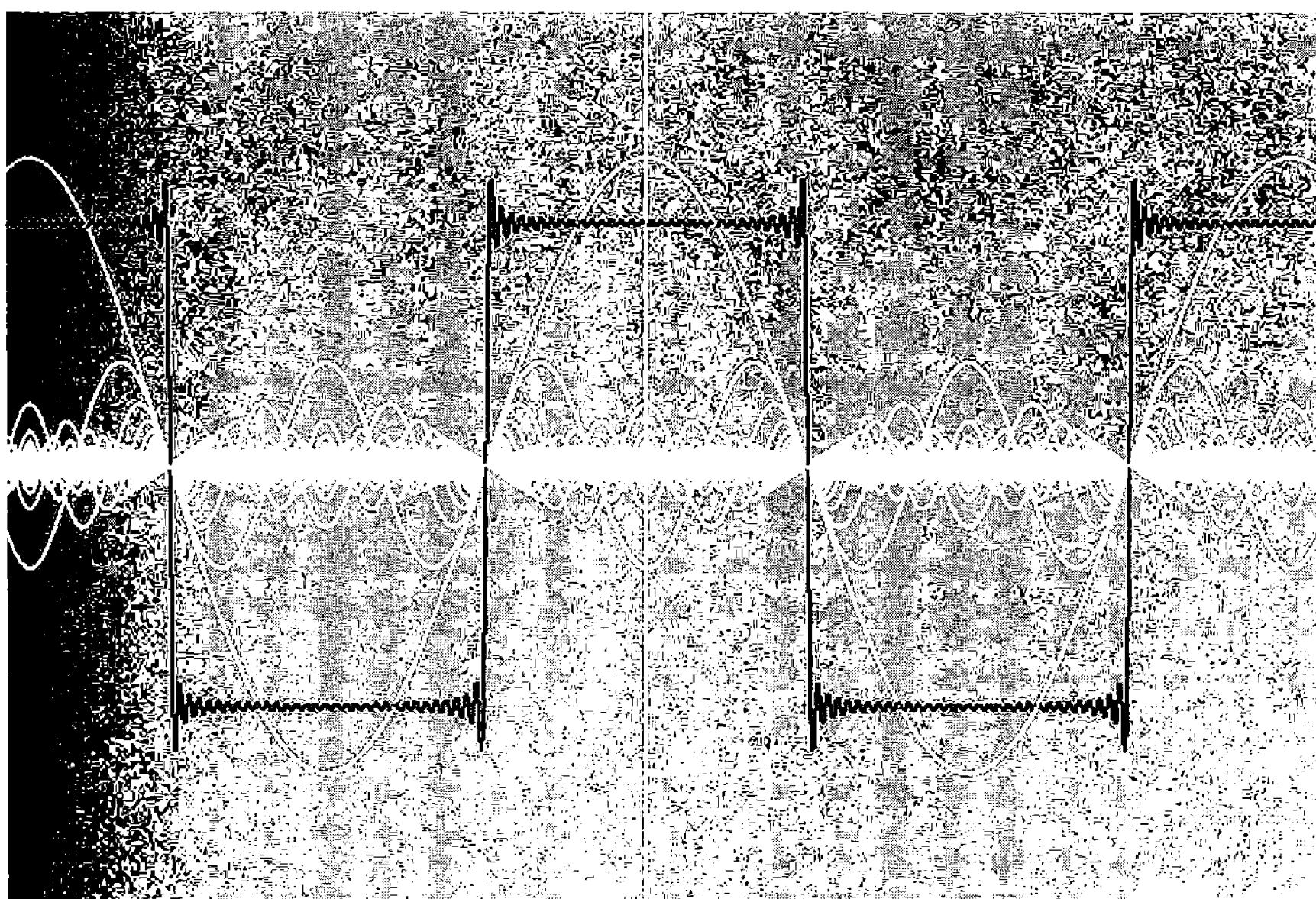
The mathematical theory that Fourier introduced in 1822 was a long way from modern digital image processing. In this section, we examine the four steps that lead from his original series summation to the fast Fourier transform used in image processing: the Fourier series, the Fourier integral, the discrete Fourier series, and the fast Fourier transform. Our goal is not to teach you the math, but to show the chain of reasoning that underlies the Fourier transform.

### 17.2.1 Periodic Functions: The Fourier Series

Fourier demonstrated that almost any periodic mathematical function could be expressed as the sum of an infinite number of sine and cosine functions. The series is written as:

$$s(x) = a_0 + (b_1 \cos x + c_1 \sin x) + (b_2 \cos 2x + c_2 \sin 2x) + \dots \quad (\text{Equ. 17.1})$$

where coefficients  $a_0, b_1, b_2, \dots, b_n$  and  $c_1, c_2, \dots, c_n$  scaled the potentially infinite



**Figure 17.3** The basis of the Fourier series is that any periodic function such as a square wave is the sum of a series of sinusoids. The curves shown in white are sinusoids, while their sum—the square wave—is shown in black. In this example, the black curve is the sum of 20 sinusoids.

number of progressively higher-frequency sinusoids to fit the function  $s$ . The only constraints are that the function  $s(x)$  must have a finite number of peaks, valleys, and discontinuities; and the area under the curve must be finite.

The simplest cases are the sine and cosine functions themselves, which require only one term. Combining a few dozen terms can replicate the waveforms produced by musical instruments, even those of the violin, which are rich in overtones. In computing a function using a Fourier series, each point  $x$  in the series generates one point,  $s(x)$ , in the function  $s$ .

The Fourier series produces some remarkable results. Although it seems counterintuitive that a series of rounded sinusoids could produce the right-angled profile of a square wave, the following series generates a square one with a period of  $2\pi$ :

$$s(x) = \cos x - \frac{1}{3} \cos 3x + \frac{1}{5} \cos 5x - \frac{1}{7} \cos 7x + \dots \quad (\text{Equ. 17.2})$$

Bear in mind that it takes an infinite number of terms to produce a true square wave, but with a few dozen terms, the approximation is fairly close. As more terms are added, the approximation to a square wave becomes more and more accurate. Also bear in mind that to add up to a square wave, the sinusoids must line up properly—that is, their phases must be correct. The minus signs in the  $-3x$  and  $-7x$  terms mean their phase is shifted  $180^\circ$ . Each term in the series represents a

## Chapter 17: Images in Frequency Space

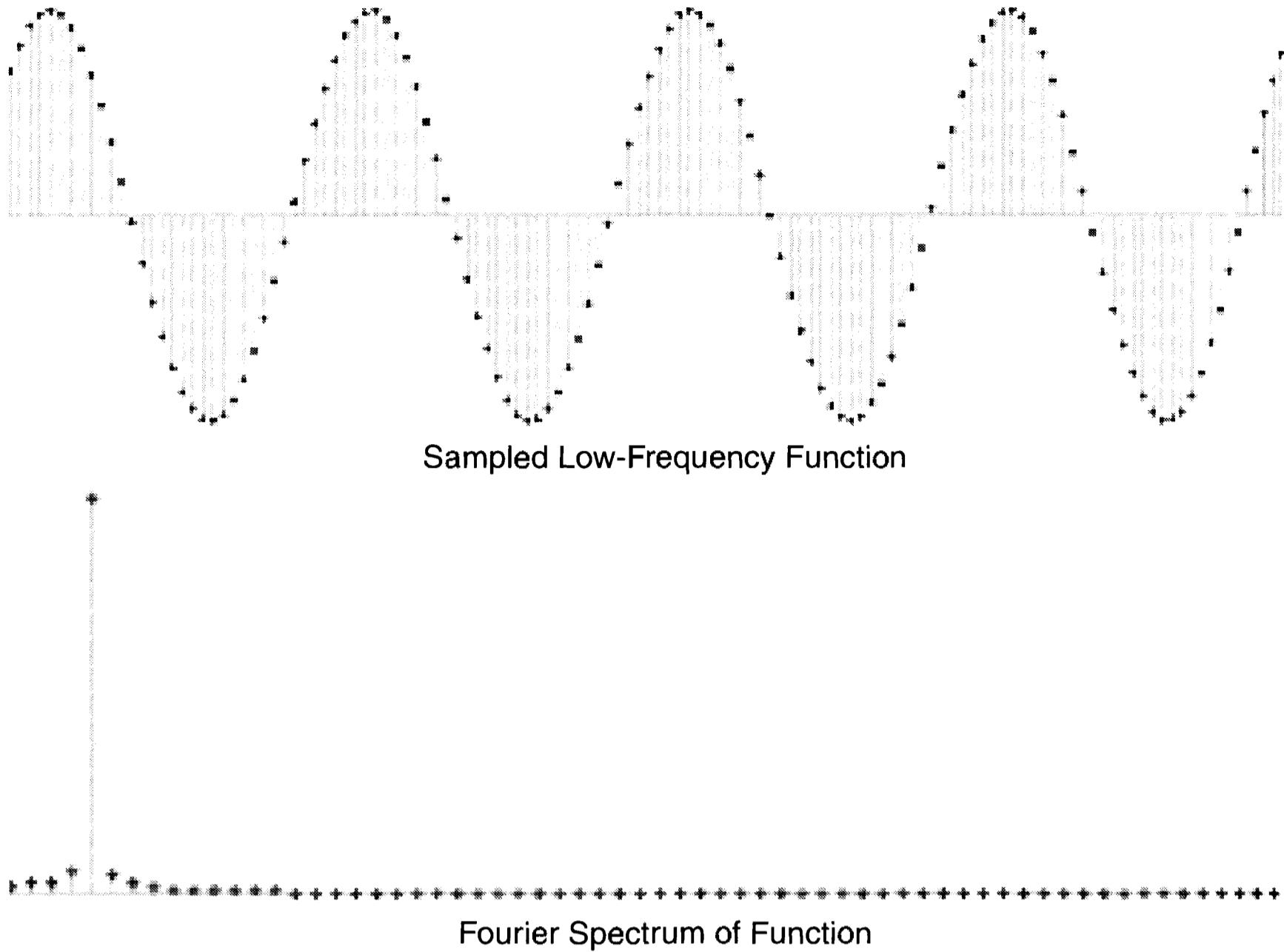


Figure 17.4 The sampled function above could represent a musical tone or one line of pixels from an image. In the spectrum below, low frequencies are to the left and high frequencies are to the right. The energy in the function is concentrated at a single frequency.

point in the frequency spectrum of the square wave.

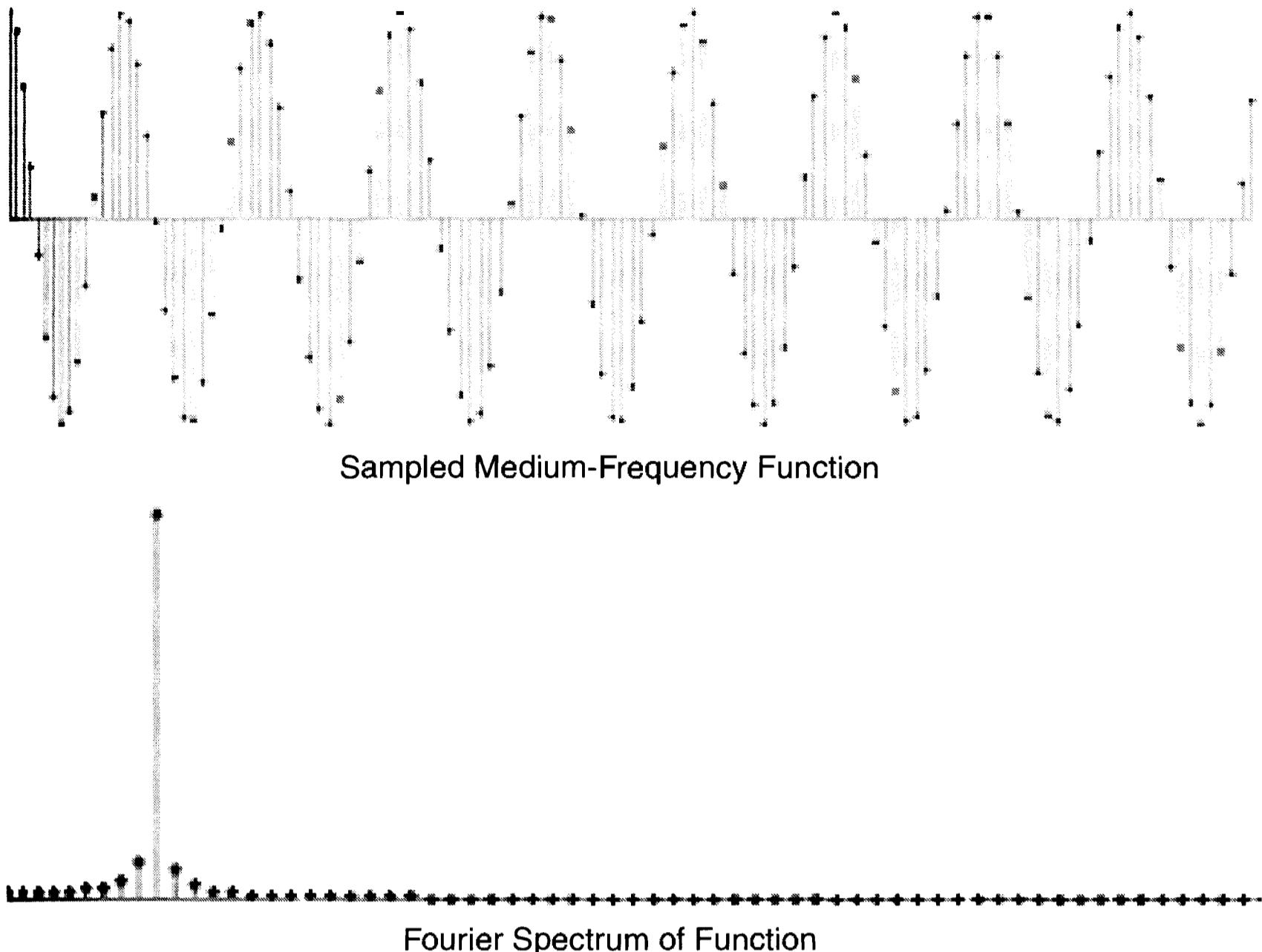
To complete one full cycle of the square wave,  $x$  runs from 0 to  $2\pi$ , so the period of the square wave is  $2\pi$ , and its fundamental frequency  $f_0$  is  $1/2\pi$ . In addition to the fundamental frequency  $f_0$ , the series says that a square wave has harmonics at  $-3f_0$ ,  $5f_0$ ,  $-7f_0$ ,  $9f_0$  and so on—with harmonics at odd multiples of the fundamental frequency. The minus signs on the terms  $-3f_0$  and  $-7f_0$  imply that the phase of these frequencies is shifted  $180^\circ$ , or  $-\pi$  radians.

### 17.2.2 Nonperiodic Functions: The Fourier Integral

In the real world, there are no purely periodic functions. Just as sounds began at some time in the past and will inevitably end at some time in the future, images have a finite extent. At some point in time or space after  $-\infty$  and before  $+\infty$ , the sound or image function dies away.

However, one way of looking at this is that the function is still periodic, but that the period of the function has become infinitely long, which causes the fundamental frequency to become infinitesimally small. If the harmonics are so close together that they become indistinguishable, the spectrum of the function becomes continuous. The Fourier integral is a reformulation of the Fourier series using continuous functions rather than coefficients in a series of terms.

## Section 17.2: Fourier Theory



**Figure 17.5** This function has a shorter wavelength and higher frequency than that shown in Figure 17.4. The higher frequency changes over a smaller span of samples, and the spectrum shows the energy at correspondingly higher frequencies. As the frequency rises, a cycle contains fewer samples.

The Fourier series looks like this:

$$s(x) = a_0 + \sum_{n=1}^{\infty} (b_n \cos(2\pi n f_0 x) + c_n \sin(2\pi n f_0 x)). \quad (\text{Equ. 17.3})$$

This equation is simply Equation 17.1 coming back in summation notion. In the Fourier series,  $n$  runs through an infinite number of terms. The sine and cosine functions can be rewritten in exponential form:

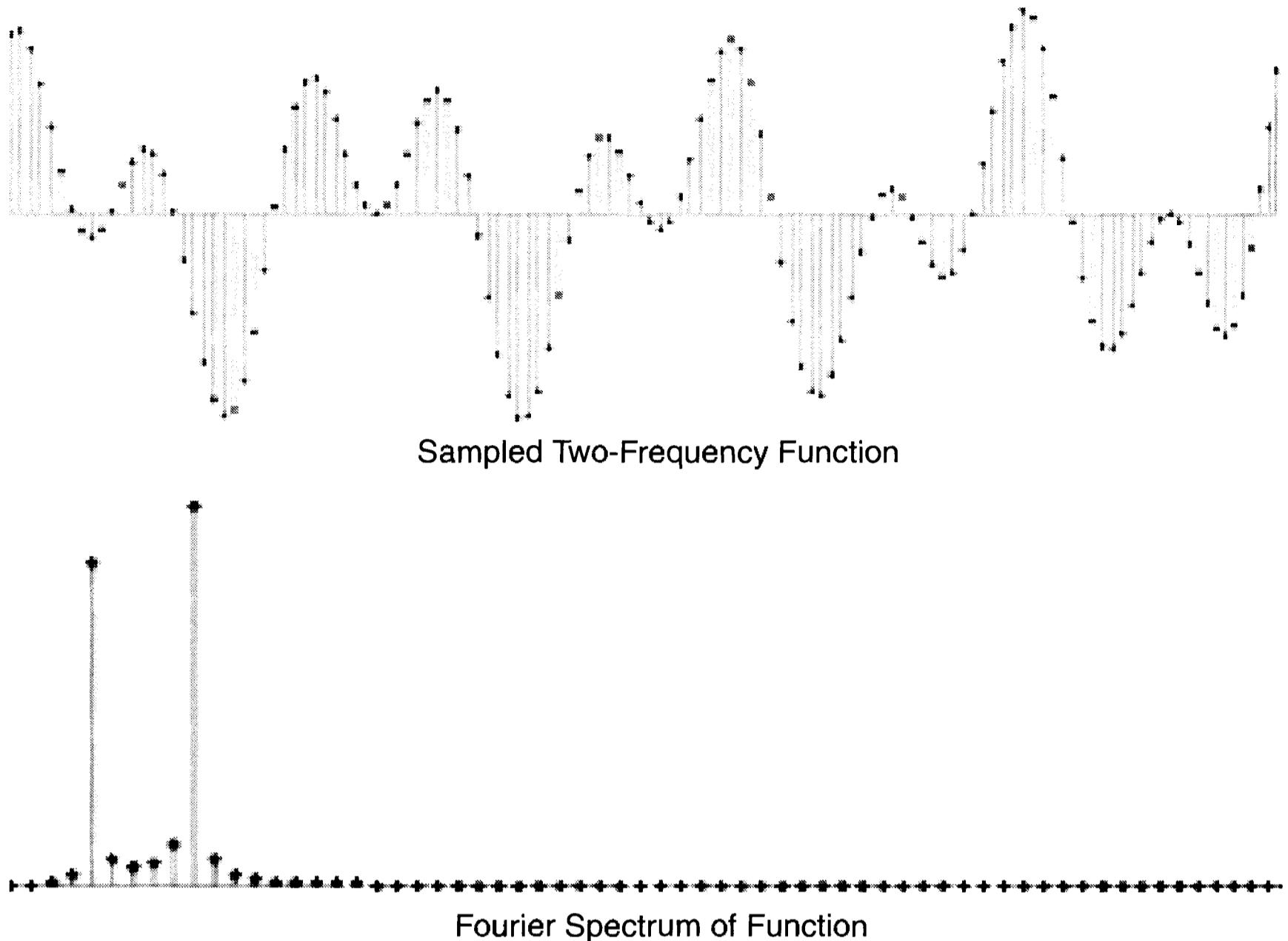
$$\cos(2\pi n f_0 x) = \frac{e^{i2\pi n f_0 x} + e^{-i2\pi n f_0 x}}{2} \quad (\text{Equ. 17.4})$$

and

$$\sin(2\pi n f_0 x) = \frac{e^{i2\pi n f_0 x} - e^{-i2\pi n f_0 x}}{2i} \quad (\text{Equ. 17.5})$$

where  $e$  is the base of natural logarithms, and  $i$  is the infamous  $\sqrt{-1}$ , the square root of minus one. Substituting these into the series equation:

## Chapter 17: Images in Frequency Space



**Figure 17.6** This function is the sum of two sinusoids, the low-frequency one from Figure 17.4 and the medium-frequency one from Figure 17.5. Even with as few as two sinusoids, the eye cannot easily pick out the frequencies—yet the Fourier spectrum clearly reveals the two separate frequencies.

$$s(x) = \sum_{n=-\infty}^{\infty} a_n e^{i2\pi n f_0 x}. \quad (\text{Eqn. 17.6})$$

In this form, the series contains an infinite number of terms. To convert the summation to an integral, we allow  $x$  to become infinitesimally small, which results in the integral form:

$$S(f) = \int_{-\infty}^{\infty} s(x) e^{-i2\pi f x} dx \quad (\text{Eqn. 17.7})$$

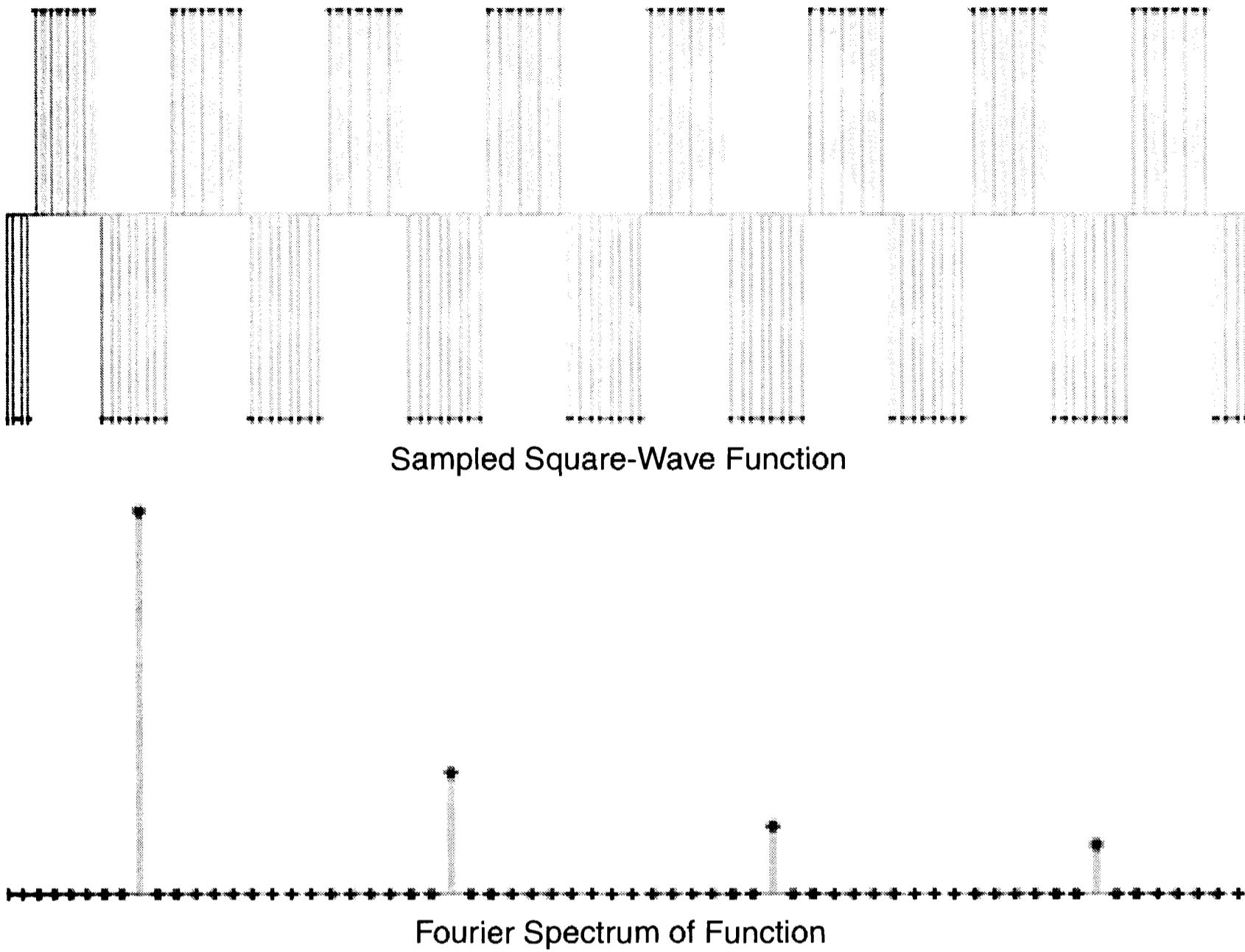
where the function  $S(f)$  is the Fourier transform of  $s(x)$ . If  $s(x)$  is a sound wave, then  $S(f)$  is its frequency spectrum.

The Fourier transform can be derived from the original function using the integral:

$$s(x) = \int_{-\infty}^{\infty} S(f) e^{i2\pi f x} dx. \quad (\text{Eqn. 17.8})$$

These equations have something important to say: *you can break any one-dimensional function—whether it's a sound wave or a slice through an image—into a spectrum; and given the spectrum, you can apply the inverse Fourier trans-*

## Section 17.2: Fourier Theory



**Figure 17.7** A square wave contains a large number of frequency components, increasing in frequency and declining in amplitude. Just as we built a square wave from sinusoids in Figure 17.3, here we have transformed a sampled square wave into a clearly defined set of frequencies.

*form to recover the original function.*

Although these equations apply to one-dimensional functions, it is possible to develop equations for two-dimensional functions; that is, for images. In image processing, we transform an image into frequency space, modify its frequency content, and then apply the inverse transform to recover the modified image.

In the two-dimensional Fourier transform, the image exists in spatial coordinate system  $(x, y)$ , while the spectrum exists in frequency space, with its frequency axes customarily written as  $(u, v)$ . Here is the Fourier transform in two dimensions:

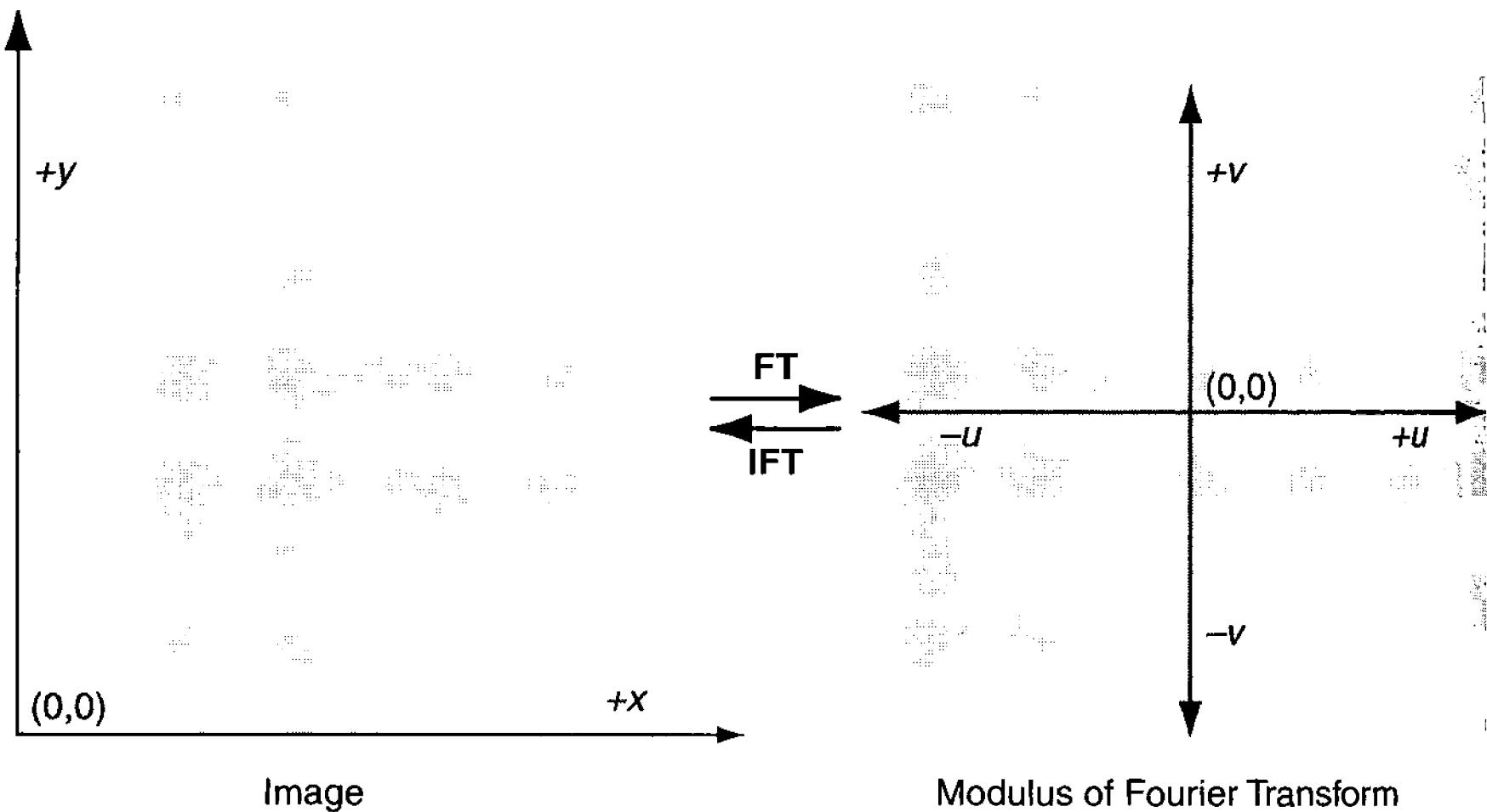
$$S(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(x, y) e^{-i2\pi(ux + vy)} dx dy . \quad (\text{Equ. 17.9})$$

Here is the inverse Fourier transform in two dimensions:

$$s(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S(u, v) e^{i2\pi(ux + vy)} du dv . \quad (\text{Equ. 17.10})$$

The two-dimensional equations show that you can break a two-dimensional image into a two-dimensional frequency spectrum in the  $(u, v)$  plane; and that giv-

## Chapter 17: Images in Frequency Space



**Figure 17.8** Images occupy  $(x, y)$  space, with the origin in the lower left corner, while the forward Fourier transform occupies  $(u, v)$  frequency space with the origin in the center. The Fourier transform converts images to frequency space; while the inverse Fourier transform converts frequency space into images.

en a two-dimensional Fourier transform, you can apply the inverse Fourier transform to generate an image. An image and its spectrum form a Fourier pair, each of which can be transformed into the other.

### 17.2.3 Properties of the Fourier Transform

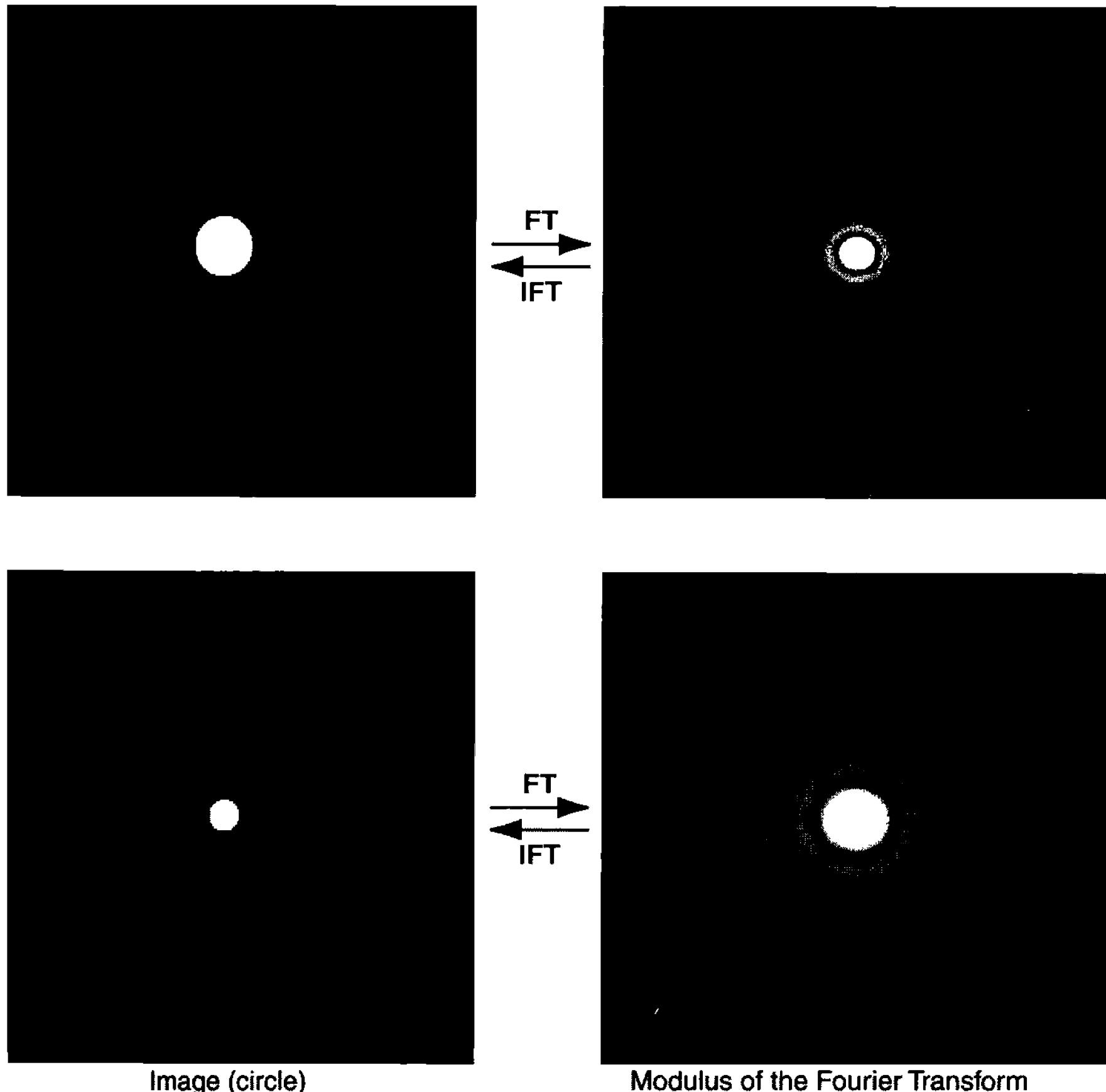
The Fourier transform has properties that make it uniquely useful for processing images. These properties are linearity, symmetry, and inverse scaling. Not only do they apply to the Fourier transform of one-dimensional functions, but also to two-dimensional ones such as sampled images.

**Linearity.** The linearity property says that the sum of the Fourier transforms of the two functions equals the Fourier transform of the sum of the two functions. In other words, if the Fourier transform of  $s(x)$  is  $S(f)$ , and the Fourier transform of  $t(x)$  is  $T(f)$ , then the Fourier transform of  $s(x) + t(x)$  is  $S(f) + T(f)$ .

This reiterates a key point: the Fourier spectrum of frequencies is the sum of the functions in spatial coordinates.

The linearity property also says that the Fourier transform of a constant times a function equals a constant times the Fourier transform of the function. Given a function  $s(x)$  with Fourier transform of  $S(f)$ , the Fourier transform of  $cs(x)$  is  $cS(f)$ . In other words, if we multiply the amplitude of a signal by a factor of  $c$ , the Fourier transform increases by a factor of  $c$ .

**Symmetry.** For real-valued functions such as images, Fourier transforms are symmetrical around zero, so that if you reflect an image about its axis, its Fourier transform stays the same. If the Fourier transform of  $s(x)$  is  $S(f)$ , then the symmetry property says that the Fourier transform of  $s(-x)$  is  $S(f)$ , and the inverse Fourier



**Figure 17.9** The Fourier transform of a circle is a familiar pattern: the diffraction disk and rings. The large “aperture” circle has a low fundamental frequency and produces a compact transform, while the small “aperture” circle has a higher fundamental frequency and produces a larger pattern.

transform of  $S(-f)$  is  $s(x)$ . This is borne out when you see Fourier transforms of images: they are always symmetrical about the origin.

**Inverse Scaling.** If the Fourier transform of  $s(x)$  is  $S(f)$ , then for a real-valued function, the scaling property says that the Fourier transform of  $s(cx)$  is:

$$\frac{1}{|c|} S\left(\frac{f}{c}\right).$$

The inverse scaling property means that if you double the size of an object in an image, the amplitude and frequency of its spectrum are cut in half; in other words, the wavelengths that make up the image (i.e., the size of the image) are the inverses of the frequencies. The inverse relationship is also true: given a Fourier transform  $S(cf)$ , the inverse Fourier transform is:

## Chapter 17: Images in Frequency Space

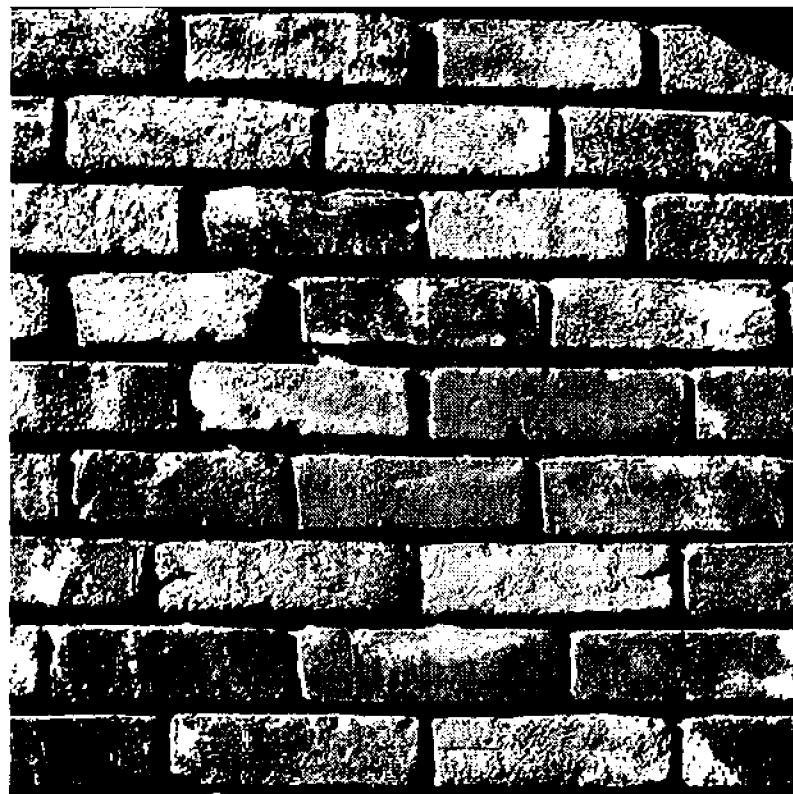
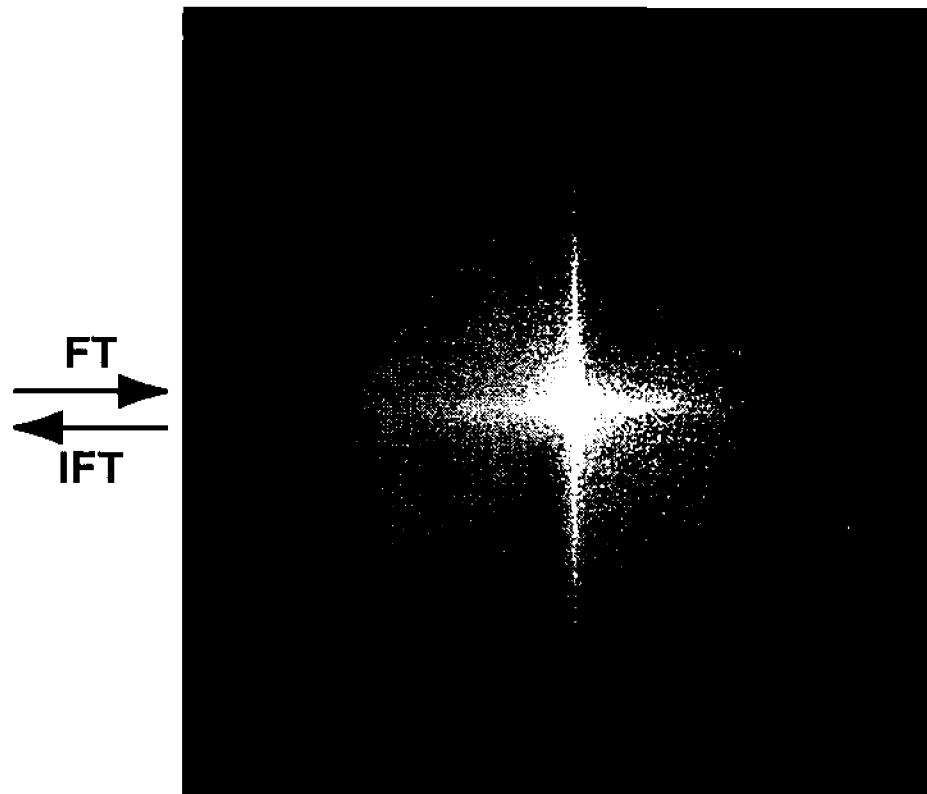


Image of Brick Wall



Fourier Transform of Brick Wall

**Figure 17.10 Case study:** The mortar in a brick wall has a well-defined set of spatial frequencies, visible as bright horizontal and vertical bands in the modulus of the Fourier transform. The diffuse halo in the Fourier transform contains information about the texture on the brick surfaces and mortar.

$$\frac{1}{|c|} s\left(\frac{x}{c}\right).$$

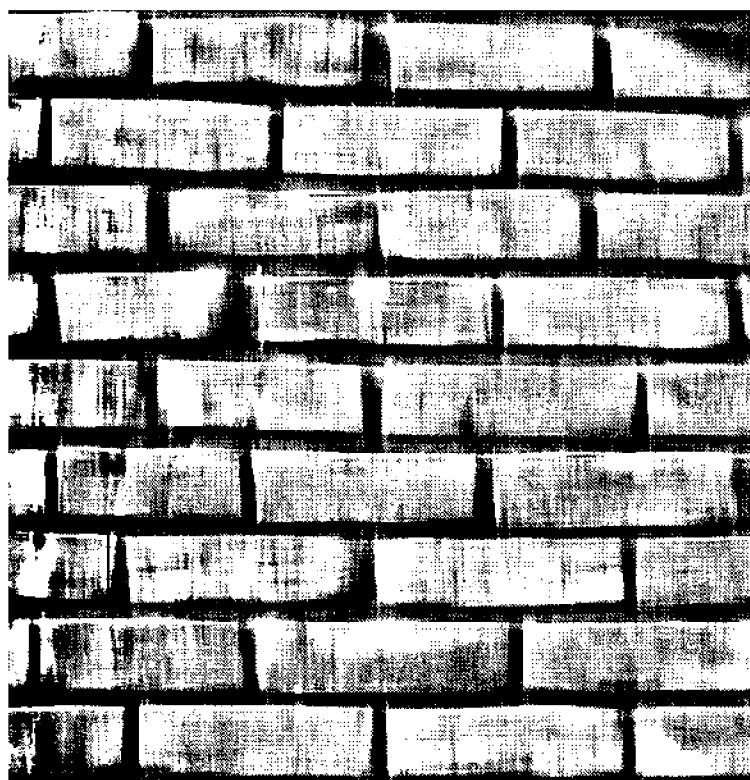
This says that if you double the frequencies in the spectrum of an image, objects in the image become half as large and half as bright.

In Fourier transforms of images, small objects have mainly high-frequency spectra, and large objects have mainly low-frequency spectra. Information about the fine details of an image is contained in the high spatial frequency part of the spectrum around the edges, while information about the overall distribution of light and dark in the image is contained in the very center of the Fourier transform, at the very lowest spatial frequencies.

### 17.2.4 Convolution via Fourier Transform

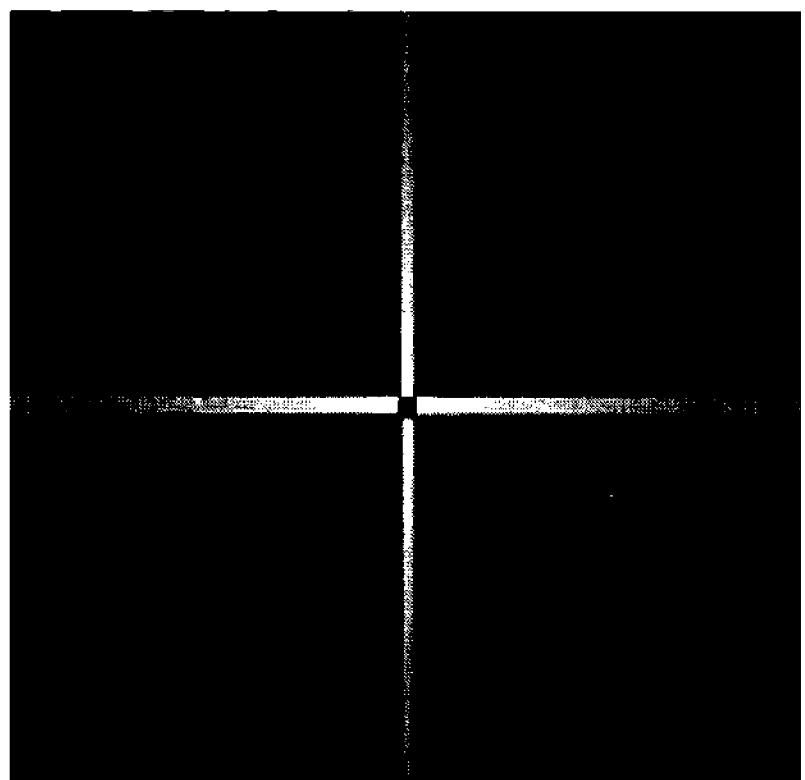
In Chapter 14, on Linear Operators, we saw that one function, such as an atmospheric blur, could modify another function, such as an image, in a process called convolution. The convolution operator served as a weighting function for combining pixels over some region. We alluded to the fact that blur operators removed high spatial frequencies, and that sharpening operators and unsharp masking increased the strength of high spatial frequencies. It is obvious that a link exists between convolution and spatial frequency.

The link is that convolving two functions is the same as multiplying their Fourier transforms. If you take the Fourier transform of a convolution kernel and multiply it by the Fourier transform of an image, then take the inverse Fourier transform of their product, the result is the same as convolving the image and the convolution kernel. Stated symbolically,

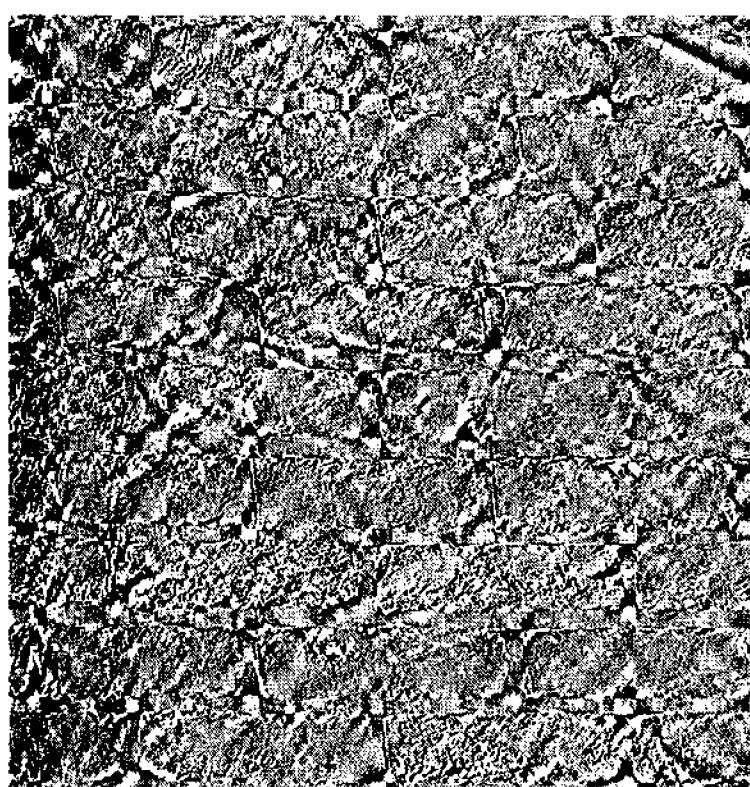


Enhanced Brick Pattern

FT  
↔  
IFT

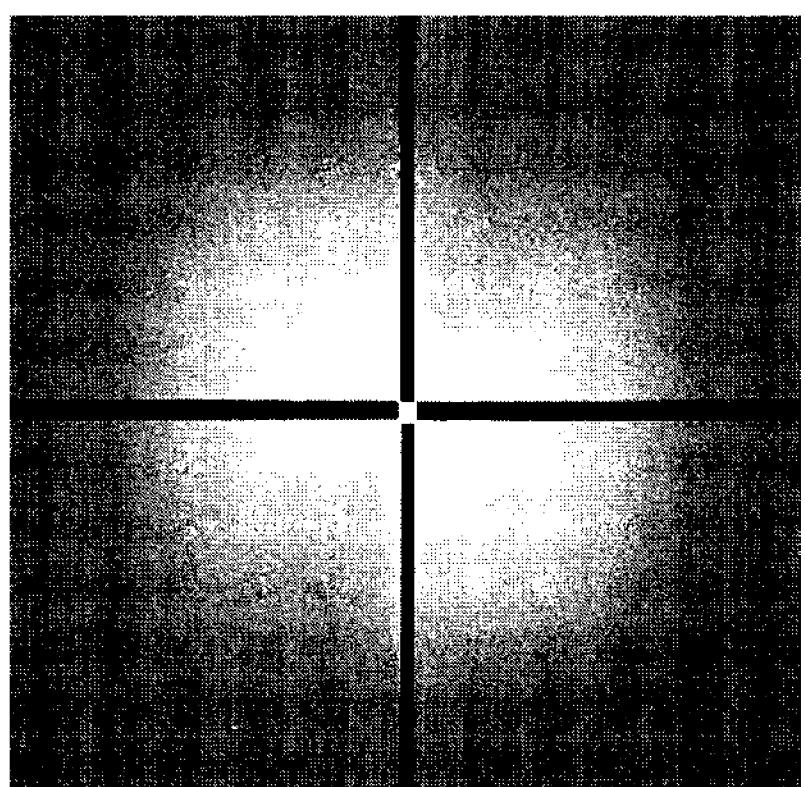


Masked Fourier Transform



Blocked Brick Pattern

FT  
↔  
IFT



Masked Fourier Transform

**Figure 17.11** By passing only the horizontal and vertical bands of frequencies defining the pattern of bricks, the pattern is isolated, and the inverse Fourier transform shows only pattern. However, by blocking those specific frequencies, the pattern is spatially filtered out, leaving only the surface texture of the wall.

The Fourier transform of  $s(x) \otimes t(x)$  is  $S(f)T(f)$

and

The inverse Fourier transform of  $S(f)T(f)$  is  $s(x) \otimes t(x)$ .

For computing small convolution kernels, direct convolution is faster than computing two Fourier transforms, multiplying them, and then computing an inverse Fourier transform. For isolating ranges of spatial frequency, where kernels are large and complex, the Fourier transform route is faster and more flexible.

The convolution property of the Fourier transform establishes that image enhancement by linear operators and by Fourier transform are exactly the same.

The choice between kernel methods and Fourier transform methods comes down to which is easier or faster to compute. For image enhancement with point-

## Chapter 17: Images in Frequency Space

spread functions smaller than 6 pixels radius, linear operators are faster; for complex enhancements over a wide range of spatial frequency—that is, those that would have very large kernels—Fourier methods are faster.

### 17.2.5 Parseval's Theorem

For image processing, the final key property of the Fourier transform is Parseval's Theorem. It states that the total energy in the spatial domain,  $s^2(x)$ , and the total energy in the frequency spectrum,  $S^2(f)$ , must be equal. In other words, the Fourier transform conserves energy in both directions.

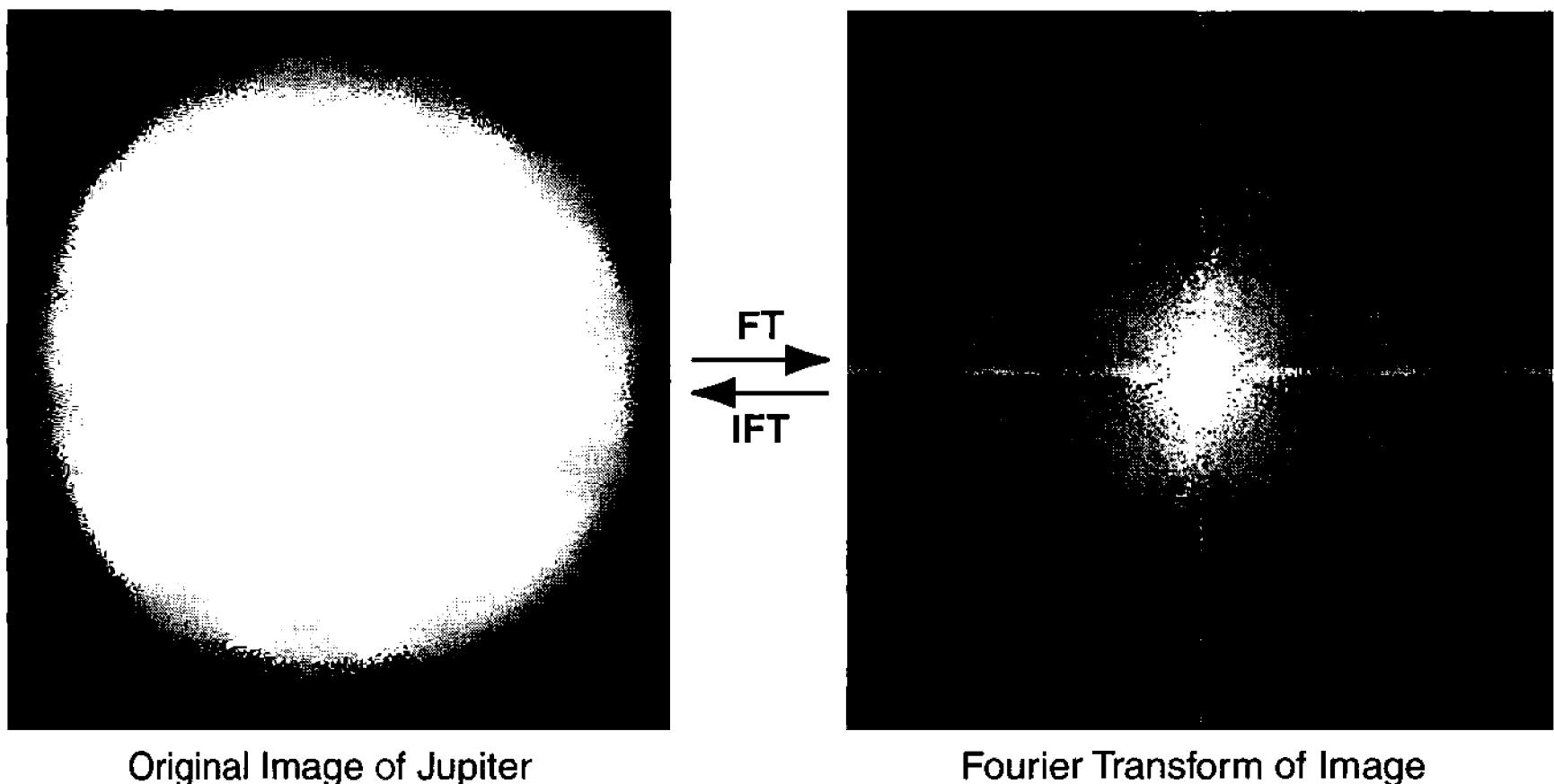
Considered in light of the linearity property, Parseval's Theorem is important, because energy must be conserved during Fourier transforms when different functions are added and scaled. This implies that we can take an image composed of a range of spatial frequencies, apply the Fourier transform, and then plot the energy by squaring the amplitudes of the Fourier frequency components. The energy plot is called the modulus of the transform. The energy we see at each location in the spectrum is the energy contained at some frequency in the image. We also know, from the linearity property, that we can manipulate the frequency components independently.

Accordingly, if we change the modulus by multiplying some group of frequencies, we multiply the energy of those frequencies. Upon performing the inverse Fourier transform, the function will be enhanced at the corresponding wavelengths. Thus, Parseval's Theorem tells us that by altering its two-dimensional Fourier transform, we can alter the image.

Figure 17.11 shows a dramatic case study. In the figure, an image of a brick wall contains an obvious regularly repeating pattern of bricks and mortar. The many frequency components necessary to define the pattern overlap, and produce horizontal and vertical bands in Fourier spectrum. By masking the Fourier spectrum to pass only those components, only the brick and mortar pattern appears in the new image. Nearly all of the detail in the bricks and mortar is absent. An inverse mask blocks only the brick and mortar pattern, and passes the complex detail on the brick and mortar surfaces.

### 17.2.6 The Discrete Fourier Series

In the real world, functions do not range from minus infinity to plus infinity. Just as a sound wave is finite—it began some time after the formation of the universe, and will likely end before the end of the universe—images are finite. In fact, digital images are not only finite, they are only a few hundred or a few thousand samples on a side. Sounds, electronic signals, and images exist within a window of time or the spatial window imposed by the detector. Furthermore, any digital sound, signal, or image on a computer is not a continuous function, but consists of a set of discrete samples. Because the digitized sound wave, digitized electronic waveform, or a digitized image consists of a finite number of samples in a defined window of time or space, it is possible to analyze a discrete (limited) sound or im-



**Figure 17.12** The Fourier transform of this Jupiter image shows distinct features in frequency space. There is a bright low-frequency haze, an extended halo of high frequencies, broadband spikes aligned with the  $u$  and  $v$  axes, and an intense low-frequency spike near the center. Image by Don Parker.

age with a finite number of terms in the series, each mathematical term in the series corresponding to a frequency in the spectrum of the sound or image.

Unlike the infinite problem of computing a Fourier integral, computing a discrete Fourier series is not only feasible, but quite straightforward. For a one-dimensional signal,  $s(n)$ , consisting of  $N$  samples, the discrete Fourier transform is:

$$S(k) = \sum_{n=0}^{N-1} s(n)e^{-i2\pi kn/N} \quad (\text{Equ. 17.11})$$

where  $k$  is the index of an element in the discrete frequency spectrum  $S(k)$ , and  $n$  is the index of an element in the signal,  $s(n)$ .

The inverse transform, which allows you to recover the array of sampled values,  $s(n)$  from their frequency spectrum,  $S(k)$ , is:

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k)e^{i2\pi kn/N}. \quad (\text{Equ. 17.12})$$

The equations above are not used in computing the discrete Fourier transform because the complex exponential is hard to compute. Instead, the exponential term is replaced with its sine and cosine components. The discrete Fourier transform can be computed from:

$$S(k) = \sum_{n=0}^{N-1} s(n) \cos\left(\frac{2\pi kn}{N}\right) - i s(n) \sin\left(\frac{2\pi kn}{N}\right). \quad (\text{Equ. 17.13})$$

The inverse of the discrete Fourier transform can be computed from:

## Chapter 17: Images in Frequency Space

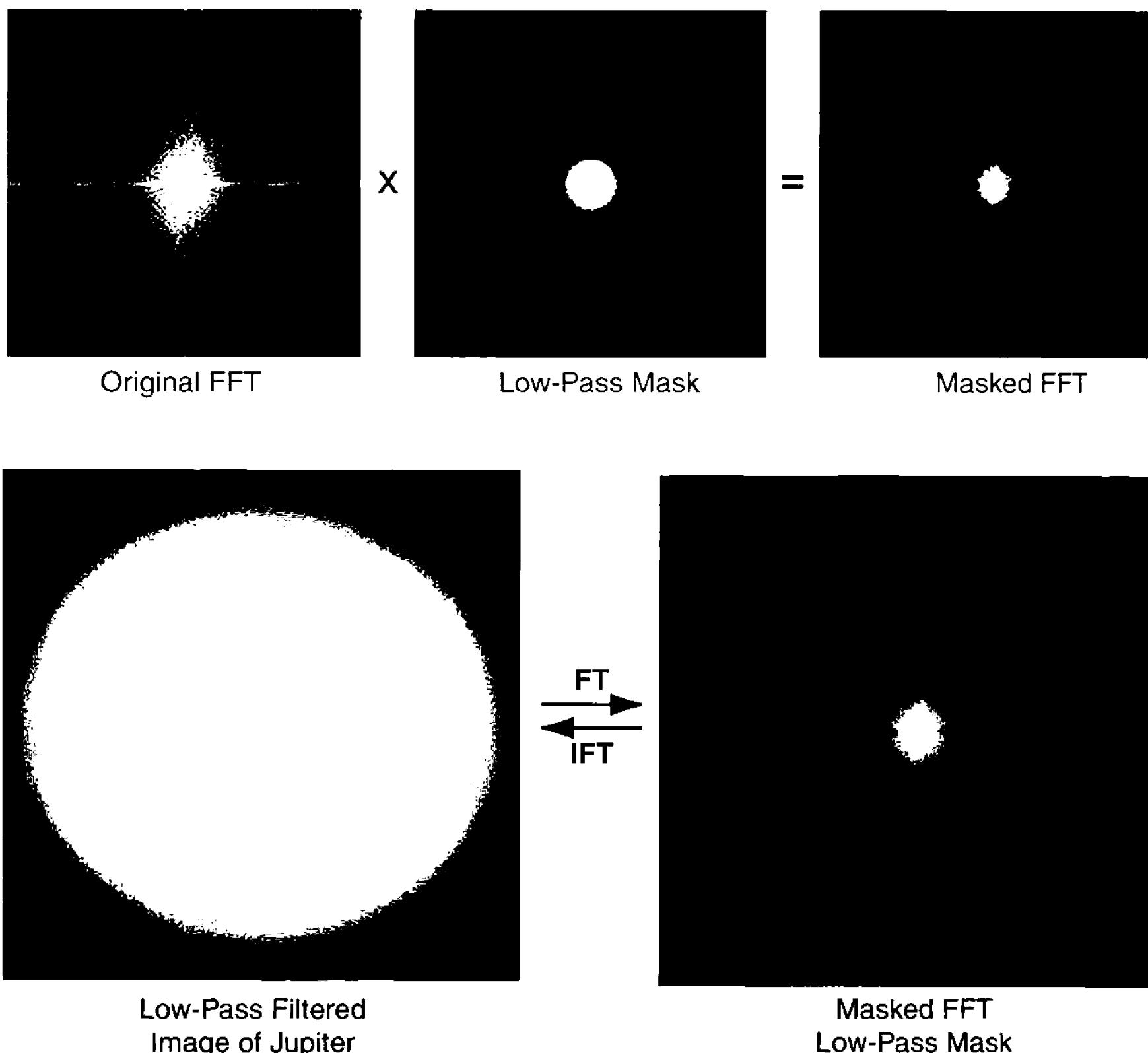


Figure 17.13 Using a low-pass Butterworth filter to isolate the central low-frequency haze produces a softened image of the planet. The original Fourier transform is multiplied by the mask, then converted back to a filtered image by computing the inverse Fourier transform of the filtered Fourier transform.

$$s(n) = \sum_{k=0}^{N-1} S(k) \cos\left(\frac{2\pi kn}{N}\right) - iS(k) \sin\left(\frac{2\pi kn}{N}\right). \quad (\text{Equ. 17.14})$$

The two-dimensional version of the discrete Fourier transform for images requires a double summation, but is similar in form.

The values in the computed Fourier spectrum,  $S(k)$ , are complex numbers; that is, they have both real and imaginary components. Each value consists of a real part, the cosine term, plus an imaginary part, the sine term. Complex numbers are written in the form  $a+bi$ , where  $i^2$  equals  $-1$ , so that  $i = \sqrt{-1}$ . Each complex number represents a frequency in the Fourier transform with an associated amplitude and phase. The amplitude is  $\sqrt{a^2 + b^2}$  and the phase is  $\arctan(b/a)$ .

Note that to compute each term in the transform, you must compute a term for each of the samples; so computing the transform of  $N$  samples requires  $N$ -squared evaluations of the sine and cosine terms. This is not a problem when  $N$  is small, but as  $N$  rises from a few hundred to a few thousand to a few million samples, the com-

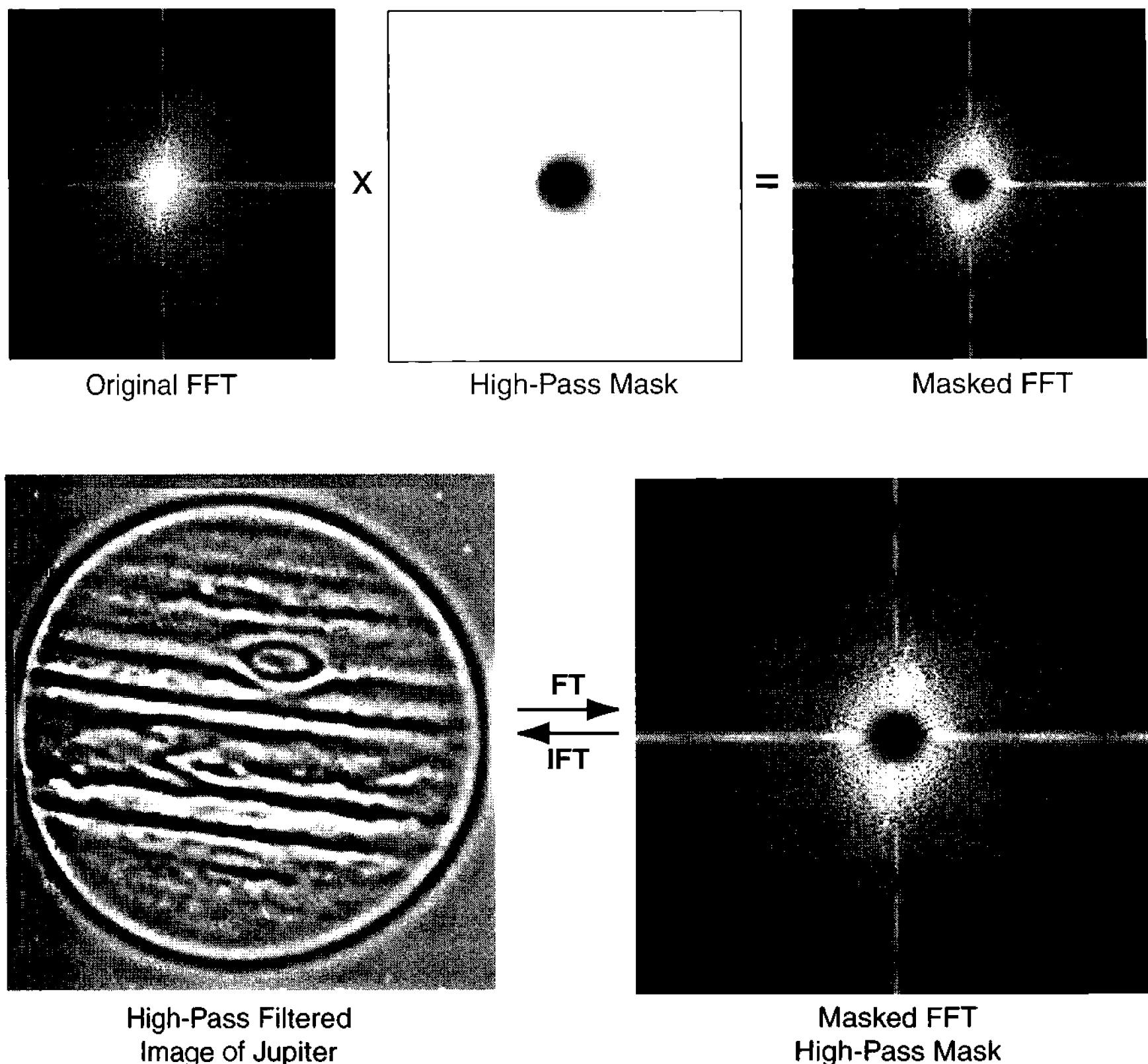


Figure 17.14 A high-pass Butterworth filter blocks low-frequency components, leaving only the detail-bearing high-frequency and edge components. The disk of the planet is a low-frequency object, so that at high frequencies, the center of the planet and the sky are both rendered in the same shade of gray.

puting burden grows very rapidly. Even with today's fast computers, calculating the Fourier spectrum of a large image using the discrete Fourier transform is simply too big a task to complete in a reasonable time.

### 17.2.7 The Fast Fourier Transform

In 1965, J. W. Tukey and P. M. Cooley published a much faster method for computing discrete Fourier transforms for values of  $N$  that are powers of 2. Their method is called the fast Fourier transform, or FFT. The FFT produces exactly the same results as the discrete Fourier transform, but requires only  $N \log_2 N$  calculations rather than the  $N^2$  required for the discrete transform. Computing a discrete Fourier transform with 1,024 elements requires 1,048,000 operations, but using the FFT requires only 10,240 operations—so the job will get done 100 times faster! With larger arrays, the time savings is even greater.

## Chapter 17: Images in Frequency Space

The FFT works by progressively dividing the interval  $N$  into subintervals of length  $N/2$ , and continues subdividing the range until  $N = 2$ , at which point the transform becomes the average of the two values in the interval. Tukey and Cooley's algorithm made computing the Fourier transform of two-dimensional images feasible, with a relatively minor constraint that the image dimensions must be a power of two—64, 128, 256, 512, 1024, and so on. In terms of results, the fast Fourier transform algorithm produces exactly the same result as the discrete Fourier transform.

The values in the computed Fourier transform are complex numbers. In the computer's memory, the real and imaginary components of each point are stored in two arrays. These can be converted into frequency components in a spectrum—with a phase and an amplitude—or used to compute the modulus, or energy at each frequency. Since the total energy in the image must be either zero (no light) or positive (some light), according to Parseval's Theorem, the total energy of the spectrum must also be positive, and, in fact, the energy at every frequency must be either zero (no energy) or positive (some energy). The modulus is therefore always positive.

However, in typical Fourier spectra, the modulus varies over an enormous range—typically greater than 1,000,000 to 1. This is a much greater range than a computer screen can display, so Fourier transforms are usually displayed as the logarithm of the modulus:

$$N(x, y) = \log(1 + S^2(u, v)) \quad (\text{Equ. 17.15})$$

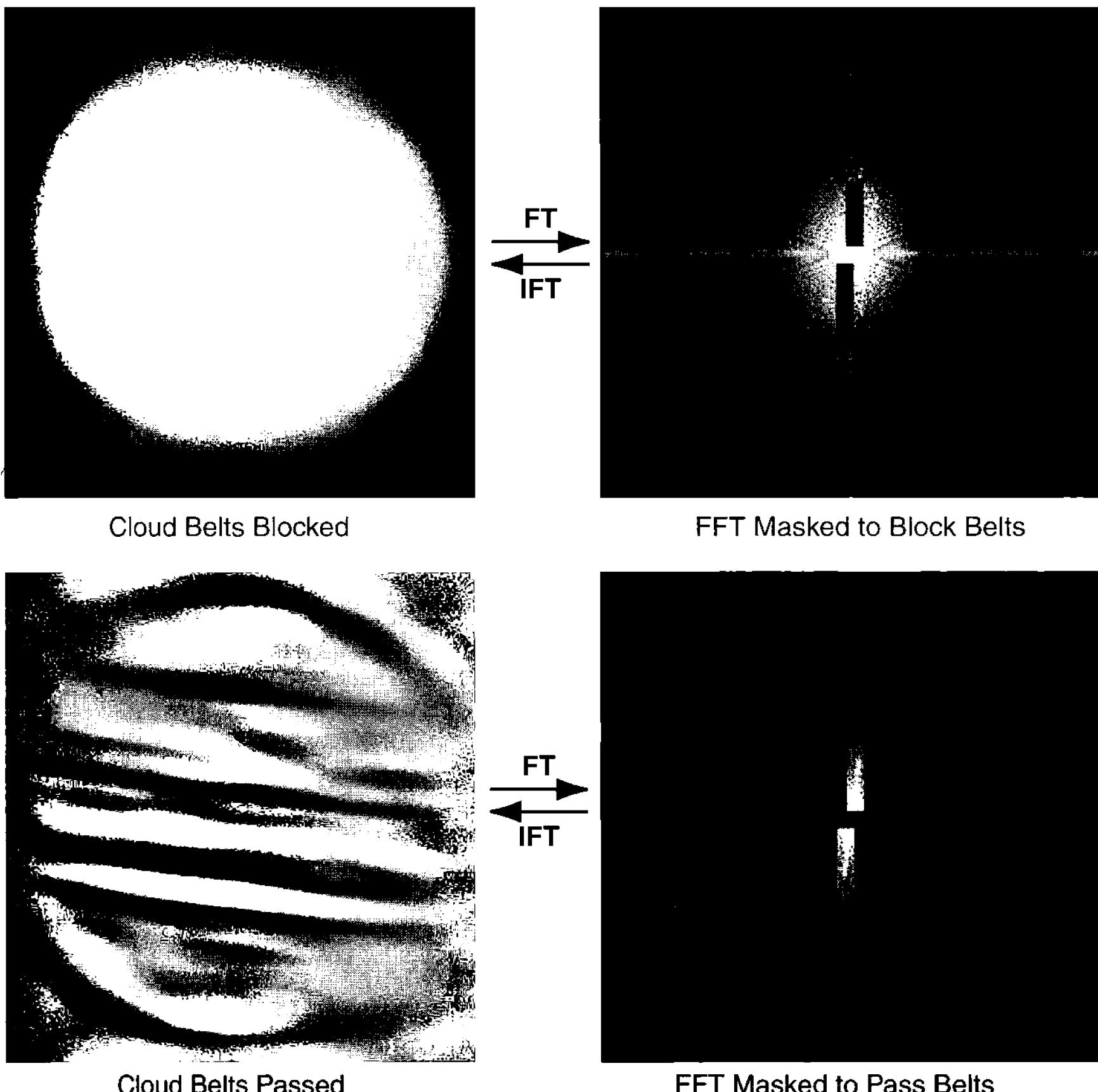
where  $S(u, v)$  is the Fourier transform and  $N(x, y)$  is the new image created to display the modulus. Squaring the Fourier transform yields its modulus, which is always positive. The logarithmic display shows the low-level high-frequency components as well as the strong low-frequency components that usually dominate the Fourier transform of astronomical images.

### 17.3 Image Processing using the Fourier Transform

The Fourier transform gives us the ability to convert images into spatial frequency space and vice versa. This turns out to be a powerful tool for analyzing and enhancing astronomical images, because the spatial frequencies that make up the image are intimately mixed together throughout it. The Fourier transform separates the components, allowing us to understand and manipulate them individually. See Figures 17.10 and 17.11.

Figure 17.10 shows an image of a brick wall. The pattern formed by the bricks is obviously periodic, and therefore rich in well-defined spatial frequencies. The brick and mortar surfaces are very different, composed of small grains of clay and sand, and studded with tiny cracks, holes, and bubbles. We would therefore expect that the brick and mortar surfaces are rich in high-frequency components. Indeed, the Fourier transform shows this to be case. Spatial frequency bands extend along the  $u$  and  $v$  axes, strong at low frequencies and gradually tapering off

## Section 17.3: Image Processing using the Fourier Transform



**Figure 17.15** The intense low-frequency spike is blocked or passed in these images. In the upper one, the spike is blocked, leaving the disk of the planet without cloud belts; in the lower one, the spike is isolated, producing an image consisting of very little except the cloud belts.

toward high ones. The strong pattern is enveloped in a halo of high-frequencies. Note the distinct weakening along an axis at the one o'clock position, in line with the shadow's orientation in the image.

Figure 17.11 shows the effect of masking the Fourier transform to isolate and separate these spatial frequency components. In the top Fourier pair, the transform was multiplied by a mask with pixel values of  $p v_{\max}$  along the  $u$  and  $v$  axes and 0 everywhere else. After multiplying this mask times the transform, only the strongly aligned components remained. The inverse Fourier transform of the masked Fourier transform contains the brick-and-mortar pattern pretty much devoid of surface detail.

The bottom Fourier pair shows the effect of blocking the strong components along the  $u$  and  $v$  axes. The inverse Fourier transform now shows the wall with its

## Chapter 17: Images in Frequency Space

semi-random high-frequency details of the clay and sand surfaces, without the brick-and-mortar pattern. A fairly faint checkerboard pattern remains, generated by the low frequencies in the mask itself. You can also see an outer region of smeary high frequency signals; these are due to the CCD characteristics and noise in the electronics of the digital camera used to make the picture of the wall.

- **Tip:** *Efficient fast Fourier transform and inverse fast Fourier transform are built into **AIP4Win**. They work with any image with a power-of-two size. For other images, use the Float tool to put the image into the middle of an image with power-of-two dimensions. **AIP4Win** displays the logarithm of the modulus of the Fourier transform as an image that you can modify.*

### 17.3.1 Butterworth Spatial Frequency Filters

Fourier transforms make it possible to manipulate spatial frequencies directly in the frequency domain, then create a new image from the modified frequency spectrum. Perhaps the most flexible general-purpose filter for spatial frequencies is the Butterworth filter. By choosing several simple parameters, you can easily generate a Butterworth filter to pass low frequencies, to pass high ones, to pass a band of frequencies, or to block a band.

A spatial filter such as a Butterworth filter, is simply an image that will be multiplied with the Fourier transform. When a pixel value of zero in the filter is multiplied times the Fourier transform, it creates a zero pixel value in the new Fourier transform—completely blocking that frequency. Other pixel values act to multiply the Fourier transform by the factor  $pv / pv_{\max}$ . When the pixel value in the filter is  $pv_{\max}$ , the Fourier transform passes through the filter unchanged.

The equation for the low-pass Butterworth filter is:

$$B_{\text{Low-Pass}}(u, v) = \frac{1}{1 + \left(\frac{\sqrt{u^2 + v^2}}{d_0}\right)^{2\gamma}} \quad (\text{Equ. 17.16})$$

where  $B_{\text{LowPass}}(u, v)$  is the value of the filter at location  $(u, v)$  in the spatial frequency plane,  $d_0$  is the cutoff frequency, and  $\gamma$  is the order of the filter, taking positive values of 1, 2, 3, etc. The low-pass Butterworth filter transmits 100% at the center (low spatial frequencies), 50% at distance  $d_0$  from the center (middle spatial frequencies), and 0% at the outer edge (the highest spatial frequencies in the image). The rapidity of the transition from 100% to 0% is controlled by the order,  $\gamma$ . The higher the filter order, the more abrupt is the filter cutoff.

To use the filter, each element in the Fourier transform array is multiplied by the corresponding value in the filter:

$$S'(u, v) = B(u, v)S(u, v) \quad (\text{Equ. 17.17})$$

where  $S'(u, v)$  is the modified Fourier transform and  $B(u, v)$  is the filter.

### Section 17.3: Image Processing using the Fourier Transform

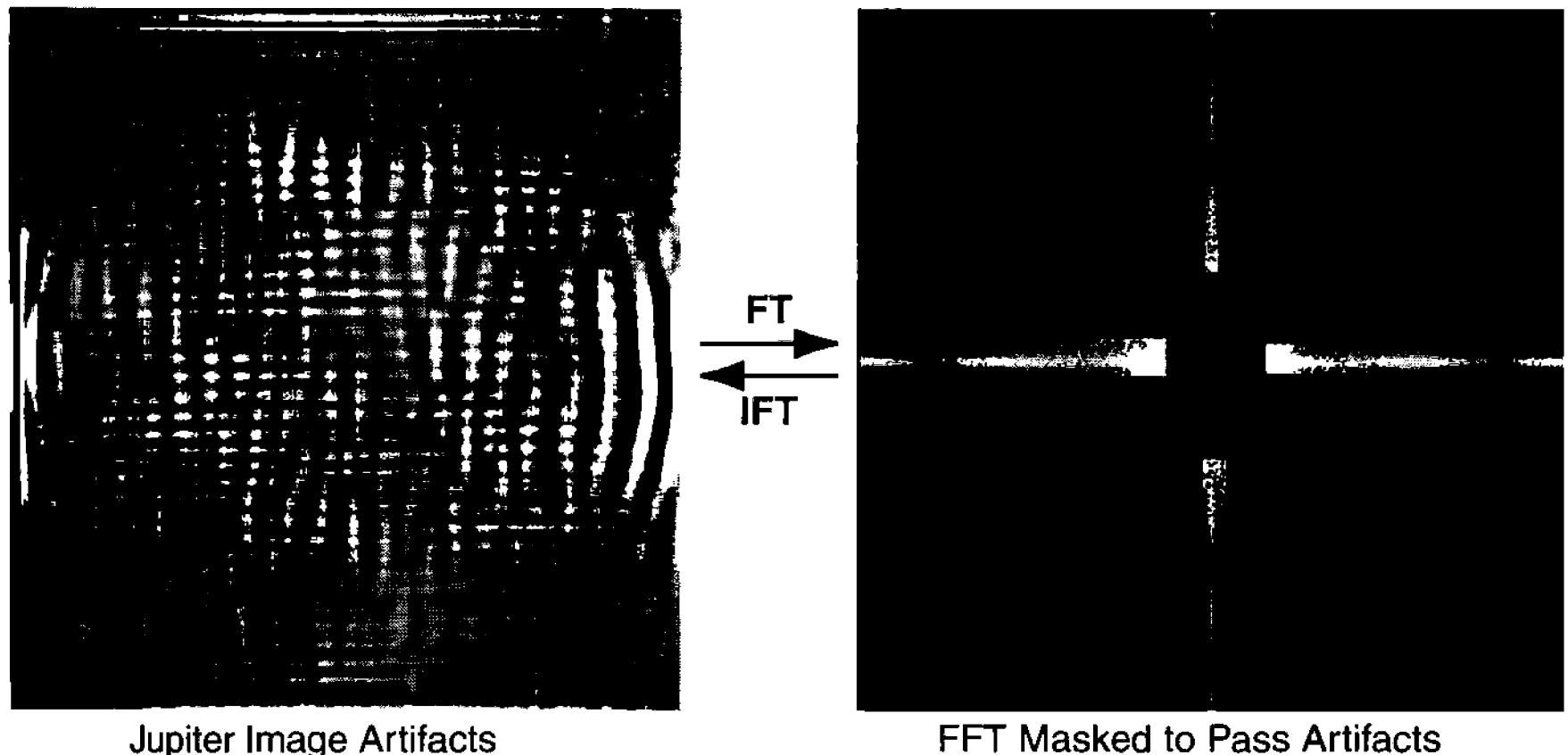


Figure 17.16 The prominent broadband features in the Fourier transform turn out to have been generated from artifacts created when the original image ( $192 \times 165$  pixels) was centered and resampled to  $256 \times 256$  pixels. Although it is normally not visible, resampling leaves traces in the Fourier spectrum.

In a computer, the Fourier transform is stored in two arrays, one containing the real components and the other the imaginary ones. The multiplication must be carried out for both components:

```

FOR v = 0 to vmax
    FOR u = 0 to umax
        filter = butterworth(u, v)
        real(u, v) = filter * real(u, v)
        imag(u, v) = filter * imag(u, v)
    NEXT u
NEXT v

```

where `real()` is the array of real numbers, `imag()` is the array of imaginary numbers, and `butterworth()` is the filter array.

Figure 17.13 shows the effect of low-pass filtering: it acts to blur the image. The high frequencies that create fine detail in the image are stopped, leaving only the low frequencies' contribution to it. Since this image is low in high frequencies to begin with, the blurring does not appear very dramatic.

The equation for a high-pass Butterworth filter is:

$$B_{\text{High-Pass}}(u, v) = \frac{1}{1 + \left(\frac{d_0}{\sqrt{u^2 + v^2}}\right)^{2\gamma}} \quad (\text{Equ. 17.18})$$

where the variables all have the same meanings. The high-pass Butterworth filter transmits 0% at the center, 50% at distance  $d_0$  from the center, and 100% at the outer edge. The parameter  $\gamma$  controls the abruptness of the transition from passing to blocking.

## Chapter 17: Images in Frequency Space

You can see the effect of a high-pass filter on an image of Jupiter in Figure 17.14. Because the zero-frequency component is absent, the center of the planet appears the same average brightness as the surrounding black sky. It is clear that high-frequency components define the edges of the belts, zones, and Great Red Spot.

To block a band of frequencies, use a Butterworth band-stop filter:

$$B_{\text{Band-Stop}}(u, v) = \frac{1}{1 + \left( \frac{d_w \sqrt{u^2 + v^2}}{(u^2 + v^2) - d_0^2} \right)^{2\gamma}} \quad (\text{Equ. 17.19})$$

where the variables mean the same as above, and  $d_w$  defines the band-width that the filter blocks. A Butterworth band-stop filter transmits 100% at the center and 100% at the edge, and 0% at frequency  $d_0$ . The transmission at the edges of the blocked band of frequencies is 50%.

The complement of a band-stop filter is a band-pass filter:

$$B_{\text{Band Pass}}(u, v) = 1 - \frac{1}{1 + \left( \frac{d_w \sqrt{u^2 + v^2}}{(u^2 + v^2) - d_0^2} \right)^{2\gamma}}. \quad (\text{Equ. 17.20})$$

A Butterworth pass-band filter blocks at the center and at the edge, but transmits 100% at frequency  $d_0$ . The transmission at the edges of the transmission band of frequencies is 50%. Band-pass and band-stop filters are good for highlighting or blocking textures and patterns characterized by a well-defined size scale.

- **Tip:** *AIP4Win has a built-in Butterworth filter tool. Select the size of the cutoff filters and the bandwidth of the band-pass and band-stop filters, and AIP4Win makes the filter to your specifications. The filter must have the same power-of-two dimensions as the image.*

### 17.3.2 Feature Masking in Frequency Space

Image features sometimes have well-defined frequency characteristics, even when they are spread around an image. When this happens, an observer can create a special mask to either pass or block the defining frequencies. The mask must be designed to match the pattern in the Fourier transform, so it will have a size and shape determined by the image itself.

The images of Jupiter in Figures 17.14 and 17.15 show simple feature-passing and feature-blocking masks. Blocking areas are set to a pixel value of zero, and passing areas are set a pixel value of  $p v_{\max}$ ; the Fourier transform is multiplied by the mask to remove the frequency bands; and finally the masked Fourier transform is inverse Fourier transformed back into an image.

## Section 17.3: Image Processing using the Fourier Transform

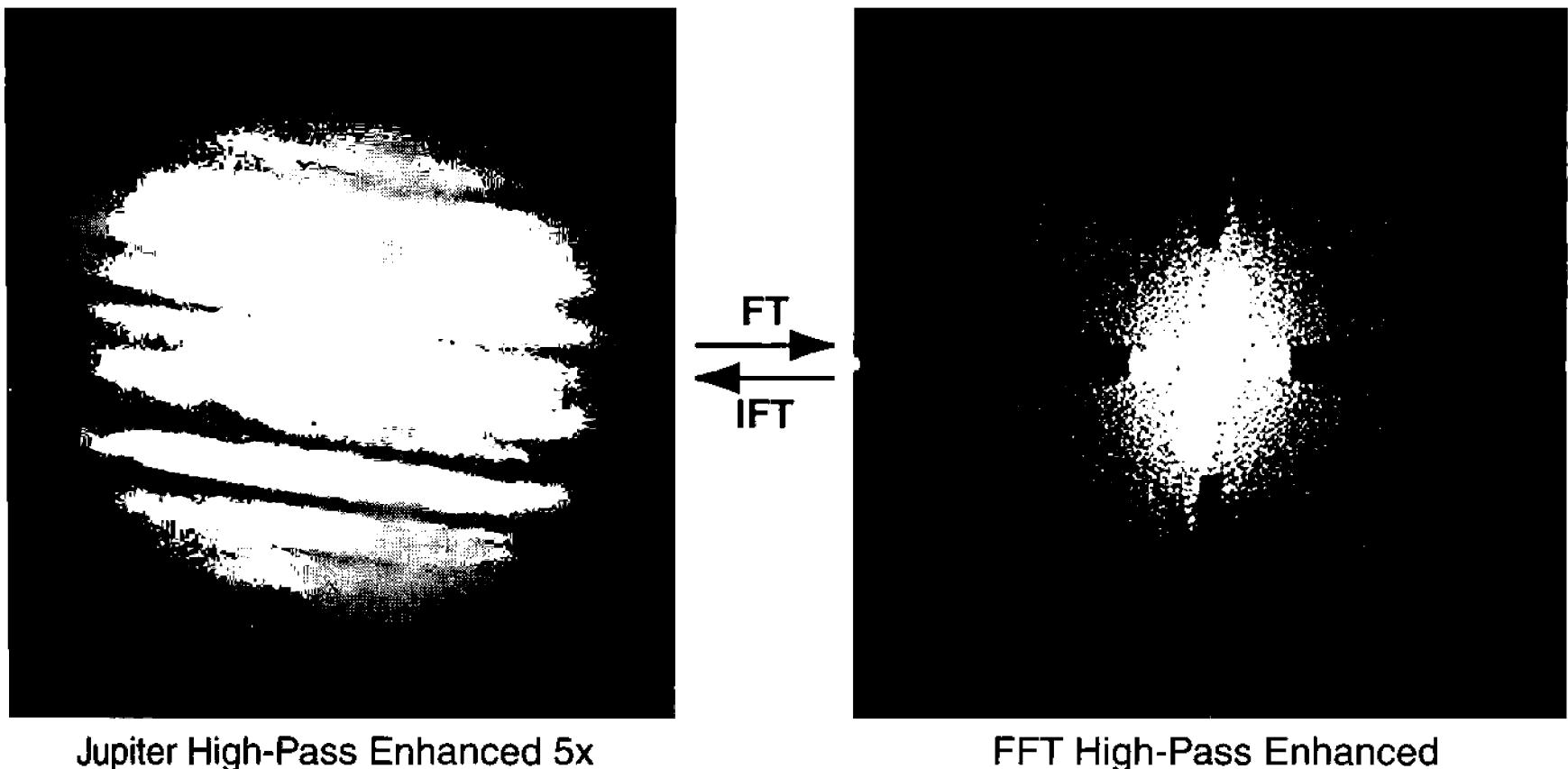


Figure 17.17 With its low frequencies passed intact and its high frequencies enhanced by a factor of five, Jupiter reveals cloud structures that are all but invisible to the eye in the original image. The artifact components have also been blocked, yielding an exceptionally “clean” image of the giant planet.

- **Tip:** To selectively attenuate spatial frequencies using **AIP4Win**, duplicate the modulus image of the Fourier transform. Use it to create a mask using the region-filling function of the Pixel Editor tool, and set the frequencies you want to block to zero.

### 17.3.3 Feature Enhancement in Frequency Space

Parseval’s Theorem holds the key to practical image enhancement using Fourier transform techniques. The theorem states that the energy is conserved between an image and its Fourier transform. Blocking spatial frequencies removes energy from the image by reducing the brightness of the image at those frequencies. However, filters can do much more than block spatial frequencies.

Since the pixel values in a Butterworth filter range from 0 to 1, when a spectrum is multiplied by the filter, it is weakened. However, if a spatial frequency in the spectrum is multiplied by a number greater than 1, Parseval’s theorem suggests that the corresponding spatial frequencies in the image will be boosted—and this can be used to enhance desirable frequencies.

For ease of application, the filter is scaled by adding a base transmission factor,  $b$ , and a contrast factor,  $c$ . The filter can thus be adjusted to pass any desired fraction of the original Fourier transform and boost the desired range of spatial frequencies:

$$B'(u, v) = b + cB(u, v). \quad (\text{Equ. 17.21})$$

In the above,  $B'(u, v)$  is the new value for the adjusted filter,  $b$  is the base transmission, and  $c$  is the contrast enhancement.

In computer code, the multiplication is carried out for both components:

```
FOR v = 0 to vmax
```

## Chapter 17: Images in Frequency Space

```
FOR u = 0 to umax
    boost = b + c * filter(u, v)
    real(u, v) = boost * real(u, v)
    imag(u, v) = boost * real(u, v)
NEXT u
NEXT v
```

The image of Jupiter in Figure 17.17 is an enhanced one created by passing 100% of the original image and boosting high spatial frequencies by a contrast factor of five. The resampling and centering artifacts were blocked to produce a final clean image.

- **Tip:** When you apply a filter to a Fourier transform in **AIP4Win**, you can set the Base Value and Contrast. The default values are 0 and 1, values that filter without modification. To enhance images, the base should be between 0 and 1, and the contrast should be greater than 1.

---

# 18 Wavelets

---

Wavelets are a relatively recent invention, a hybrid between the spatial methods of operators and kernels, and the frequency-based methods of the Fourier Transform. In many ways wavelet image processing offers the best of both worlds. In this chapter, we'll explore one method that uses wavelet analysis to enhance detail and two methods that use wavelet analysis to reduce image noise.

So—what is a wavelet? The formal definition—a function with positive and negative values locally, zero everywhere else, and an integral of zero—does little to illuminate the matter. However, a graph of a wavelet is more revealing:



this wavelet has a positive region closely flanked by negative regions, but elsewhere its value is zero. It resembles a ripple on a smooth pond, which originally suggested the name.

This particular wavelet is the “Mexican Hat function,” so called because of its sombrero appearance. Integrating from  $+\infty$  to  $-\infty$  yields zero. Suppose that you applied a Mexican Hat function as a convolution kernel—what would happen? The answer, not surprisingly, is that it would act as a bandpass filter. The skinny positive core acts as a low-frequency high-pass filter, while the wide negative wings act as a high-frequency low-pass filter. Between them, the two filters isolate a narrow band of spatial frequencies.

Wavelet analysis uses functions like the Mexican Hat to split an image into different *wavelet scales* or *levels*, each corresponding to a specific band of spatial frequencies. As does convolution, wavelets operate locally on an image; and as the Fourier transform does, wavelets allow you to process images in the frequency domain. In this chapter, we shall see how this hybrid between the spatial domain and frequency space produces an powerful image processing tool for astronomy.

## 18.1 The Wavelet Transform

The most efficient way to compute the wavelet transform is to use a function like

## Chapter 18: Wavelets

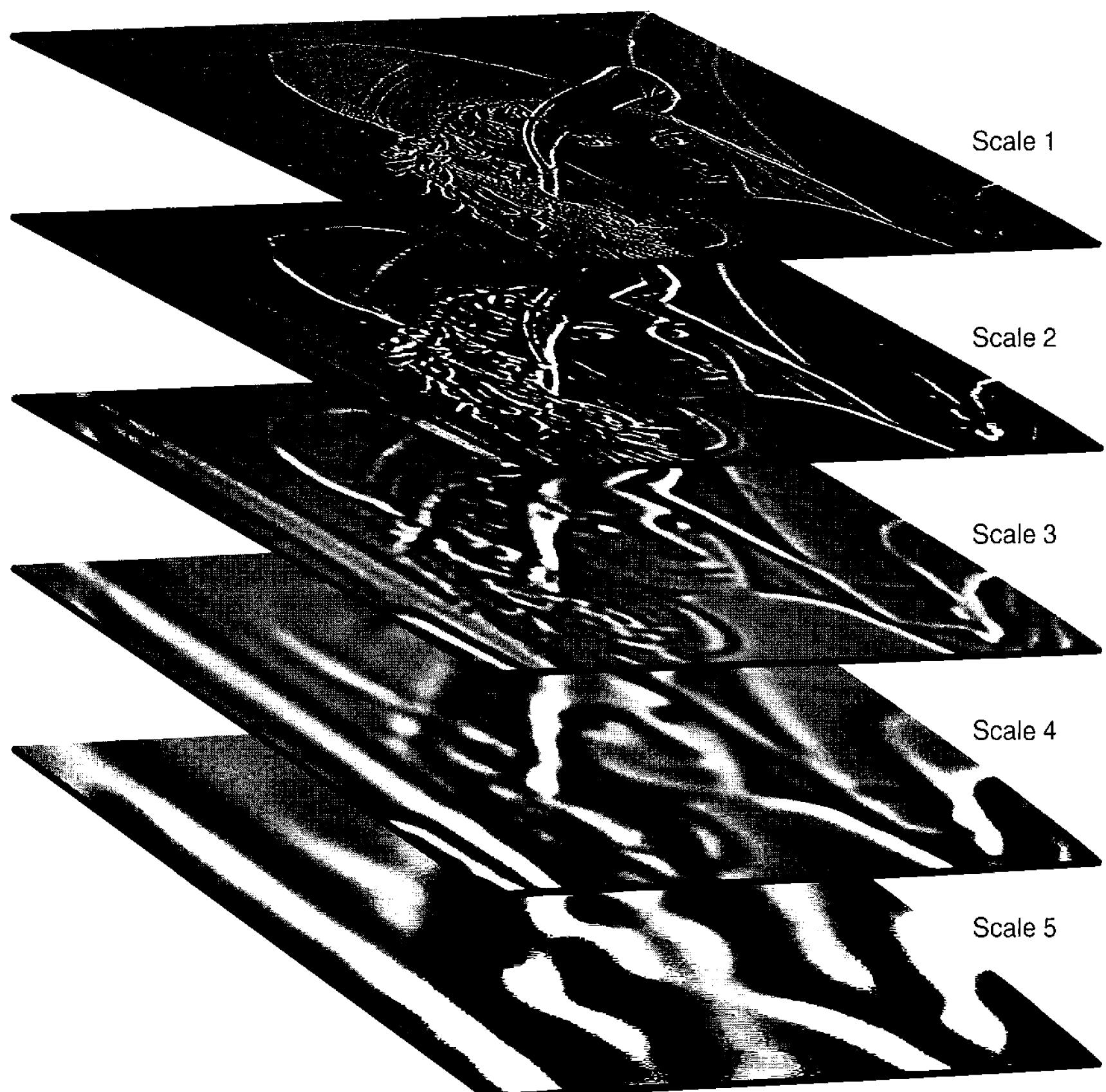


Figure 18.1 Five wavelet scales reveal different spatial structures in the classic Lena image. Scale 1 shows small spatial structures with a kernel radius of 1; the radius of the Scale 2 is 2, that of Scale 3 is 4, and so on—each wavelet scale doubling the radius of the preceding scale and showing large structures.

the Mexican Hat transform to isolate each wavelet scale (i.e., each band of spatial frequencies) from the original image in turn. For astronomy, the *à trous* algorithm is particularly effective, and it has the added advantage of being computationally fast and efficient. The name of the algorithm—*à trous*—means “with gaps,” since, as we shall see, at the higher wavelet scales, the wavelet convolution kernel has big holes in it.

The *à trous* wavelet transform builds wavelet scales iteratively. It begins by applying a  $3 \times 3$  high-pass filter to the original image to form Scale 1. This level contains all of the small-scale high-frequency components in the image, and because the sum of elements in the high-pass filter is zero, the mean value of Scale 1 is also zero. The *à trous* procedure then subtracts Scale 1 from the original image, leaving a residual image free of small-scale high-frequency components.

The algorithm now iterates by applying a high-pass filter “with gaps” to the

## Section 18.1: The Wavelet Transform

residual image to create Scale 2. This filter is diluted by placing its outer elements two pixels from the center. The procedure then subtracts Scale 2 from the residual image, creating a new residual free of both scale 1 and Scale 2 components.

The algorithm iterates again, applying an even more dilute high-pass filter in which the outer elements lie four pixels from the center, creating Scale 3. It then subtracts Scale 3 from the residual to make a new residual.

At each following step, the radius of the high-pass filter doubles: 8, 16, 32, 64, 128, 256, subtracting the next larger structure from the image. Successive wavelet scales contain progressively coarser structures, as shown by the example in Figure 18.1. By Scale 8, the residual contains very little structure. However, when the residual and all the wavelet scales are added together, they recreate the original image exactly.

Because the mean of each scale removed from the original is zero, the total pixel value of the residual remains the same as the original image. This has the interesting consequence that—without making the image darker or lighter—we can multiply, divide, or threshold wavelet scales to enhance, control, or remove image features based on their spatial frequency content.

### 18.1.1 The Wavelet Function

The Mexican Hat function is a continuous function equal to the difference between a positive Gaussian function and a negative Gaussian function of twice the width and half the value. The smallest equivalent one-dimensional convolution kernel is:

$$[-0.25 \ 0.50 \ -0.25]. \quad (\text{Equ. 18.1})$$

The smallest equivalent two-dimensional wavelet kernel can be formed as the convolution of the row kernel with the column kernel:

$$\begin{bmatrix} -0.25 & 0.50 & -0.25 \end{bmatrix} \otimes \begin{bmatrix} -0.25 \\ 0.50 \\ -0.25 \end{bmatrix} \quad (\text{Equ. 18.2})$$

Because this is a separable kernel, convolution of the image to produce wavelet scales is fast and efficient.

### 18.1.2 Properties of the *A Trous* Wavelet Transform

Understanding the properties of the wavelet transform is the key to using it effectively. Here is the nomenclature:

- The original image is  $S_0$ , and a pixel in image  $S_0$  at location  $(x,y)$  is  $S_0(x, y)$ .
- The wavelet kernel is  $\varphi$ , containing elements  $\varphi(i, j)$ .

## Chapter 18: Wavelets

- The wavelet transform associated with  $S_0$  is  $w$ , and the  $j$ -th scale in  $w$  is called  $w_j$ .
- The wavelet transform for a two-dimensional image is a three-dimensional array of *wavelet coefficients*.
- We designate the wavelet coefficient at location  $(x,y)$  in the  $j$ -th scale of the wavelet transform as  $w(x,y,j)$ .

To create the wavelet transform  $w(x,y,j)$ , the *à trous* procedure operates iteratively as follows:

$$w(x,y,j+1) \leftarrow \varphi(i,j) \otimes w(x,y,j) \quad (\text{Equ. 18.3})$$

during which kernel elements  $\pm i$  and  $\pm j$  act on the  $\pm(x \pm 2^j)$  and  $\pm(y \pm 2^j)$  element of wavelet scale  $w_j$ . Each successive scale therefore samples its scale at twice the size, half the spatial frequency, and half the resolution of the previous scale; the factor-of-two step from one scale to the next is called *dyadic*. Values of  $j$  typically run from 1 to 8, over a 1 to  $2^8 = 256$  range of resolution. Wavelet coefficients in  $w_j$  are both positive and negative, and have a mean value of zero.

After decomposing an image with the wavelet transform, the image can be reconstructed exactly using the inverse wavelet transform:

$$S_0 = \sum_{j=1}^J w_j + S_R, \quad (\text{Equ. 18.4})$$

where  $J$  is the maximum scale in  $w_j$ , and  $S_R$  is the residual image left after the extraction of scales  $w_1$  through  $w_J$ . This equation simply states that the original image is the sum of the wavelet scales plus the residual image. To reconstruct any pixel  $(x,y)$  in  $S_0$ , it is only necessary to sum the corresponding wavelet coefficients plus the residual image:

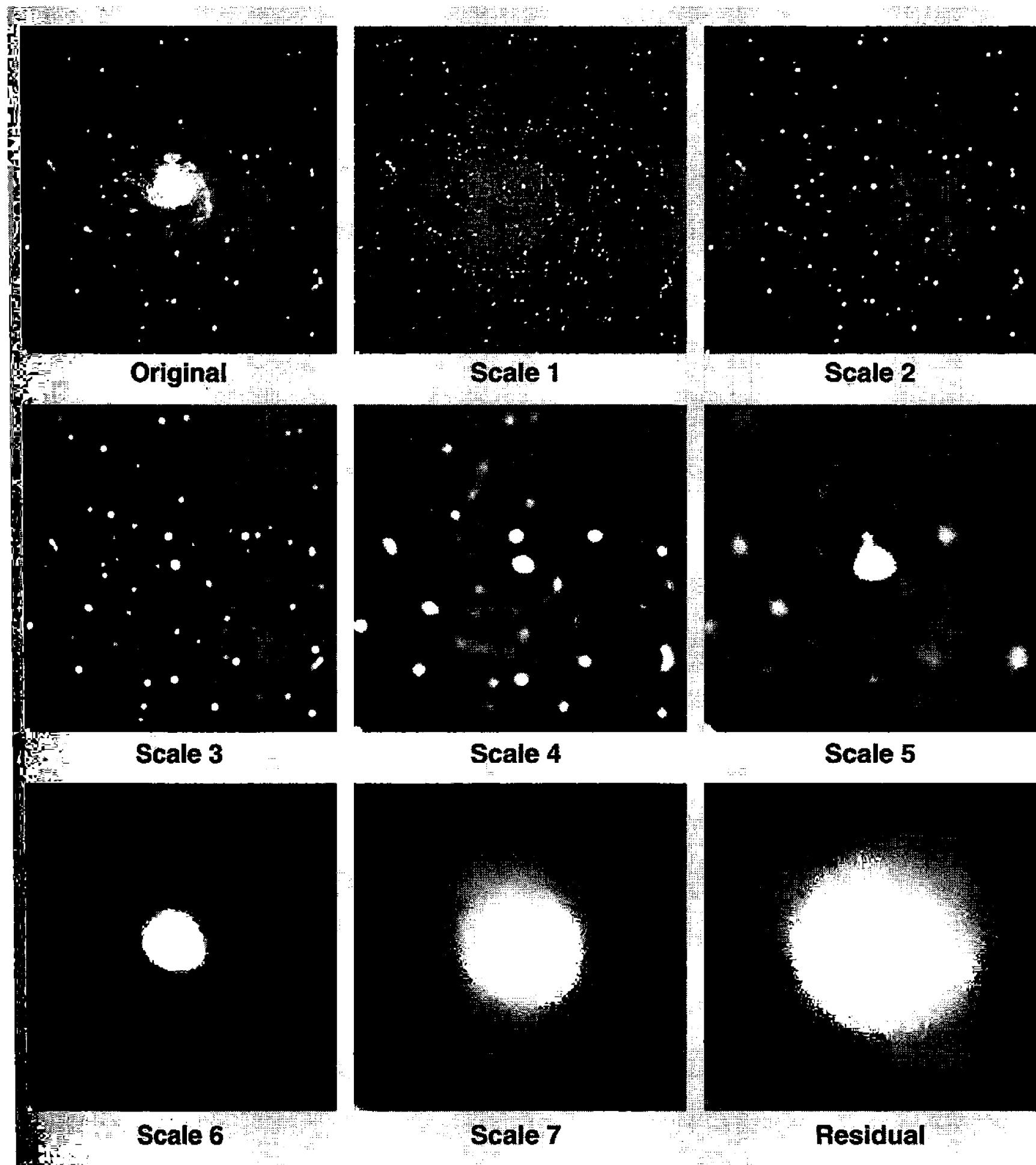
$$S_0(x,y) = \sum_{j=1}^J w(x,y,j) + S_R(x,y). \quad (\text{Equ. 18.5})$$

Figure 18.2 shows an image of the galaxy Messier 101 decomposed into wavelet scales  $w_1, w_2, w_3, \dots, w_7$  plus the residual image. Examine these images carefully. You will see that  $w_1$  shows virtually none of the large-scale structure. The galaxy's spiral arms, for example, are completely absent at a scale of one pixel.

Scales  $w_2$  and  $w_3$ , with resolutions of 2 and 4 pixels respectively, pick out knots and the central ridgeline of the arms. Star images also show as significant peaks at resolutions of 2 to 4 pixels.

The spiral arms finally become prominent in scales  $w_4$  and  $w_5$ . Resolutions of 8 and 16 pixels match the characteristic spatial scale of the arms in this particular image. Note that while the star images are fading (because they lack features

## Section 18.1: The Wavelet Transform



**Figure 18.2** Because they contain information at different spatial resolutions, the appearance of different wavelet scales varies greatly. Scale 1 contains the smallest features; higher scales contain information extracted by the wavelet transform about the structure of the image at progressively lower resolutions.

16 pixels across), the broad nucleus of the galaxy comes into its own!

After  $w_6$  removes the last vestige of fine-scale structure defining the arms and nucleus, very little remains for scale  $w_7$  to display except that, on a scale of 64 pixels, the galaxy is bright in the center and dim at the edges. Even though the very broad structure shown in  $w_6$  and  $w_7$  appears dull to the eye, it nonetheless represents an important large-scale feature of the galaxy.

Finally, the residual image contains the galaxy's leftover brightness after its structure has been extracted. Without this background, a reconstruction of the galaxy image would show its structural features but would not show its overall dis-

## Chapter 18: Wavelets

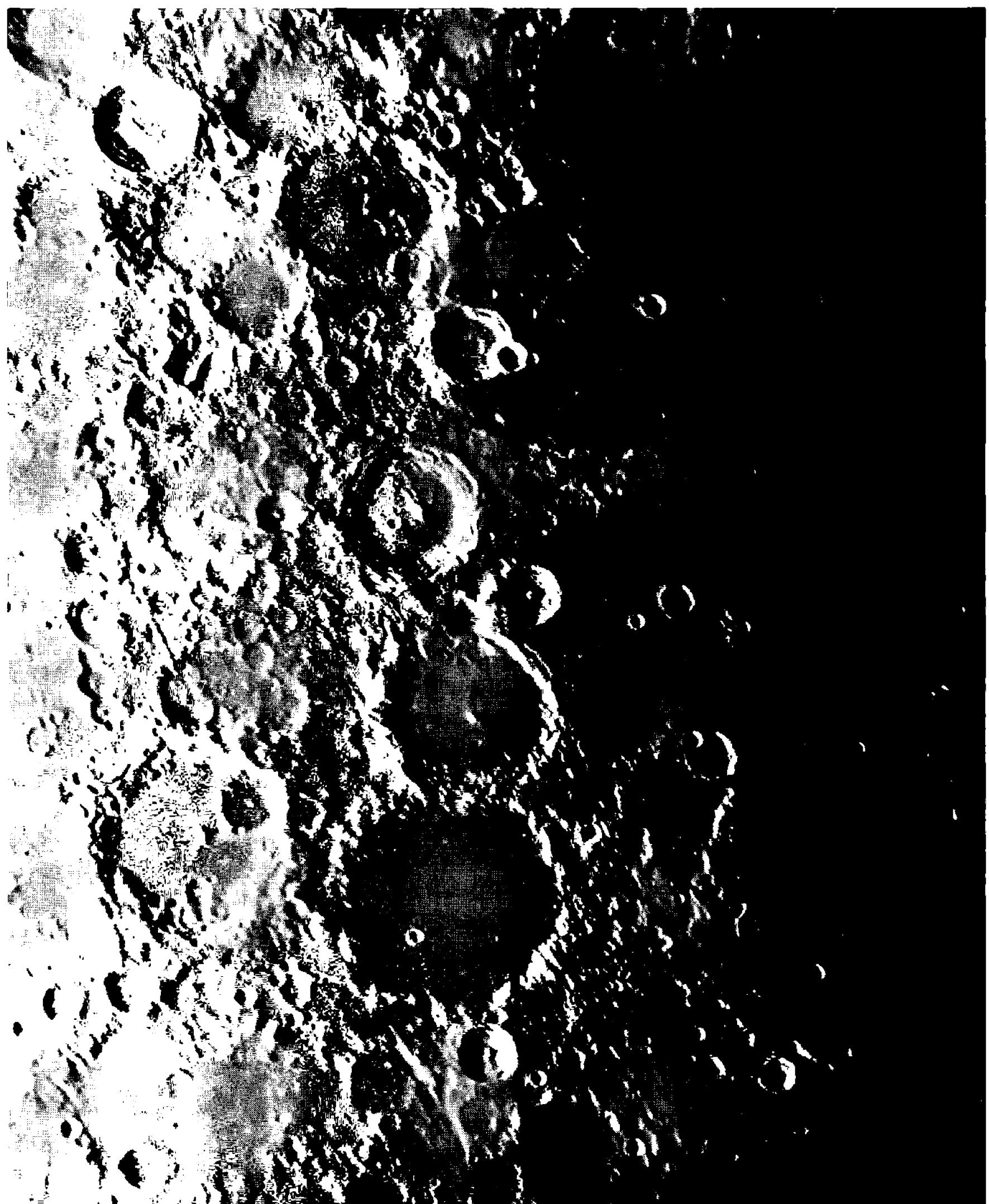


Figure 18.3 Above and on the opposite page are images of the Moon before and after spatial-frequency enhancement using wavelets. The original image appears slightly deficient in high frequency information (fine detail) and at the lowest frequencies, displays a feature-hiding darkness on the terminator side.

tribution of light. For correct reconstruction of the original image, every wavelet scale contains important information. To be useful, however, the wavelet transform must do more than deconstruct and reconstruct images. The next section explores how wavelet processing can act as a filter to suppress and/or enhance information at different spatial resolutions.

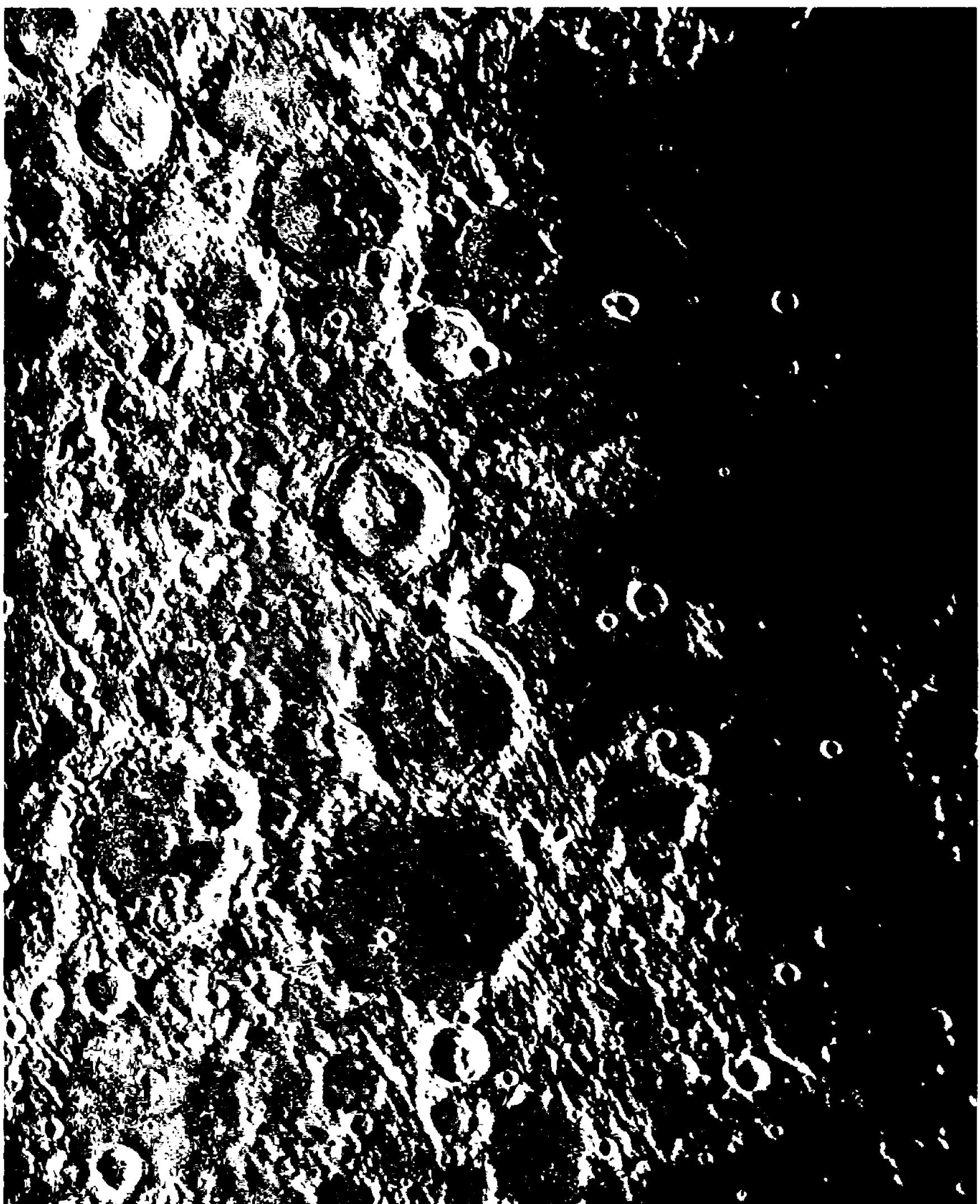


Figure 18.4 Wavelet processing enhanced wavelet Scales 1, 2, and 3 and reduced Scales 6, 7, and 8, thereby increasing the amount of visible detail in craters and maria surface, while simultaneously revealing detail that was lost before due to low-frequency spatial components along the shadowed terminator.

### 18.1.3 Spatial Filtering with the Wavelet Transform

Unsharp masking acts as a spatial filter to strengthen small-scale detail (*i.e.*, high spatial frequencies)—but it is limited to strengthening detail smaller than the radius characteristic of its convolution kernel. In contrast, the wavelet transform makes a wide range of spatial frequencies available for manipulation—and they are available simultaneously in the array of wavelet coefficients. Spatial filtering

## Chapter 18: Wavelets

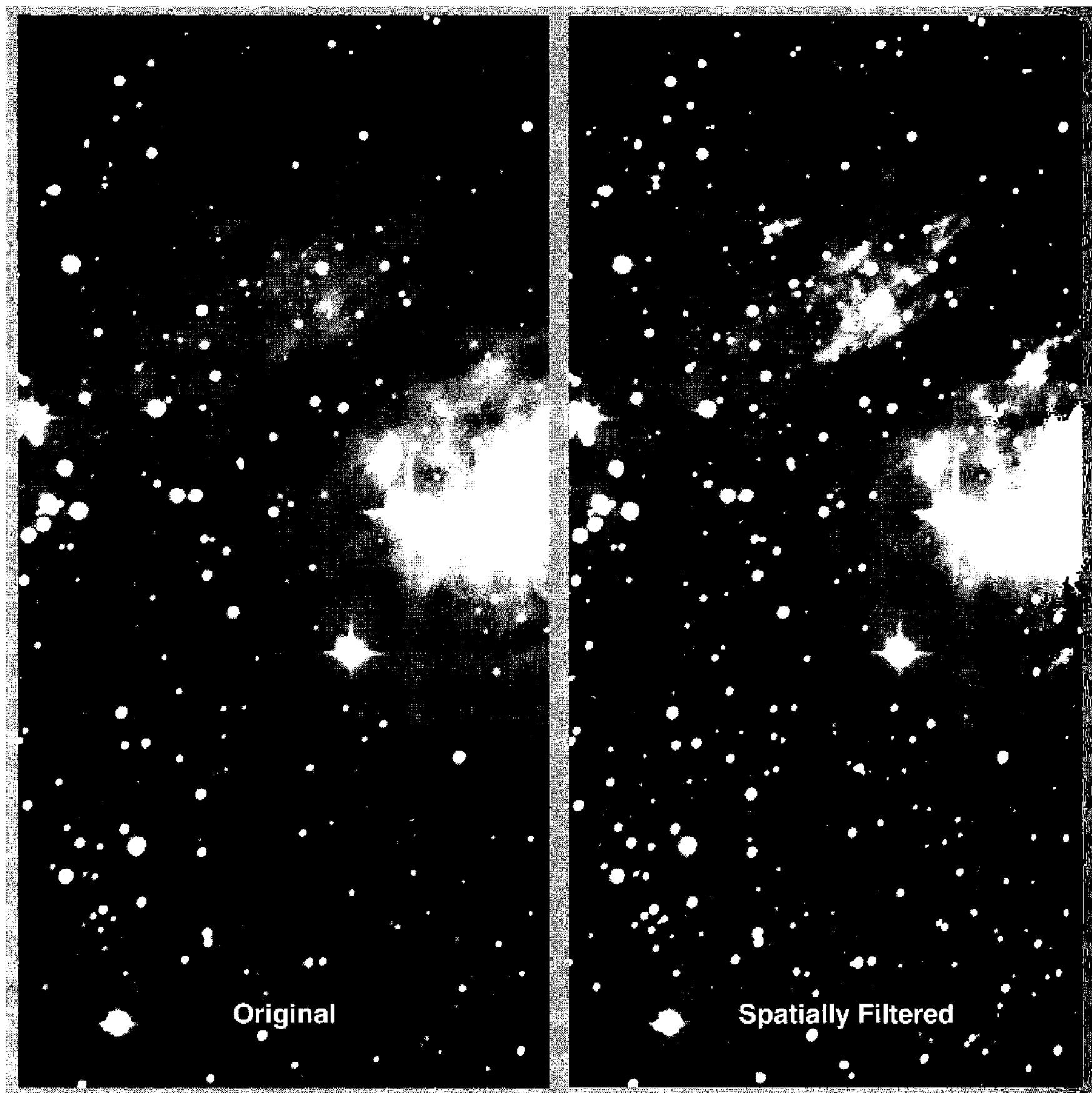


Figure 18.5 With the aid of a conditional operator, wavelets can enhance deep-sky images without generating dark circles around bright star images. In this image of the Lagoon Nebula, wispy nebulosity and faint stars have become more visible without altering the overall image brightness.

is a simple matter of multiplying each wavelet scale  $w_j$  by an appropriate filter factor  $s_j$  to produce a new  $w_j$ :

$$w_j \leftarrow s_j w_j \text{ for } j = 1 \text{ to } J. \quad (\text{Equ. 18.6})$$

Figures 18.3 and 18.4 illustrate the flexibility and power of wavelet spatial filtering. If the wavelet transform of the original image were reconstructed, the result would be the original image. However, if Scales 1, 2, and 3 were increased two or three times, and Scales 6, 7, and 8 were reduced to half or one-third of their original values, fine detail would be enhanced and the large-scale shadings would be reduced. Finally, replacing the residual image with a constant background would eliminate the image darkening at the terminator. This is, in fact, how the original image in Figure 18.3 was filtered to produce the image in Figure 18.4.

Because it acts on all spatial scales at the same time, wavelet spatial filtering with a set of filter factors  $s_j$  can produce an enormous range of effects—much as the graphic equalizer on a stereo system “shapes” the sound of music. Boosting the treble (Scale 1), for example, enhances fine detail; attenuating treble frequencies smooths noisy music (and blurs noisy images).

Wavelet spatial filtering can do anything that unsharp masking can do, and can do it faster and in a more controlled fashion. And, although Fourier transform methods offer considerably finer frequency discrimination, wavelet spatial filtering is faster, more intuitive, and easier to use.

- **Tip:** *The Wavelet Spatial Filter in AIP4Win makes it easy to boost, attenuate, or mute wavelet Scales 1 through 8, and to scale, mute or replace the background image with a constant.*

### 18.1.4 Spatial Filtering with Star Images

In processing astronomical images, stars present a terrific problem. Because pixel values in a star image rise from nothing to a high value and fall to a low value, stars approximate an *impulse function*. In frequency space, the impulse function has components at all frequencies, meaning that a bright star produces a positive wavelet coefficient in every wavelet scale—but because the sum of the coefficients in each wavelet scale is zero, surrounding the positive coefficient of the star image is a ring of negative coefficients.

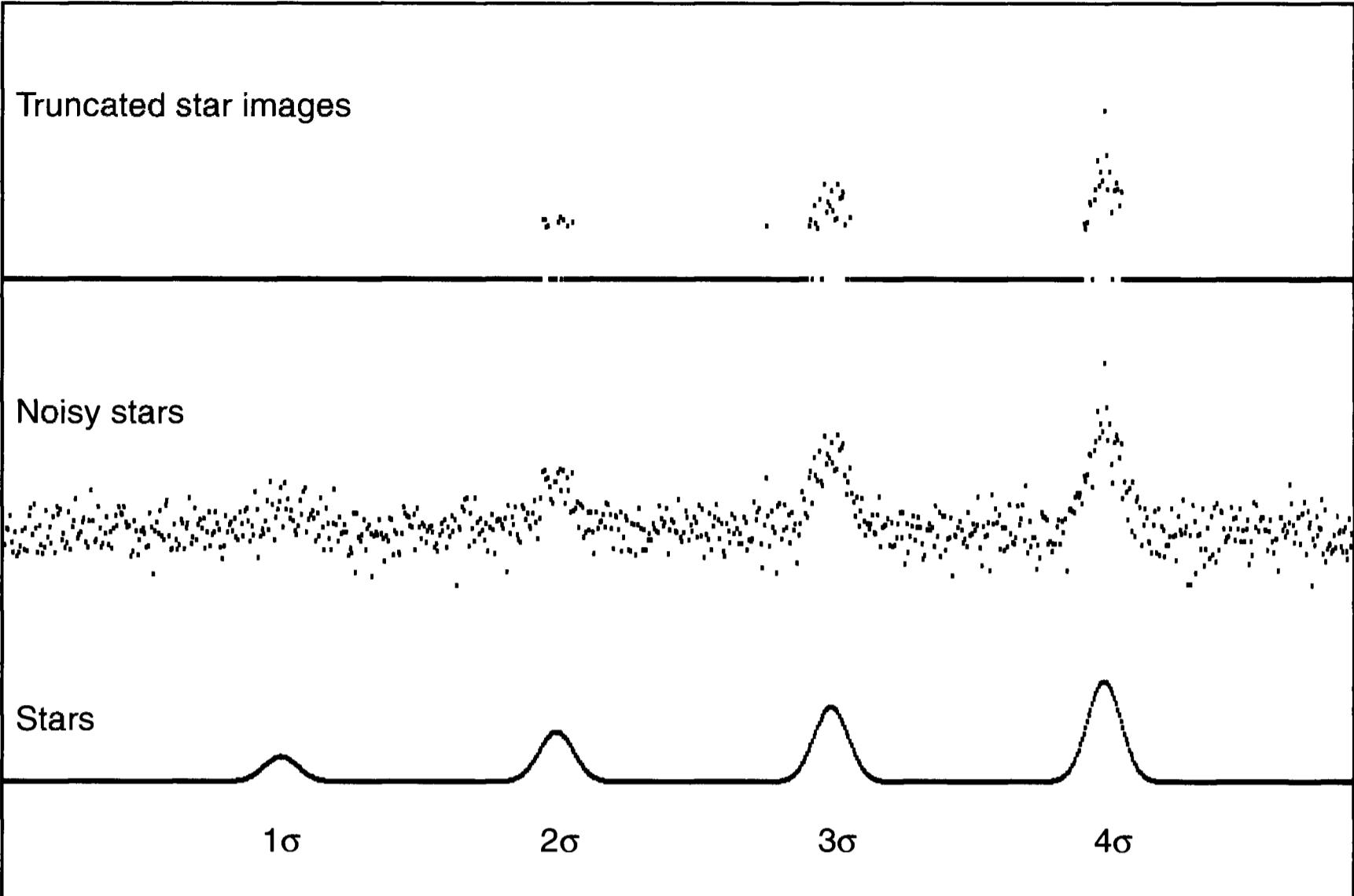
Dark rings occur when wavelet scales are increased. The positive coefficients are more positive, but the negative coefficients are also boosted, making them more negative, producing the dark rings around stars.

To prevent dark rings, we introduce a *non-linear operator* (see Chapter 15) into the inverse wavelet transform to remove the negative wavelet coefficients. The small- and mid-range features in Figure 18.5 were enhanced using a logical “trap-door” operator to prevent a reconstructed pixel from having a lower value than it did in the original image. Although the overall brightness range of the image remains unchanged, small-scale and medium-scale nebular features are brighter and more clearly visible, as are star images—but without dark circles.

- **Tip:** *When using the Wavelet Spatial Filter in AIP4Win to enhance deep-sky images, be sure to check “Prevent Dark Rings.” This activates the conditional operator to prevent dark rings around bright star images when the image is reconstructed.*

## 18.2 Wavelet Noise Filters

Noise is the bane of all astronomers. Poisson noise is inherent in the photon-generated signal, and sensor electronics add a measure of Gaussian noise to every image. Techniques that average pixels in the spatial domain by blending pixels also blur the image, and Fourier techniques fail to eliminate noise because, in frequency space, random noise is present and inseparable from image structure at every



**Figure 18.6** In this figure, the bottom row shows simulated star images with amplitudes of 1, 2, 3, and 4. The middle shows the same stars but with Gaussian noise (with  $\sigma = 1$ ), and the top shows the same data truncated. Reliable noise rejection requires a signal of  $\sim 3\sigma$ .

frequency. Wavelets, as a spatial/frequency hybrid, offer some effective methods of separating and removing random noise from significant signals.

### 18.2.1 When Is an Image Feature Significant?

Against a noisy sky background, the image of a star consists of a few pixels that have a slightly higher value than their neighbors. If a sky background has a mean value of  $B$  and a standard deviation of  $\sigma_B$ , then 68.3% of sky pixels have a value between  $B - \sigma_B$  and  $B + \sigma_B$ , while the remaining 31.7% of pixels lie outside that range—so a pixel that is within  $1\sigma_B$  of the mean sky value is hardly rare enough to qualify as real signal. A pixel that departs  $2\sigma_B$  from the mean is considerably less common—only 4.5% lie outside  $B \pm \sigma_B$ —but still not sufficiently rare to qualify as genuine signal. However, random noise pushes a mere 0.27% of pixels outside the range of values between  $B - 3\sigma_B$  and  $B + 3\sigma_B$ —and for most practical purposes, a so-called “three-sigma detection” constitutes reasonable evidence that a pixel departing  $3\sigma_B$  from the mean is indeed a significant signal.

Suppose that we try to use this criterion to locate stars in an image, and suppose that the image consists of nothing but an average background sky value of  $B$  plus a scattering of stars. (See Figure 18.6.) To find the stars, we measure the standard deviation of the sky background,  $\sigma_B$ , in star-free region and then set any pixel with a value less than  $B + 3\sigma_B$  equal to  $B$ . Any pixel under this threshold is considered a sky pixel, and any pixel greater than the threshold belongs to a star

## Section 18.2: Wavelet Noise Filters

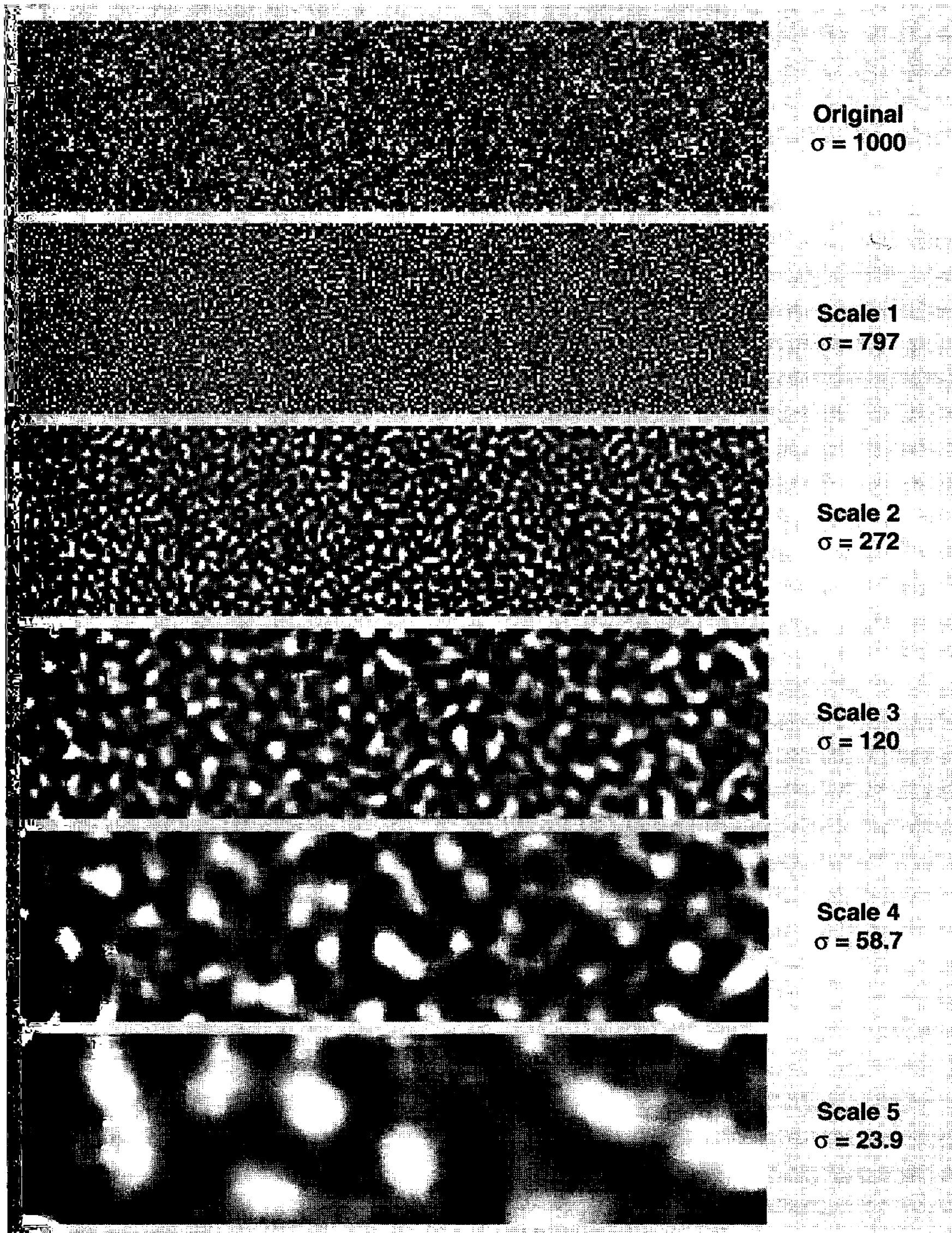


Figure 18.7 Random noise with a Gaussian deviate follows a predictable pattern. If the standard deviation,  $\sigma$ , of the original image is 1.000, then we expect a value of  $\sigma = 0.800$  in Scale 1, 0.274 in Scale 2, 0.120 in Scale 3, and so on. This characteristic allows us to separate real features from random noise.

image. Such a scheme can, in fact, locate star images reliably, especially if we set a higher threshold— $5\sigma_B$ —and eliminate all single-pixel detections.

However, there are many things besides stars in astronomical images. The value of a single pixel in an image of a galaxy may never satisfy a  $B + 3\sigma_B$  crite-

## Chapter 18: Wavelets

rion, but a galaxy image is made from hundreds or even thousands of pixels. This is where wavelets enter the scene, because wavelet analysis can tell us whether a cluster of hundreds or thousands of pixels is statistically significant.

Wavelet noise-filtering methods are based on the characteristic behavior of random noise. Consider an image consisting entirely of random noise, such as the example shown in Figure 18.7. For a given standard deviation of the original image,  $S_0$ , in successive wavelet scales  $w_j$  the standard deviation,  $\sigma_j$ , drops in a predictable way. As each wavelet level is computed, and noise at higher frequencies is stripped away, we're left with smoother and less noisy wavelet scales.

Suppose that we have an image of a faint galaxy that is 32 pixels across. The image will be most evident at wavelet scale 5, and for Gaussian noise, the amplitude of random noise will have dropped by a factor of  $\sim 40$  from  $\sigma_B$ . If the galaxy's wavelet coefficients at scale 5 exceed  $B + (3/40)\sigma_B$ , then that cluster of pixels constitute a significant signal, and the galaxy has been detected!

Determining which features in an image are statistically significant and which are due to noise is the basis for wavelet noise filtering. However, we must overcome three obstacles before we can subject an image to a wavelet noise filter and reject the noise. These are:

- Our scheme relies on the properties of Gaussian noise, but in astronomical images, most noise is Poisson noise with quite different characteristics.
- In images that are full of features, we must measure the noise amplitude that would exist if there were no features.
- As we reject noise in the filtering process, we must test the rejected wavelet coefficients to insure that we have not accidentally rejected genuine features with the noise.

Fortunately, it *is* possible to surmount each of these obstacles. In the following sections, we discuss each in turn and show how to overcome it.

### 18.2.2 Transforming Poisson Noise to a Gaussian Deviate

In images from CCDs and digital cameras, Poisson noise is usually the dominant noise source. Poisson noise differs from Gaussian noise because the standard in an image with Poisson noise depends on the pixel value. In addition to the Poisson noise, CCDs and digital cameras have additive Gaussian noise from readout.

Poisson noise arises from photon-counting statistics. Given a photon flux having an average count rate of  $\bar{n}$  photons per unit time, the standard deviation is  $\sqrt{\bar{n}}$ , thus the noise rises with the square root of the photon count. To a good approximation, if we divide the photon count by its own square root, the resulting value has a standard deviation of 1.0 regardless of the photon count. (This occurs because the noise must also be divided by the square root of the photon count, so  $\sqrt{\bar{n}}/\sqrt{\bar{n}} = 1$ .) This forms the basis for the Anscombe transform that converts Poisson noise to a signal with a constant Gaussian deviation.

In CCD images, a pixel value,  $S$ , is the sum of a bias level plus a photoelec-

## Section 18.2: Wavelet Noise Filters

tron count,  $n$ , divided by the amplifier gain:

$$S = B + n/g \text{ [ADUs]} \quad (\text{Equ. 18.7})$$

where the bias,  $B$ , is in ADUs and the amplifier gain,  $g$ , has units of photoelectrons per ADU. Solving for the photon count gives us:

$$n = gS - gB \text{ [photons].} \quad (\text{Equ. 18.8})$$

Two other factors will enter: the readout noise,  $R$ , which is composed of Gaussian noise, and a small correction term that compensates for the slight asymmetry of the Poisson distribution relative to the Gaussian distribution. Combining these gives an equation that transforms each pixel value in an image into a new pixel value with a standard deviation of 1:

$$S_{\sigma=1} = \frac{2}{g} \sqrt{g(S - B) + R^2 + 0.38g^2}. \quad (\text{Equ. 18.9})$$

To transform a CCD image into a form suitable for wavelet noise filtering, it is necessary to know:

- the amplifier gain in photoelectrons per ADU,
- the bias of the image in ADUs, and
- the readout noise in r.m.s. electrons.

Running wavelet noise filtering on an image that has not been transformed normally results in too little noise filtering in the brighter areas of the image relative to the sky background, or too little noise filtering altogether.

- **Tip:** *AIP4Win's Constant Sigma Scaling Tool performs the Anscombe transform. The transform has little effect on the visual appearance of the image, but Anscombe-scaled images work much better with Wavelet Noise Filtering.*

### 18.2.3 Measuring Noise in Images

When you measure the root mean square variation of the pixel values in an image, the result reflects not only noise, but also variations due to the structure of the image itself. For the synthetic star images shown in Figure 18.6, the noise,  $\sigma$ , was known to be 1.0, so the clipping threshold could be set at  $3\sigma$ , or 3.0. To use wavelets to remove noise, it is extremely useful to know the noise in order to set the clipping threshold correctly. Fortunately, it is possible to use wavelets to isolate and measure noise, and then to threshold wavelet scales to remove it.

The distribution of Gaussian noise follows a regular and repeatable pattern. Each successive wavelet scale in an image consisting of pure Gaussian noise with  $\sigma = 1$  has lower noise:

Scale	Noise, $\sigma_j$
image	1.0000
1	0.8002

## Chapter 18: Wavelets

Scale	Noise, $\sigma_j$
2	0.2735
3	0.1202
4	0.0585
5	0.0291
6	0.0152
7	0.0080
8	0.0044

Clearly, the bulk of the noise is in wavelet Scale 1. If we can figure out a way to isolate those pixels in the image which contain no significant image information in Scale 1—in astronomical images, these are the starless parts of the sky background—we can determine the noise in Scale 1 and use the value in the table to find the contribution due to Gaussian noise at every other scale.

To do this, we create a special type of image called a *significance image* or *multiresolution support image* that encodes significant features—that is, non-noise features—of an image. The process begins with a wavelet transform and a best-guess estimate,  $\sigma$ , of the noise in the image. At each scale in the wavelet transform, we treat a wavelet coefficient as significant if its absolute value is equal to or greater than  $3\sigma_j$ :

$$\Gamma_j(x, y) = \begin{cases} 1 & \left| w_j(x, y) \right| \geq 3\sigma_j \\ 0 & \left| w_j(x, y) \right| < 3\sigma_j \end{cases} . \quad (\text{Equ. 18.10})$$

Next, compute the significance image. Using the Greek letter gamma,  $\Gamma$ , to represent the significance image, with  $\Gamma_j(x, y)$  representing the value of  $\Gamma$  at scale  $j$ , compute pixel values for the significance image:

$$\Gamma(x, y) = \sum_{j=1}^J 2^j \Gamma_j(x, y) . \quad (\text{Equ. 18.11})$$

If  $\Gamma(x, y) = 0$ , the pixel in the image has no structure at any scale exceeding the estimated noise  $\sigma$ , so we compute the standard deviation for all pixels that satisfy this condition, to produce a new estimate of  $\sigma$ . Because the initial best-guess estimate may have been wrong, the significance image may have inaccurate—but the new value of  $\sigma$  should be closer to the correct value than the initial estimate. To get the best estimate, we use the new value of  $\sigma$  to compute a new significance image, and iterate until the value of  $\sigma$  converges, usually with a few iterations.

- **Tip:** While significance images are decidedly ugly, they are useful “inside features” that **AIP4Win** uses to measure and filter noise.



Figure 18.8 The multiresolution support images, or statistical significance images, are used internally by image-processing software to determine which parts of an image contain statistically significant information (the gray and white areas), and which are dominated by noise (the black areas).

#### 18.2.4 Rejecting Noise and Retaining Significant Features

The third obstacle in separating noise from statistically significant features in an image is to insure that even as noise is cut, significant image features are *not* cut. The best way to accomplish this comes neither from finely tuning a one-pass filter nor from some kind of brute-force technique, but by iteratively correcting and testing the image using a method similar to that used to determine the noise.

Suppose, for a moment, that you did have a filter that separated noise from signal perfectly: What would you see if you took an image, filtered it, and then subtracted the filtered image from the original noisy image? That residual image would have nothing but noise, free of any trace of the original image. This suggests that an effective method for removing noise will be an iterative process that filters, tests, and corrects the filtered image.

## Chapter 18: Wavelets

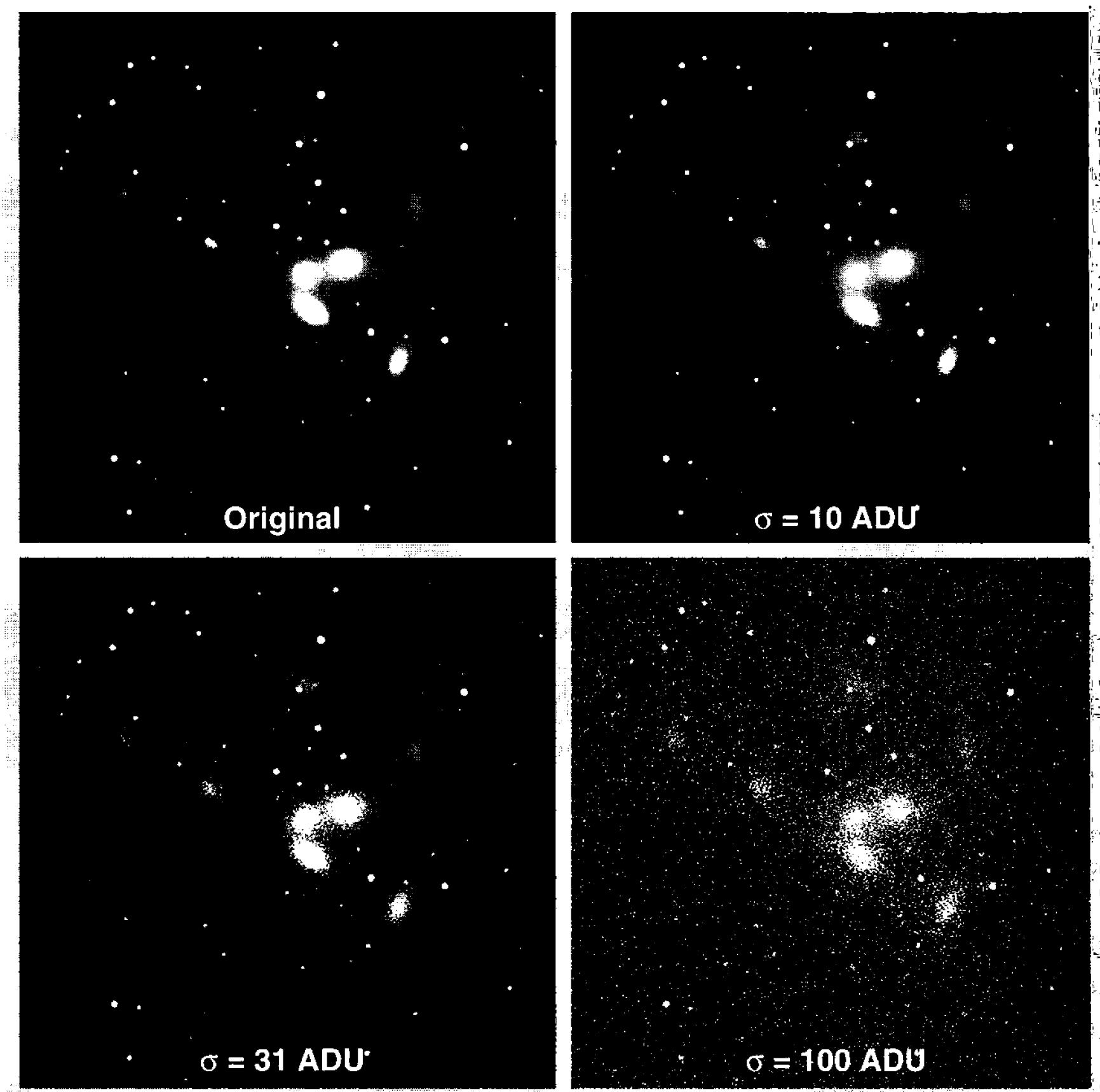


Figure 18.9 Case study: Iterative wavelet noise removal in images with different amounts of noise. The original image (upper left) contains no noise; the three other images have 10, 31, and 100 ADUs of added Gaussian noise. At 31 ADUs, noise is ugly, and 100 ADUs it has nearly destroyed the image.

### 18.2.5 The Solution: An Iterative Wavelet Noise Filter

We have now seen that the three obstacles to effective wavelet noise filters can be overcome. Images containing Poisson noise and mixtures of Poisson noise and Gaussian noise can be transformed into pure Gaussian noise. We also have an effective method for measuring noise in images, and finally, we've seen that an iterative procedure that filters, tests, and corrects should be able to cleanly separate statistically significant image features from noise.

Here then is a procedure that combines the predictable behavior of Gaussian noise, an accurate method for measuring noise, and wavelet filtering with active feedback:

1. Convert an image with a mixture of Poisson noise and Gaussian noise to Gaussian noise using the Anscombe transform. Although this requires

## Section 18.2: Wavelet Noise Filters

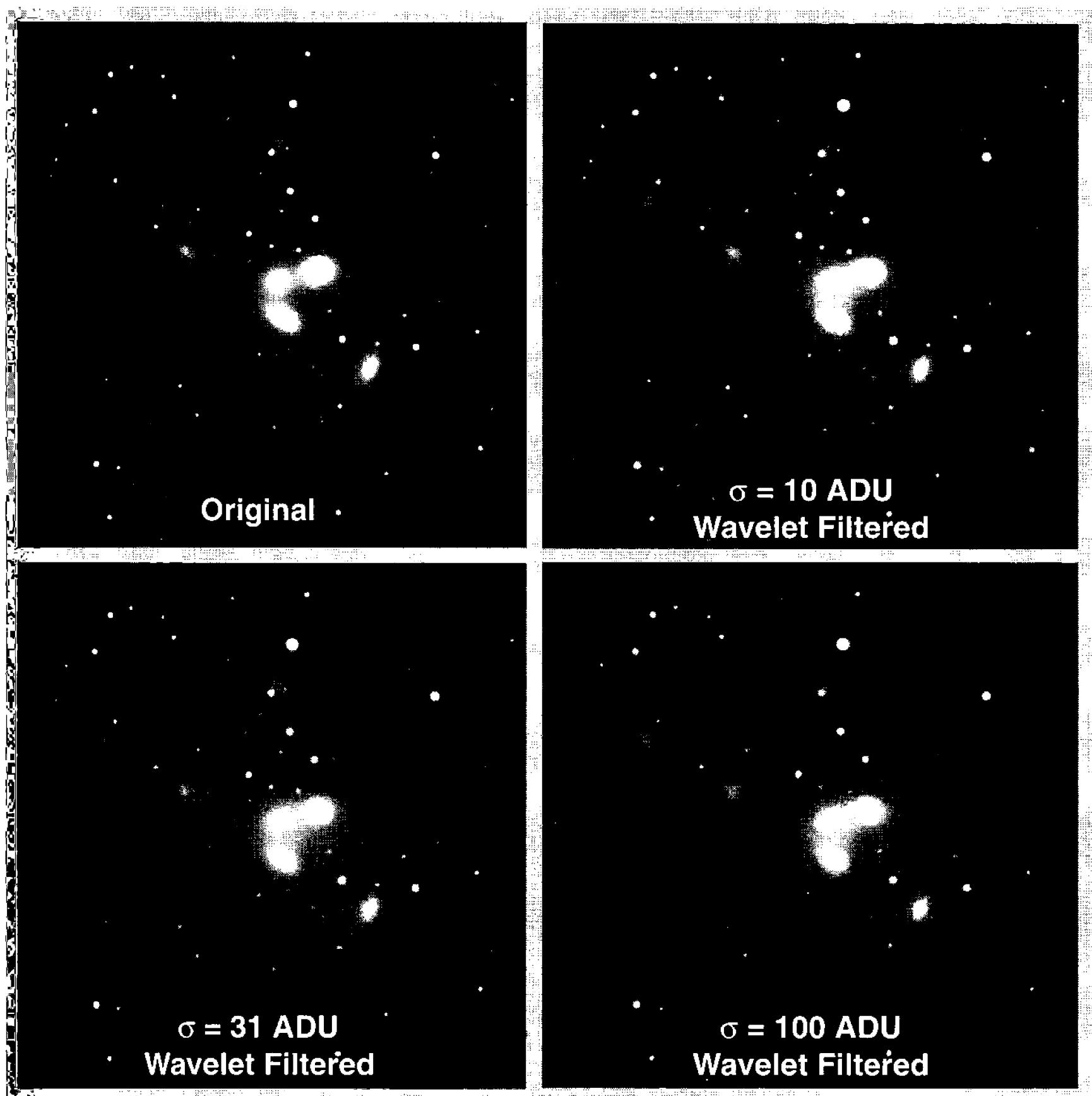


Figure 18.10 Here are the images from Figure 18.9 as they appear after iterative wavelet filtering, with the original noise-free image (upper left) for comparison. To a good approximation, the filtered images now appear the same as the original—yet before wavelet filtering they contained large amounts of noise.

knowing the gain and readout noise in your CCD camera, as well as the bias value for the original image, it insures accurate noise removal.

2. Measure the amplitude of the noise in the transformed image using the iterative noise measurement procedure described in the previous section. From the total noise, compute the noise expected at every wavelet scale, and use this information to create a significance image.
3. Create a solution image and set all pixel values in it to zero. This image begins filled with zeros, but it will eventually contain the filtered image.
4. Subtract the solution image from the original image to produce a residual image. The residual image contains the difference between the ideal filtered image and the original. Initially, it will contain a copy of the original image, but that is about to change!

## Chapter 18: Wavelets

5. Perform a wavelet transform on the residual image. Using the noise values computed for each scale, truncate the wavelet coefficients, then perform an inverse wavelet transform.
6. Add the inverse wavelet output to the existing solution image. The first time through this process, the solution image begins full of zeros; on later passes, it contains the solution from the previous iteration.
7. Subtract the solution image from the original to create a new residual image. If the noise filter were perfect, the residual image would contain only noise, but chances are that some image features were incorrectly removed and some noise remains in the solution image.
8. Measure the noise in the residual image, and use the noise figure to create a new significance image based on the residual image.
9. Iterate by returning to step 5. At each pass, significant features that remain in the residual image will be added to the solution image. When the noise you measure in the residual image stabilizes at a steady value, all significant image structure has been transferred to the solution image and all noise under  $3\sigma$  has been left in the residual image—and filtering is complete.

This procedure requires that you know your camera’s gain and readout noise, and the output bias of the image so that you can convert the noise in the image to Gaussian noise. Applying an iterative wavelet filter to images in which Poisson noise is dominant results in image artifacts and oversmoothed output images.

Iterative wavelet filtering works best for critically sampled and oversampled images. In undersampled images, a background filled with small star images can appear as noise, and produce overfiltering artifacts. In critically sampled and oversampled images, noise is determined correctly, and the image is correctly filtered.

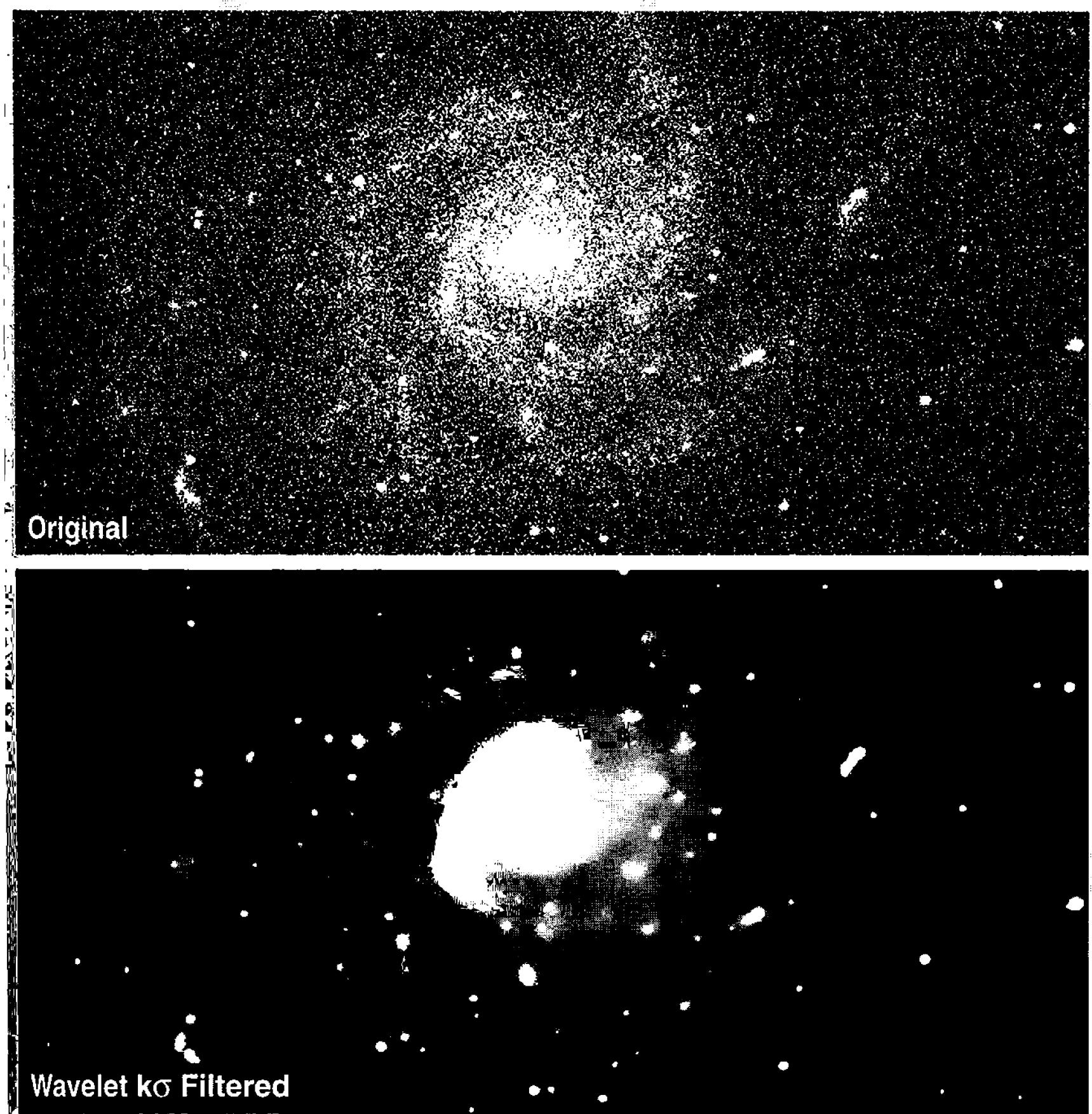
Wavelet noise filtering solves the problem of separating real features from image noise—like a lawn mower that cuts the grass without cutting the daisies. Attempting to cut noise (“grass”) in frequency space with the Fourier Transform always cuts image detail (“daisies”), and in image space, smoothing and averaging with convolution methods removes both daisies and grass. The hybrid nature of wavelets allows cutting noise without losing statistically significant image detail.

- **Tip:** *The Iterative Wavelet Filter in AIP4Win removes “grain” from images that have been transformed to Gaussian noise using Constant Sigma Scaling. Figures 18.9 and 18.10 show how effective wavelet filtering can be on images that have small, medium, and large amounts of noise.*

### 18.3 Wavelet K-Sigma Filtering

In the section above, we saw that wavelet filtering can remove noise from images—providing it is well-behaved Gaussian noise. However, CCD and digital camera images usually contain a mixture of Poisson noise, Gaussian noise, and complex noise types generated in dark subtraction, flat-fielding, and image stack-

## Section 18.3: Wavelet K-Sigma Filtering



**Figure 18.11** Wavelet  $k\sigma$  filtering filters images with any noise type, but requires that the user specify the acceptable noise level at each wavelet scale. In this example, a standardized “strong” filter smoothed away background noise without significant blurring or softening of image detail.

ing. The complexity of the noise makes it difficult to estimate the effective gain, readout noise, and bias of the processed image required for constant sigma scaling.

Furthermore, operations used in stacking images such as translation, rotation, scaling, and resampling act as blur filters, reducing the amplitude of the wavelet scale  $j=1$  coefficients with little effect on higher wavelet scales.

Taken together, calibration and stacking deprive us not only of the ability to gauge the total noise in an image, but also of the ability to predict the noise at the high scales based on the noise measured in scale  $j=1$ . And without *a priori* knowledge of how much noise to expect at each scale, iterative solutions are prone to failure. Unfortunately, almost all CCD images and many digital camera images have been calibrated and many have been stacked. What to do?

## Chapter 18: Wavelets

Wavelet  $k\sigma$  ( $k$ -sigma) filtering is designed to deal with the messy noise types found in such images. Rather than assuming well-behaved noise, the wavelet  $k\sigma$  filter requires the observer to specify an acceptable noise threshold,  $k_j$ , for every wavelet scale. In operation, the filter then determines the expected noise at each scale,  $\sigma_j$ , and truncates wavelet coefficients smaller than  $k_j\sigma_j$ . After truncation, an inverse wavelet transform reconstructs a filtered image.

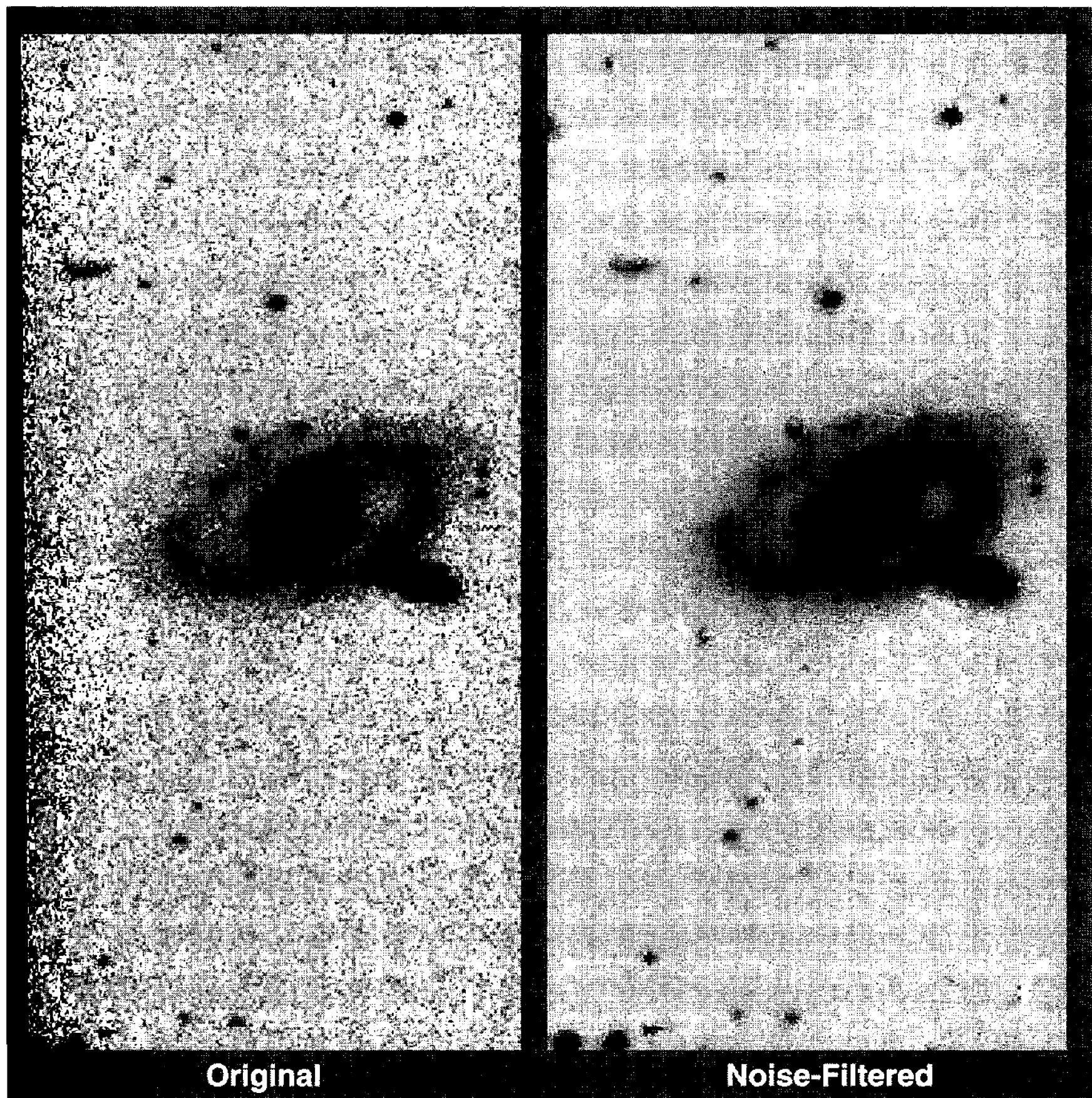
The  $k\sigma$  filter is extremely versatile. For example, by setting  $k_j$  equal to 3 for wavelet Scales 1, 2, and 3, the user can reject noise that is probably similar to Gaussian noise in the lower levels. Since a  $k\sigma$  filter determines  $\sigma_j$  independently for each level, it doesn't matter whether the noise behaves like Gaussian noise at all scales. In the higher levels, which normally contain little noise, then taper the  $k_j$  values so that  $k_4=2$ ,  $k_5=1$ , and  $k_6$ ,  $k_7$ , and  $k_8$  all equal 0. The effect will be that the filter removes 99.7% of the "noise" it finds in Scales 1, 2, and 3, 95.4% of the "noise" in Scale 4, 68% in Scale 5, and none in Scales 6, 7, and 8. However, significant wavelet coefficients—defined as those coefficients greater than  $k_j\sigma_j$ —will pass through the filter undiminished.

With ordinary CCD images, it is useful to define standard filter types that produce good results with a wide variety of images, and also to allow the user to develop custom filters. Here are some proven  $k_j$  values for wavelet  $k\sigma$  filters with effects from gentle to strong:

Wavelet Scale	Gentle	Normal	Strong
1	3	4	5
2	2	3	4
3	1	2	3
4	0	1	2
5		0	1
6			0
7			0
8			0

To define an effective custom filter, it is easiest to crop a test section from the whole image. Start with the standard filter that most nearly matches the filtration you wish to achieve, and then modify  $k_j$  values and test the action on the small test image. When the custom filter works well, process the whole image.

- **Tip:** *The Wavelet K-Sigma Filter in AIP4Win provides a versatile way to remove noise and graininess from images that have a mixture of noise types. The user checks whether each wavelet scale will be active, and specifies a  $k_j$  value for each active wavelet scale. The Wavelet K-Sigma Filter is powerful and effective, but it has a fairly steep learning curve.*

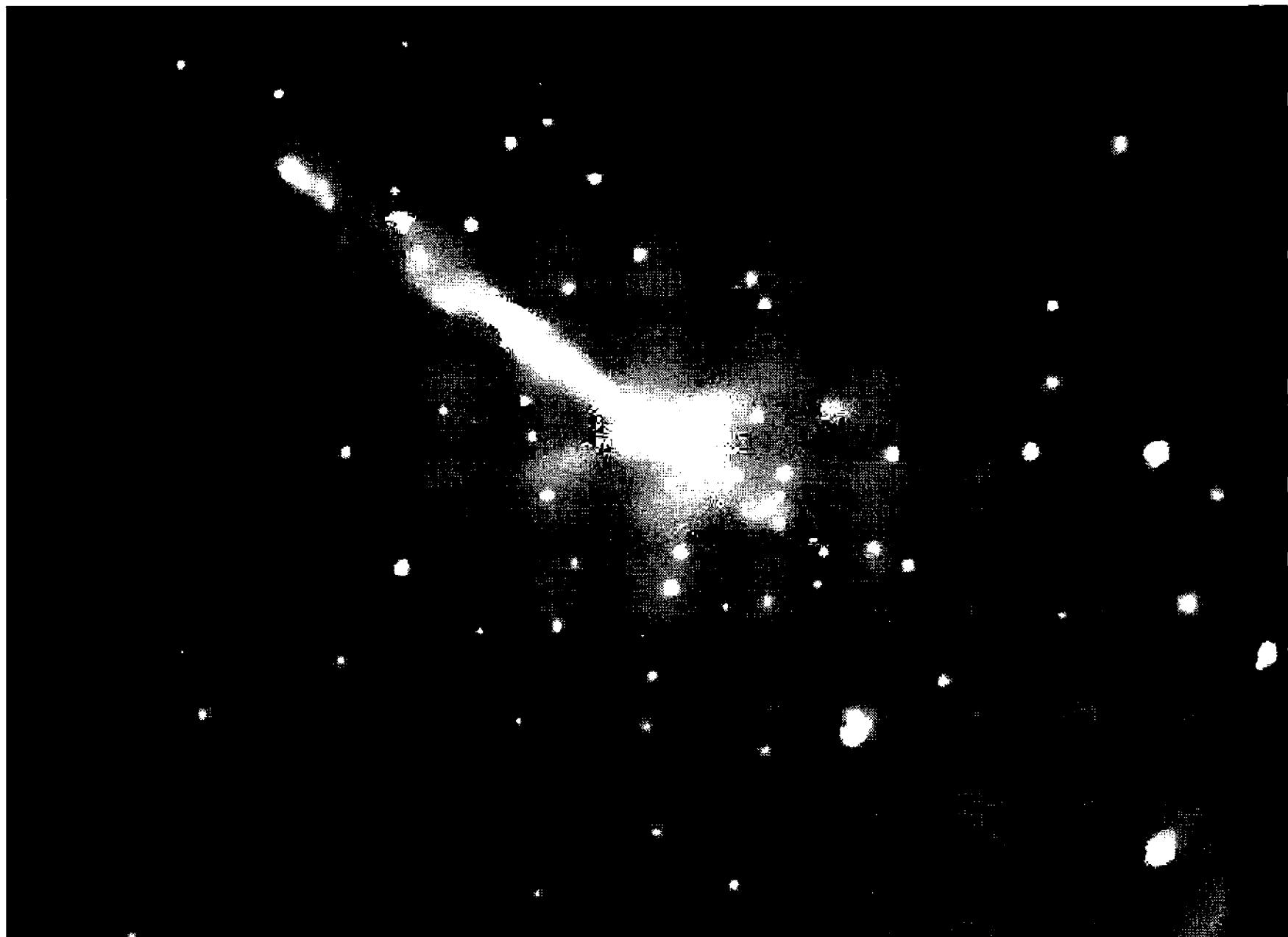


**Figure 18.12** This galaxy is only one among hundreds in Abell 1367. In the original image, structural details are difficult to distinguish from the random sky noise. After wavelet iterative noise processing, the sky background is smooth, and faint background galaxies show readily. Image courtesy of Greg Bothun.

## 18.4 Using Wavelets

How do wavelets fit into the processing of astronomical images? In this chapter, we have discussed two primary applications: image sharpening and noise removal. For sharpening images of the Moon and planets, wavelets enable you to enhance an image on multiple spatial scales at the same time. You can multiply some detail strongly, tweak mid-size structure just a bit, and gently soften unwanted image shadings and gradients—simultaneously. Wavelet spatial filters do a nice job on landscapes, portraits of people, and close-up pictures of flowers, too!

Wavelet noise removal is an important option for observers who want to push their imaging close to the limits imposed by faintness and noise—for observations hindered by the lack of photons, or when observing time is limited. If you can reach a signal-to-noise ratio better than  $\sim 30$  against the sky background with



**Figure 18.13** This x-ray image of the jet in Centaurus A was created with photon-statistics algorithms related to wavelet processing. It displays the characteristic milky appearance of wavelet noise-filtered images. Wavelet filtering handles both point and extended sources. Image courtesy of CXO/SAO/NASA.

exposures totalling an hour or two, wavelet noise removal can smooth the image and lend it a “milky” or “nebular” appearance, but you would be better off to expose longer and to collect more photons. However, for photon-starved imaging where the object is lost in a salt-and-pepper sky, where the longest feasible integration times yield a signal-to-noise of 1 or 3 or 10, wavelet noise filtering provides an objective method of separating objects from clusters of noise pixels.

In professional astronomy, wavelets play a role in processing images and spectra. Data from the European Space Agency’s Infrared Space Observatory is processed to remove noise and to distinguish between instrumental artifacts and real signals. Astronomers reducing data from NASA’s Chandra X-Ray Observatory use statistical analysis software that is similar to wavelet noise filtering to make public-release images from the x-ray photons striking Chandra’s detectors (see Figure 18.13). Smoothing is necessary because although x-ray photons have a lot of energy, the number of them reaching the detector is usually very small—after days of integration, a well-exposed x-ray image may contain only 10,000 x-ray photon hits in a million-pixel image! Statistical analysis determines which clusters of pixels are point sources, which are small extended objects, and which are large extended sources.

## 19 Deconvolution

---

Deconvolution attempts to restore an image that has been degraded to its original pre-degraded condition. Image degradation stems from two sources:

- convolution of the image before detection, and
- noise added to the image in the process of detection.

Atmospheric turbulence, telescope optics, and mechanical instability account for pre-detection convolution. As parallel rays of light from a distant star pass through Earth's layered and inhomogenous atmosphere, they are refracted in slightly different directions, enlarging the image and causing it to move. In addition, the point image is convolved with the telescope aperture to generate the Airy diffraction figure. Finally, the telescope vibration and tracking errors act as a convolution function to further enlarge images. This is the image that falls on the detector.

Noise in the detector causes further degradation. Because the number of photons making up the image is finite, statistical variations in the flux of photons—Poisson noise—mean that even images from a perfect detector suffer from random variation. To this statistical error, CCD cameras add readout noise and quantization error. The brighter the image and the longer the integration time, the higher the signal-to-noise ratio, and the smaller the relative importance of noise—but noise is always present and cannot be eliminated entirely.

In mathematical terms, a digital image looks like this:

$$s(x, y) = k \otimes o(x, y) + n \quad (\text{Equ. 19.1})$$

where  $o(x, y)$  represents the original image outside the atmosphere,  $k$  is a composite point-spread function caused by atmospheric turbulence, optics, shake, and tracking errors,  $n$  is random noise, and  $s(x, y)$  represents the image captured through the telescope.

In prospect, deconvolution looks simple. From this formulation, the obvious question is: can't we simply divide the Fourier transform of the image by the Fourier transform of the point-spread function to recover the original image?

Unfortunately, the answer is “no.” In this chapter you will learn not only why that is so, but also what is practical to accomplish with deconvolution.

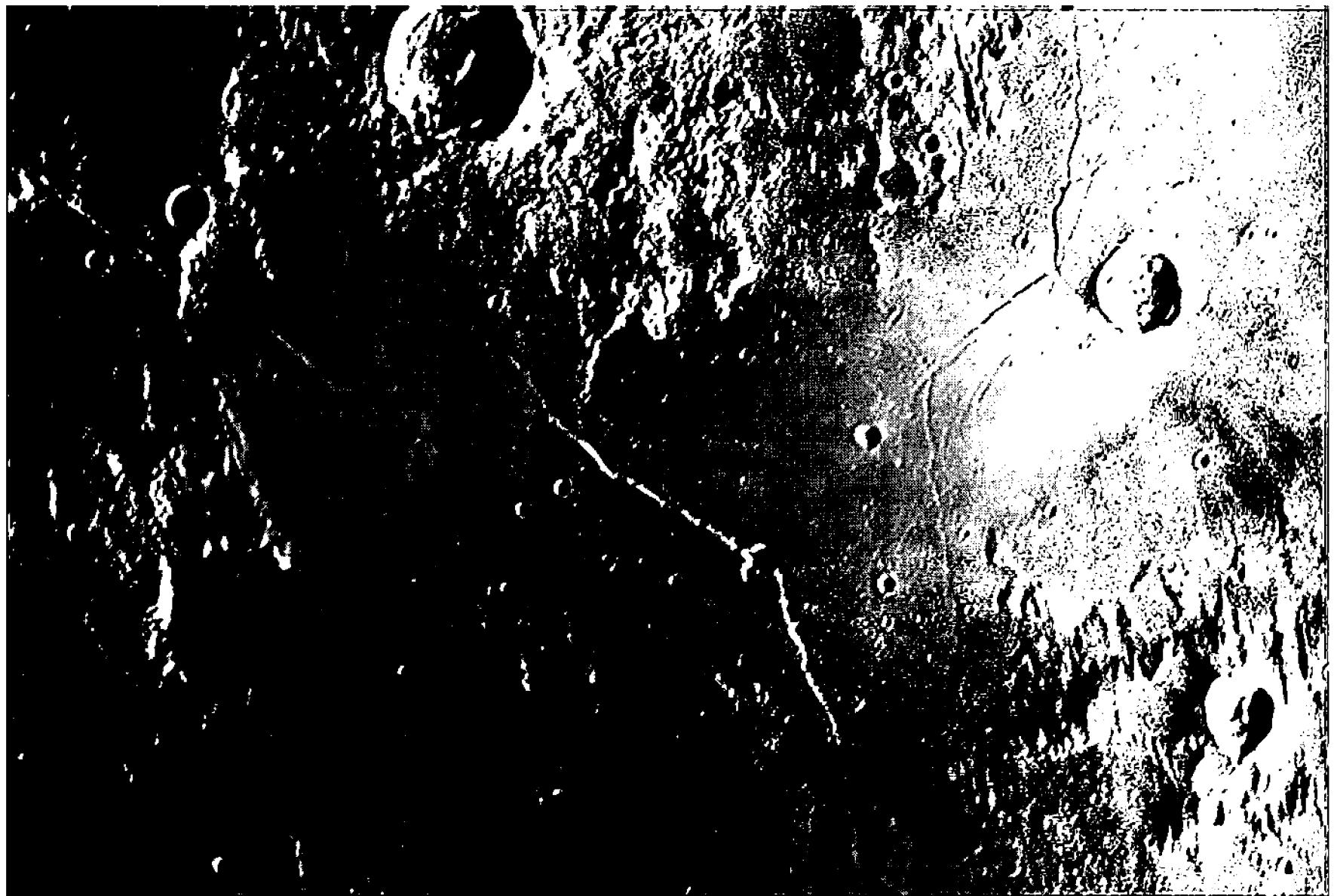


Figure 19.1 After deconvolution, Thierry Legault's image of the Hyginus, Ariadaeus, and Triesnecker rille system is sharp and full of detail right down to the single-pixel level. Processing for this image consisted of five iterations using van Cittert deconvolution followed by histogram shaping.

## 19.1 The Inverse Convolution Problem

Convolution is a determinate process: you start with an image, convolve it with some point-spread function, and get a single well-defined result. Deconvolution is not determinate. In principle, the image that you capture at the telescope could have been created from many *different* original images. Consider a one-dimensional example; this one-dimensional image has just been captured:

$$s(x) = [\dots 0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 1 \ 0 \ 0 \ 0 \ \dots].$$

You happen to know that the exact convolution kernel is:

$$k = [0.25 \ 0.50 \ 0.25],$$

which is a basic Gaussian-like blur. Recovering the original should be easy, right? Wrong. Here are two candidates for the original image:

$$[\dots 0 \ 0 \ 0 \ 0 \ 4 \ 0 \ 4 \ 0 \ 4 \ 0 \ 4 \ 0 \ 0 \ 0 \ \dots]$$

and

$$[\dots 0 \ 0 \ 0 \ 0 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 0 \ 0 \ 0 \ 0 \ \dots].$$

## Section 19.2: Image Estimation by Iteration

If you calculate their convolutions with  $k$ , you will find that both reproduce the central string of 2's in  $s(x)$  quite nicely. The only differences are the values at the ends of the 2's, and those small differences tell you nothing about the middle of the image. Like it or not, it is impossible to distinguish which of these two very different candidates produced  $s(x)$ . You can readily imagine many other candidate originals, also indistinguishable after convolution.

Let's look at the example again, but this time with an image that has a small amount of noise. The image  $s(x) + n$  contains random noise:

$$[\dots 0.1 -0.1 0.1 0.9 1.9 2.2 2.1 2.0 1.9 2.0 2.1 1.1 0 -0.1 0 \dots].$$

Adding noise means that you cannot "trust" any value in the image to better than some standard deviation. This uncertainty increases the number of candidate originals astronomically, because it is no longer necessary that they match  $s(x)$  exactly; any match within the noise envelope is a candidate solution.

Now consider the convolution kernel, which we have assumed that you know perfectly. With a real image, where would it come from? Astronomers are extremely lucky because celestial images are littered with point sources—stars—each of which is a record of the point-spread function. So you can measure the point-spread function from an image, like this image of a one-dimensional star:

$$[\dots 0 0 0 0 0 0 0.98 2.01 1.01 0 0 0 0 0 0 \dots].$$

Unfortunately, even the best point-spread function is slightly noisy. Thus, any realistic scheme for deconvolution should be able to determine the original image from a noisy detected image and a noisy point-spread function.

Mathematicians call deconvolution an ill-posed problem. The problem is ill-posed because it doesn't have a unique solution. Furthermore, in images that are noisy, it is possible that no original image can produce the image you have captured. Finally, the solution is divergent—that is, tiny errors in the detected image can produce large errors during an attempt to restore the original.

The key to image restoration lies in the physical limitations that real images place on the purely mathematical solutions. Rather than seek a *unique* original image, we look for a *statistically plausible* original. And rather than search among all possible mathematical solutions, we constrain the solution to a realistic original image by insisting, for example, that it conform to reality by having no areas of negative brightness. In the end, the key to deconvolution lies in shifting the goal from a mathematically rigorous solution to finding ways of searching for a best approximation of the original image.

## 19.2 Image Estimation by Iteration

Successful deconvolution methods work using a constrained iterative process of image estimation and correction. Rather than attempt to solve the inverse convo-

## Chapter 19: Deconvolution

lution problem in a single step, image estimation relies on a series of small steps—successive approximations—that eventually converge on a best estimate of the original image.

Let's consider how successive approximation works. We begin by defining the original ideal image and the captured image:

- $s(x, y)$  : the degraded noisy image captured by a CCD camera;
- $o(x, y)$  : the (unknown) original undegraded noise-free image;
- $k$ : the point-spread function, known from star images; and
- $n$ : random noise from the CCD.

Recall from Equation 19.1 that the relationship between the two images, the point-spread function, and image noise is:

$$s(x, y) = k \otimes o(x, y) + n ;$$

that is, the image that the CCD records is the convolution product of the point-spread function with an undegraded image, with detector noise added.

We also define another image,  $e(x, y)$ , which represents our best estimate of the convolved-but-noiseless image that fell on the detector; namely:

$$e(x, y) = k \otimes o(x, y) . \quad (\text{Equ. 19.2})$$

This image is the convolution of the original one with the point-spread function. Of course, we don't know what  $e(x, y)$  is, but we can estimate the point-spread function from star images in  $s(x, y)$ . If we knew what the original was, we could compute  $e(x, y)$  by convolving the original with the point-spread function; but, of course, the original image is what we're trying to recover.

This sounds like variation on the chicken-and-egg problem. In the world of mathematics, it is sometimes possible to solve such problems.

We begin by asking what condition(s) must be satisfied to recover the original. Suppose that we subtract  $e$  from both sides of Equation 19.1 to obtain:

$$s(x, y) - e(x, y) = n . \quad (\text{Equ. 19.3})$$

This says that the difference between these image is the noise. If we set the condition—one that we know cannot be met—that the noise is zero, or at least too small to matter, then:

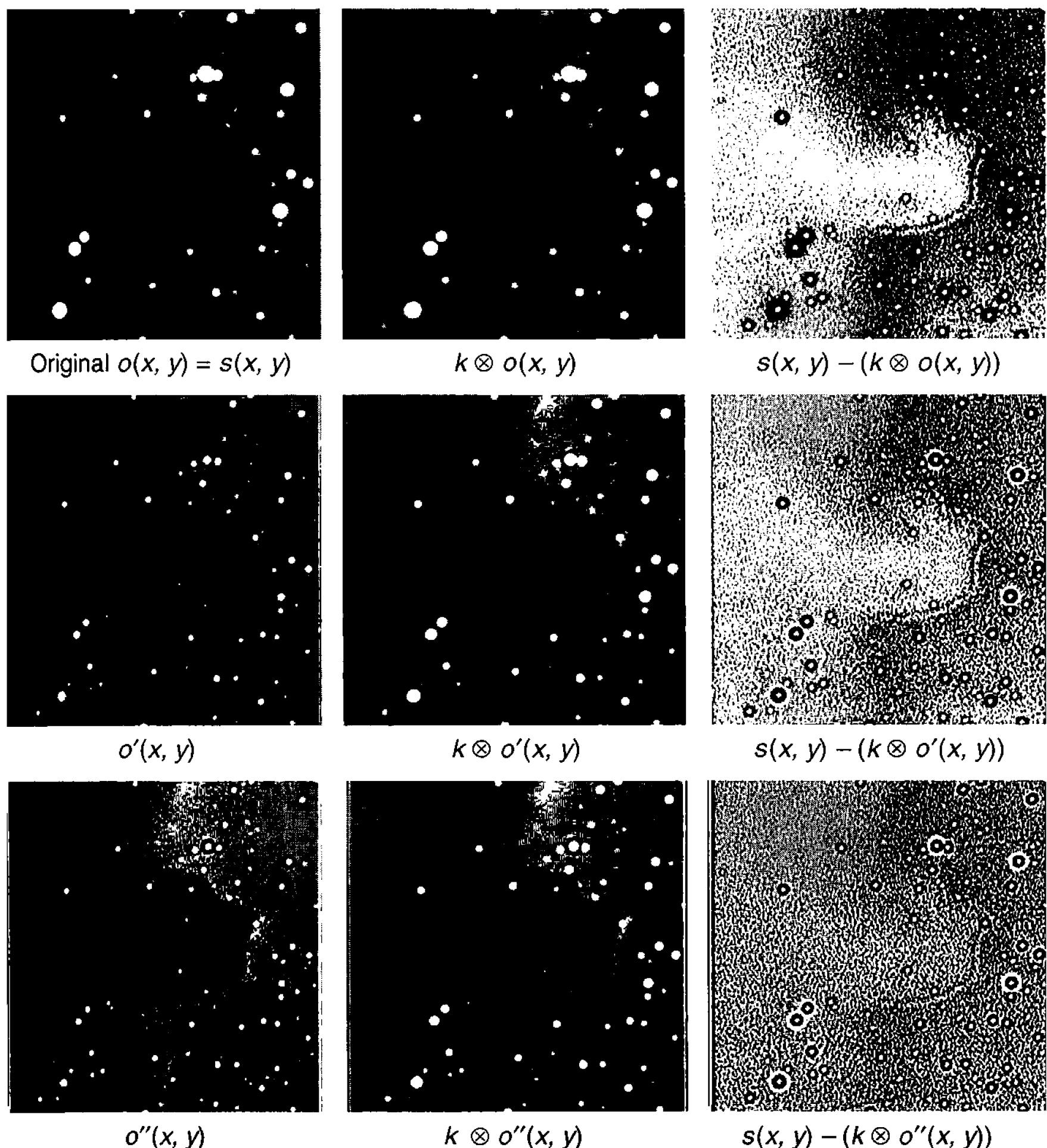
$$s(x, y) - e(x, y) = 0 . \quad (\text{Equ. 19.4})$$

This equation states that if we knew what  $e(x, y)$  was, there would be no difference (except for a negligible amount of noise) between the two images. This will remain true if we add the original image,  $o(x, y)$ , to both sides of the relationship:

$$o(x, y) = o(x, y) + (s(x, y) - e(x, y)) . \quad (\text{Equ. 19.5})$$

For the sake of clarity, we will designate the original image on the left side of the equation as  $o'(x, y)$ , and we replace  $e(x, y)$  with its definition:

## Section 19.2: Image Estimation by Iteration



**Figure 19.2** Here the mechanics of iterative image estimation are laid out in plain sight. The first column is the current estimate, the middle column the blurred version, and the right column the difference image. Note that the difference declines rapidly with successive iterations.

$$o'(x, y) = o(x, y) + (s(x, y) - (k \otimes o(x, y))). \quad (\text{Equ. 19.6})$$

What does this messy equation mean? Basically, it implies that by performing the operations on the right side of the equation, we can compute a new version of the original image. In other words, we can find a new image,  $o'(x, y)$ , by starting from  $o(x, y)$ , the old version of the original image. To get it, we take the original image and to it add the difference between the CCD image and the convolution of the original image with the point-spread function. This looks like the chicken-and-egg problem all over again, but not quite.

It is reasonable to think that if this relationship is valid for the original image,

## Chapter 19: Deconvolution

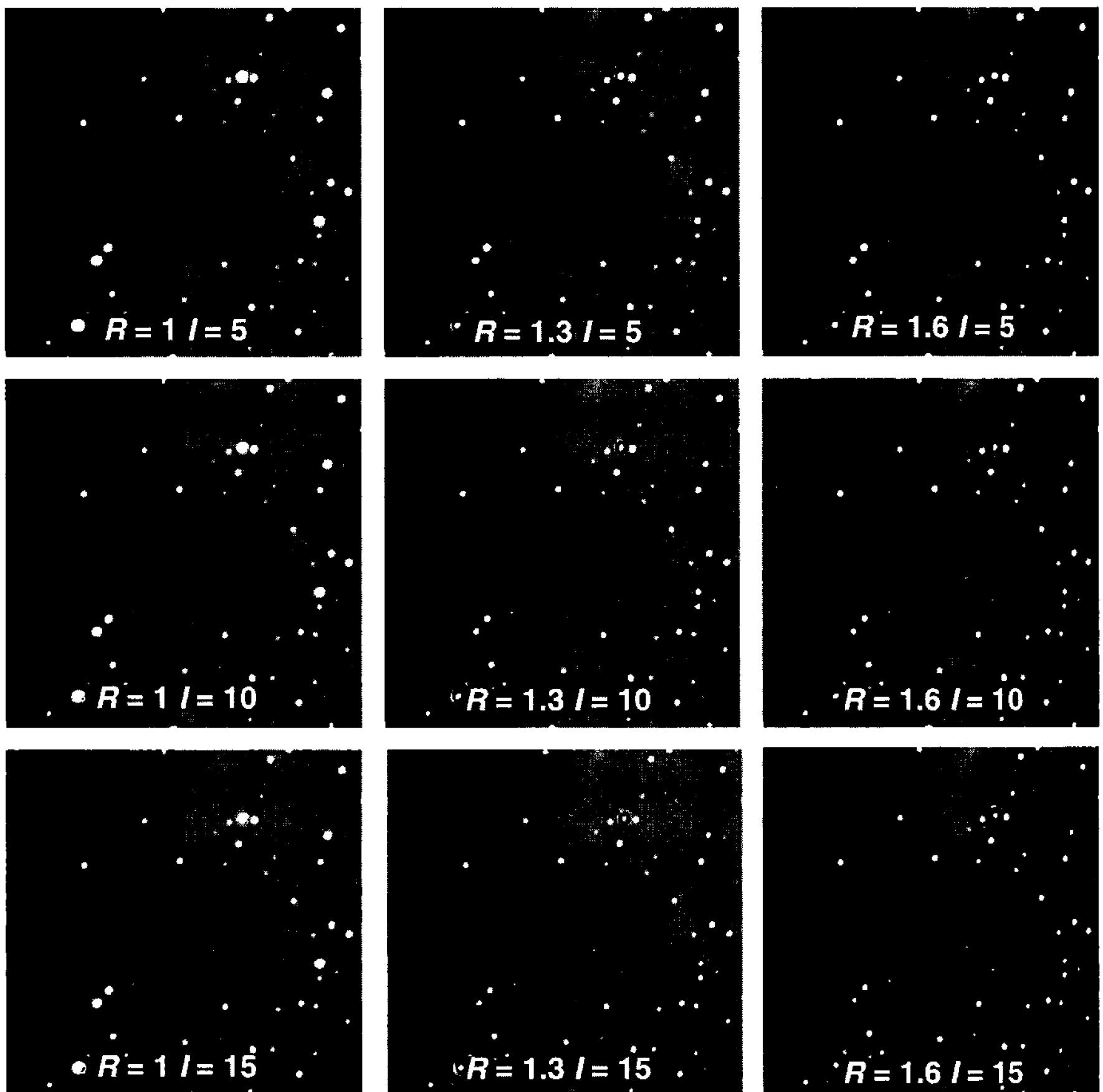


Figure 19.3 Results with van Cittert image estimation depend on the radius of the Gaussian point-spread function,  $R$ , and the number of iterations,  $I$ . A radius of 1.0 pixel is slightly too small, while 1.6 is slightly too large. The optimum number of iterations depends to some extent on personal preference.

it will also hold for an image that is similar to the original, but not exactly the same—that is, for similar inputs, we expect similar outputs. So, unless the CCD image has been terribly degraded, it is a fairly close approximation of the original one.

What happens if we simply decide that we'll compute a new version of the original image by treating the CCD image *as if it were* the original?

To accomplish this, we convolve the CCD image with the point-spread function, which generates a blurred version of it. We subtract this blurred image from the original CCD image, and obtain a difference image. Because the two initial images are similar, the difference image should contain small pixel values with both positive and negative signs. If we sum the difference image with that from the CCD, we will obtain a new version of the original image that is different from the one that we started with. So far so good.

## Section 19.2: Image Estimation by Iteration

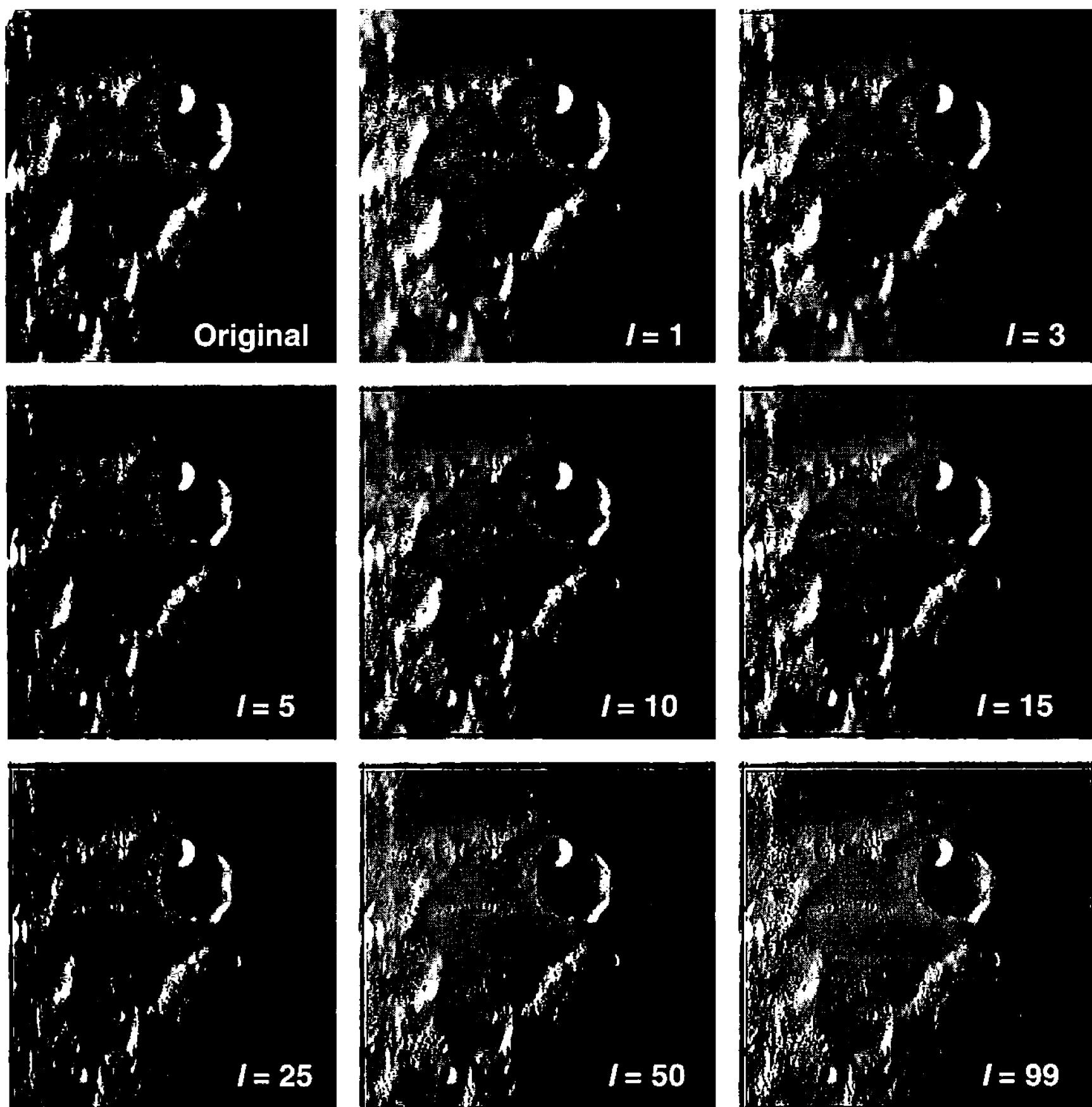


Figure 19.4 In this case study, watch the lunar crater Davy as it changes from the original image through 99 iterations. The best stopping point is subjective—a balance between enhancement and noise amplification—but most observers would probably agree that it's somewhere between 30 and 60 iterations.

The big conceptual leap is to feed the new version of the original image back into the same equation as a new old version, and to repeat this process over and over. If the process is well-behaved, at each iteration we should find an increasingly better approximation of the original image.

In other words, using the degraded CCD image to “seed” the process, we compute a new estimate of the original, and use that to compute a better estimate, and that to compute a still better estimate. When we satisfy the condition that  $s(x, y) - e(x, y) = 0$ , we will have recovered the original image.

This iterative method is so simple that anyone can try it using standard image-processing tools to blur, add, and subtract images. Figure 19.2 shows three successive approximations. Each new approximation appears sharper, and we see gratifyingly enough, that the amplitudes in the difference image become smaller with each iteration.

## Chapter 19: Deconvolution

Of course there is no guarantee that a series of approximations will converge until  $e(x, y)$  disappears. For one thing, we have ignored the noise which does have a finite value; and for another, we cannot be sure that successive difference images will not become larger and larger instead of smaller and smaller. In fact, the van Cittert and Richardson-Lucy deconvolution methods exist because we can ignore small amounts of noise and because their iterations do converge.

### 19.3 Van Cittert Image Estimation

Van Cittert deconvolution is one of the simplest and most robust methods of iterative image estimation. It was proposed by P.H. van Cittert in a 1931 paper on the influence of scattered light on the intensity distribution of spectral lines, and has been independently discovered and improved since, particularly by Landweber and Bialy, who often receive credit for a more general description of iterative image restoration than van Cittert's original paper.

Figure 19.5 At a larger image scale, the restoration of small features is dramatic. In particular, note the improvement in the visibility of the chain of tiny craters, the terraces and benches in the interior crater walls, and the contact between degraded crater walls and mare flood lavas.

The van Cittert iteration looks like this:

$$o'(x, y) = o(x, y) + w(s(x, y) - (k \otimes o(x, y))) \quad (\text{Equ. 19.7})$$

in which the terms have their previous meanings. The term  $w$  helps to insure that the iteration converges, and is called the relaxation parameter. When the relaxation parameter is set to 1, the van Cittert method becomes the iterative procedure described above.

The relaxation parameter allows the user to control the rate at which the iteration converges on the best approximation of the original image. To understand how this works, consider Equation 19.7 as two separate relationships, a correction term:

$$\text{correction term} = (s(x, y) - (k \otimes o(x, y))) \quad (\text{Equ. 19.8})$$

and the iterative relationship:

$$o'(x, y) = o(x, y) + w(\text{correction term}). \quad (\text{Equ. 19.9})$$

In any given iteration, the correction term will be the same because it does not include the relaxation parameter. However, if the relaxation parameter is less than one, only part of the correction term gets added to the best-estimate image, and the best-estimate will approach the original image more slowly.

Relaxation is more valuable when the single parameter is replaced by a relaxation function that sets a new value for each pixel based on its value:

$$w = w(o(x, y)). \quad (\text{Equ. 19.10})$$

By making relaxation a function of pixel value, the rate at which the iteration converges becomes different for different pixel values. The rate of convergence

## Section 19.4: Richardson-Lucy Image Estimation

should be slowed for the noisy low-pixel-value sky background areas, but allowed to run at full speed for the slow-to-converge low-noise pixel values that comprise bright star images.

One function that works particularly well is the sine function in the range between 0 and  $\pi/2$  radians:

$$w(p) = \begin{cases} p < p_{\text{black}} \rightarrow 0 \\ p_{\text{black}} \leq p \leq p_{\text{white}} \rightarrow \left( \sin\left(\frac{\pi}{2}\right)\left(\frac{p - p_{\text{black}}}{p_{\text{white}} - p_{\text{black}}}\right) \right)^{\gamma} \\ p > p_{\text{white}} \rightarrow 1 \end{cases} \quad (\text{Equ. 19.11})$$

where  $p$  is the pixel value of the current pixel and  $\gamma$  is a noise reduction parameter that the observer can set between 0 and 1 to control the shape of the relaxation parameter function. For values of  $\gamma > 0$ ,  $w(p)$  rises smoothly from 0 to 1, with the shape of the curve determined by the value of  $\gamma$ .

- **Tip:** In **AIP4Win**, the observer can select the point-spread function and radius for van Cittert deconvolution by matching a synthetic point-spread function against stars in the image, or by measuring real stars in the image.

## 19.4 Richardson-Lucy Image Estimation

The Richardson-Lucy iteration was described by William H. Richardson in a 1972 paper on iterative methods of image restoration, and two years later by L.B. Lucy in a paper in the *Astronomical Journal*. This method is usually called the Richardson-Lucy method, but in the literature on deconvolution you will also find it called the Lucy-Richardson method, the RL method, the expectation maximization method, and the method of maximum likelihood.

The van Cittert method is based on an additive correction; the Richardson-Lucy method is based on a multiplicative correction factor that approaches unity as the iteration runs. In its simplest form, the RL method works like this:

$$o'(x, y) = o(x, y) \frac{s(x, y)}{k \otimes o(x, y)} \quad (\text{Equ. 19.12})$$

where the terms have the same meanings as they did in the previous sections.

The correction term is the ratio between the original image and the blurred version of the image in its current iteration. Because the images are quite similar, the ratio begins close to unity and approaches unity as the iteration runs.

To retard the amplification of noise, it is possible to introduce a relaxation parameter. The relaxation parameter reduces the amplitude of the correction for low-value pixels but allows the restoration to proceed at full strength for high pixel values. The following maintains the value of the correction factor near unity:

## Chapter 19: Deconvolution

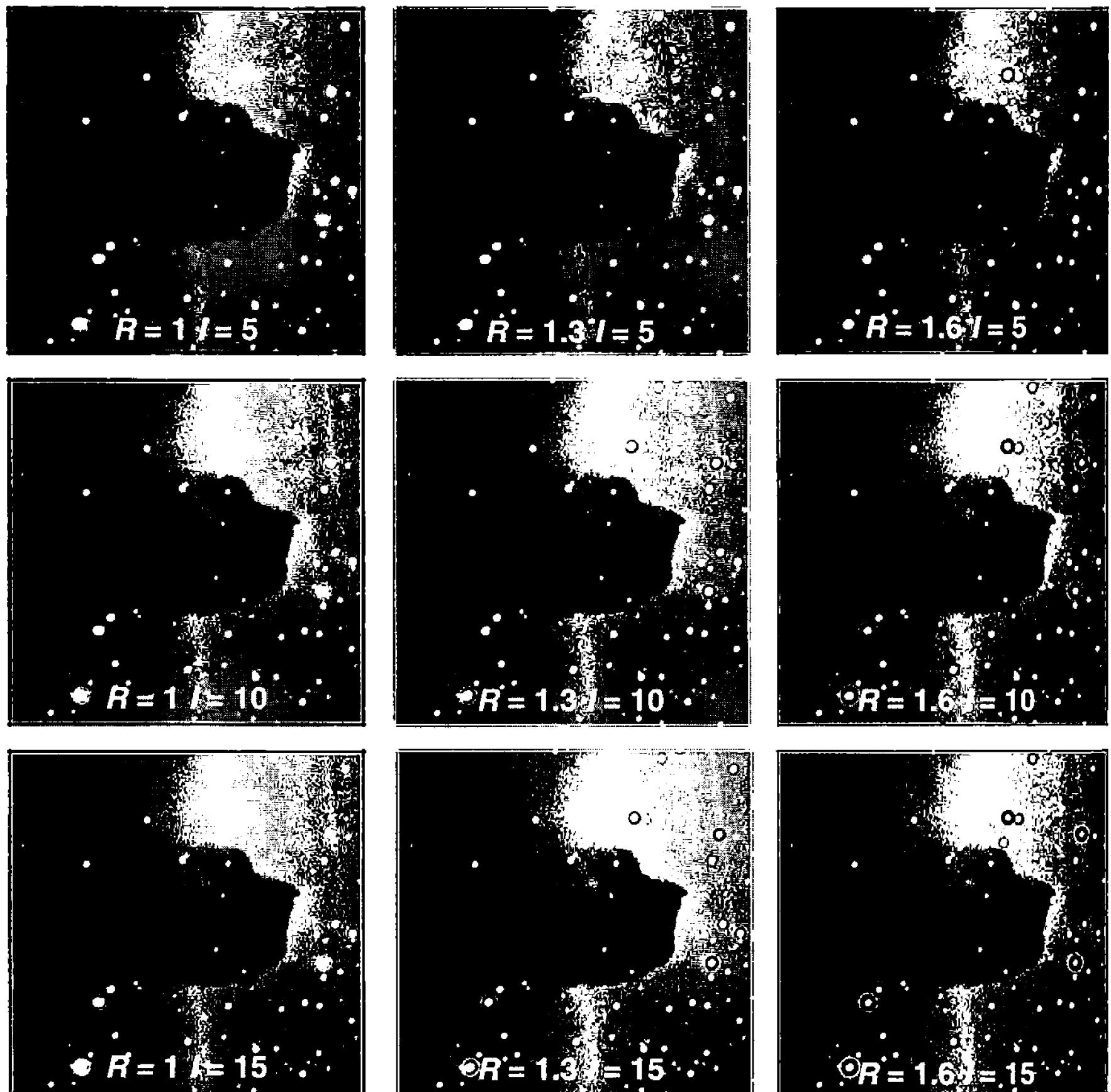
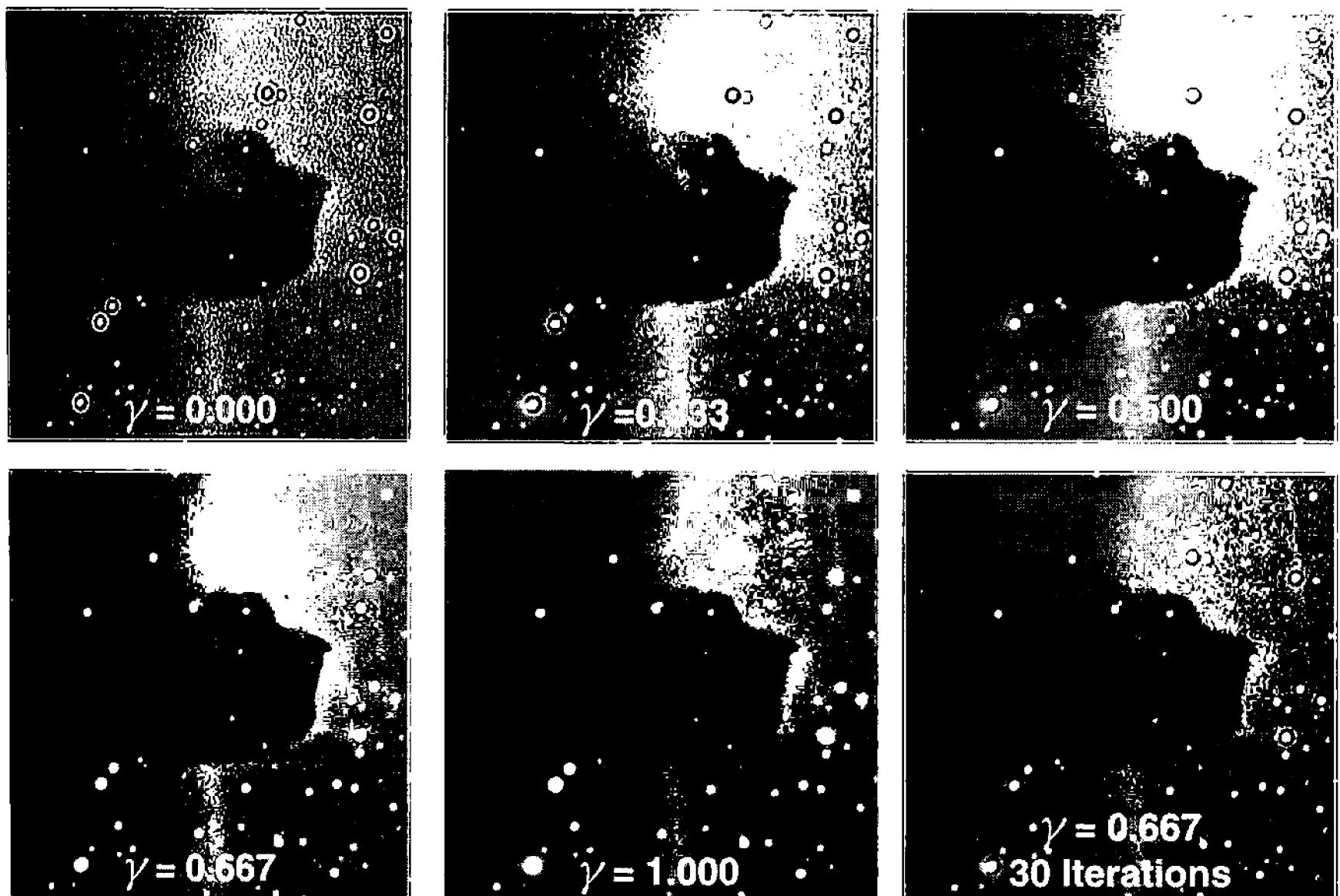


Figure 19.6 Like the van Cittert iterations shown in Figure 19.3, the Richardson-Lucy iteration is sensitive to the radius of the point-spread function and number of iterations. These two different methods of image estimation and correction converge in characteristically different ways and at different rates.

$$o'(x, y) = o(x, y) \left( w(o(x, y)) \left( \frac{s(x, y)}{k \otimes o(x, y)} - 1 \right) + 1 \right). \quad (\text{Equ. 19.13})$$

The relaxation function is a function of pixel value. For deep-sky images, the relaxation function for pixel values near the sky brightness should be close to zero, and for pixel values near the maximum in the image, should approach 1. The sine function, scaled so that  $\sin(0^\circ) = 0$  coincides with the pixel value  $p_{\text{black}}$  and  $\sin(90^\circ) = 1$  coincides with the  $p_{\text{white}}$  pixel value, is a good choice. The function can be set by the observer using a simple power law, such as  $(\sin \vartheta)^\gamma$ , with  $\gamma$  adjustable over the range 0 to 1. In this way, low values of  $\gamma$  reduce the effect of the relaxation parameter and values of  $\gamma$  approaching 1 increase the effect of the relaxation parameter.



**Figure 19.7** The relaxation function has a strong effect on the results produced by Richardson-Lucy deconvolution. Here  $\gamma$ , the noise reduction parameter, goes from 0.000 to 1.000 while  $R = 1.3$  and  $I = 10$  remain constant. The image at lower right shows tight star images with 30 iterations using  $\gamma = 0.667$ .

- **Tip:** *AIP4Win implements the Richardson-Lucy deconvolution with a user-selectable noise reduction parameter, which determines the relaxation parameter. Depending on the image quality and the relaxation parameter, between 10 and 100 iterations usually produce the best-looking results.*

## 19.5 Using Deconvolution with Astronomical Images

By their very nature, iterative methods of restoring degraded images are sensitive to noise in the image, the point-spread function, and how the telescopic image has been sampled as a digital image. Deconvolution is therefore most effective with images that have a high signal-to-noise ratio, a well-defined point-spread function, and are critically sampled and preferably oversampled.

In this section, we examine how the characteristics of the original image impinge on deconvolution and suggest strategies to improve the performance of the deconvolution process.

**Signal-to-Noise Ratio.** Noise is the overriding problem in astronomical CCD images. Even lengthy exposures seldom have a really good signal-to-noise ratio. Under dark sky conditions, accumulating a sky signal of 10,000 electrons may take 60 minutes of integration time. It will yield a signal-to-noise ratio as high as  $\sqrt{10,000} = 100$  due to Poisson statistics, but probably somewhat less with readout noise. Deconvolution is very sensitive to noise, and prone to amplify

## Chapter 19: Deconvolution

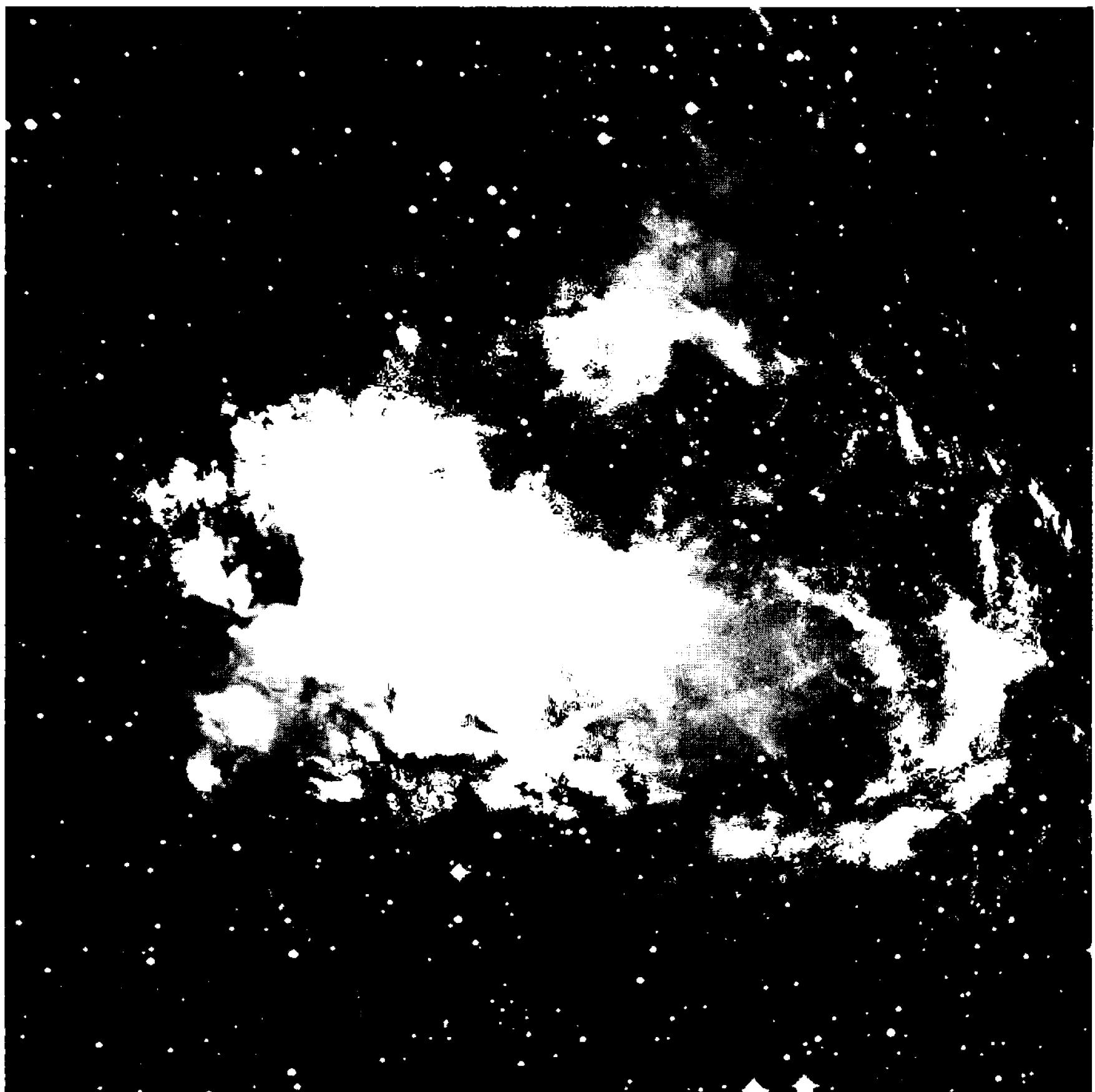


Figure 19.8 Enhanced nebular structure and compact star images are the result of applying deconvolution to the original image of the Omega Nebula through an H $\alpha$  filter taken by Jim Burnell. This example is the result of 64 iterations of the Richardson-Lucy (slow) algorithm.

bright and dark pixels. Images for deconvolution must be accurately calibrated and have a high signal-to-noise ratio.

**Point-Spread Function.** Although stars define the point-spread function in every deep-sky image, individual star images tend to be both distorted and noisy. If the point-spread function is simply a copy of a bright star, any small eccentricities in the star image will be applied repeatedly to the entire image. For deconvolution, a star image used for the point-spread function must be carefully massaged to obtain a smooth function, or approximated with a substitute such as a Gaussian distribution.

**Undersampled Original Images.** Astronomical images are usually sampled as pixels at or below the minimum focal ratio suggested by the Nyquist sampling theorem. As a result, information about the structure of the original image

## Section 19.5: Using Deconvolution with Astronomical Images

with higher spatial frequencies than the high-frequency cutoff imposed by the sampling interval is irrecoverably lost. For deconvolution to reach the highest spatial frequencies present in the telescopic image, both the image and the point-spread function must be sampled at twice the highest spatial frequency expected in the restored image.

**Proper Sampling for Restored Images.** If the original image is undersampled or critically sampled, then the restored image will also be undersampled—and probably severely so. In other words, if the new image is too small, there will be too few pixels in the restored image to display the information the iterative algorithm can extract. Before deconvolving undersampled and critically sampled images, resample them to at least twice the highest spatial frequency you expect in the restored image.

**Noise Amplification.** To a deconvolution algorithm, noise looks like detail that has been smeared and blurred to low amplitude. As the restoration algorithm runs, noise spikes that initially had small amplitudes grow, and the surrounding pixels decrease. The net result is that smooth areas such as the sky background take on a mottled or speckled appearance during deconvolution. To minimize the amplification of noise, use a noise suppression or relaxation function if one is available. Also, monitor the progress of the algorithm and stop iteration before noise becomes objectionable.

**Blocky Star Images.** Stars pose a severe test for any image estimation routine. Because they are point sources, star images should become considerably smaller than one pixel in the restored image. However, unless the star is perfectly centered in a single pixel, defining its location accurately between pixels requires that the restored image include four partially-illuminated pixels. If the deconvolution is optimized for extended regions, star images will probably look rough or blocky. Before deconvolution, resample the image for better-sampled stars, and after deconvolution, smooth the image with a Gaussian blur with a radius of 0.7 pixel.

**Dark Circles Around Star Images.** When deconvolution operates on a digital image containing stars, the cores of the star images become smaller. Because the total energy in the image should remain constant, the peak intensity in these star images may rise to values that exceed the highest allowed pixel values, and as a result, their peaks will be truncated. When the new image is blurred, these stars will have less energy than they should, and as a result, the total energy in and around the star image must be reduced by darkening the surrounding sky background. Dark circles can be minimized through careful selection of the point-spread function, and by subtracting the low-frequency components of the image, consisting primarily of the sky background, before deconvolution.

## Chapter 19: Deconvolution

---

## 20 Building Color Images

---

In this chapter, we discuss the key technical issues involved in making astronomical images in color. Today there are two basic routes to making color images: by combining filtered images made with a monochrome sensor, or by processing color data from a one-shot filter-matrix sensor.

In making color images, it is crucial to recognize color as a product of the human visual system. When light falls on the retina, all the subtle wavelength discriminations in an object's spectrum are reduced to just three cellular responses: one each for long, medium, and short wavelengths. We call these sensations red, green, and blue. The eye and brain combine these responses to produce the full range and subtlety of the colors that we see.

Color imaging is based on the proposition that by using a CCD camera to split the spectrum into the same sections as does the human eye, we can reproduce the same color sensations that the eye sees. In this chapter, we examine the operation of the human visual system and the distribution of light energy from astronomical objects as a function of wavelength. It is the *distribution* of light energy (the spectrum) rather than the total energy (brightness) that determines the apparent color of astronomical objects.

Splitting the spectrum into three segments sounds simple, but a host of devils reside in the technical details. Everything from the passbands of the filters used to split the spectrum to the atmosphere over our telescopes influences the final color product, sometimes in obvious ways and other times in subtle ways. As we shall see many such problems can be surmounted by using sun-like stars to correct the "raw" data from the CCD camera into accurate color images.

Furthermore, astronomical color imaging presents unique problems because the range of brightness encountered in astronomical images is enormous, but output devices—such as computer monitors, ink-jet printers, and the printed page—offer only a very restricted range of brightness.

In this chapter we discuss how the spectrum is split apart and recombined to produce a color image, and how by separating the luminance component (brightness) from the chrominance component (color) we can solve the dynamic range issues, to produce exquisite color images of the heavens.

## 20.1 Human Color Vision

Although celestial objects emit light over a range of wavelengths, it takes the human eye to perceive color in the Universe. Because it enables us to distinguish subtle features of the natural world, color vision is valuable to humans. Color vision is the result of comparing and ratioing the signals from the different types of cone cells in the retina and in the brain.

However, the very low light levels typically encountered in astronomy fall below the threshold of vision for cone cells. Instead, low-light-level vision depends on their monochrome counterparts, the rod cells. It would serve little purpose to make “realistic” color images showing objects in shades of gray or dark greenish gray. Therefore, in making astronomical color images, our goal is to recreate the colors that we would see *if celestial objects were bright enough to stimulate full color vision*. Thus astronomical color images have the ability to reveal a Universe that we can never see directly.

### 20.1.1 The Trichromatic Basis of Color Vision

Human color vision is *trichromatic*; that is, the retina contains three different types of cone cells, each of which detects a specific range of wavelengths. *Beta* cone cells respond most strongly to light between 400 nm and 500 nm; *gamma* cones to light between 450 nm and 620 nm; and *rho* cones most strongly to light between 480 nm and 660 nm, but their response extends to about 740 nm. This is an important point: we do not see the spectrum of incoming light; we detect only the responses of the three types of cone cells.

Because of this, it is not necessary to reproduce the spectrum of the light source in its full complexity. Rather, we need only capture images at wavelengths that correspond to the wavelength sensitivity of the eye’s cone cells. Then, by playing those images back into the eye using light in the three colors that excite those same cells, we can reproduce the identical color sensation as the original light source—even though the spectrum of the light is not the same.

One key to the successful recreation of color images is choosing filters that isolate the right wavelength bands in the spectrum. Another key is accurate processing of the filtered images to correct for atmosphere extinction and small errors in the choice of filters. Finally, the output device we use to view the color image must emit a spectrum that recreates, in the eye of the beholder, the sensations that the original celestial object might have produced.

### 20.1.2 Luminance and Chrominance

Although the underlying mechanism of color vision involves three color information channels, people seldom discuss their visual sensations in terms of three additive primary colors. Instead, they describe light in terms of its brightness and color. The brightness of an object is its *luminance* and the color is its *chrominance*.

Luminance varies over an enormous range. The luminance of the Full Moon is roughly a billion times greater than the faintest galaxies we can glimpse in a

**Table 20.1 Wavelengths of the Colors**

Color	Wavelengths
Ultraviolet	below 380 nm
Violet	380 nm to 450 nm
Blue	450 nm to 480 nm
Blue-green	480 nm to 510 nm
Green	510 nm to 550 nm
Yellow-green	550 nm to 570 nm
Yellow	570 nm to 590 nm
Orange	590 nm to 630 nm
Red	630 nm to 700 nm
Infrared	700 nm and above

Note: Color boundaries are subjective; colors blend gradually into one another.

telescope, and our vision functions quite well over that entire range. At low light levels—using rod-cell vision—we can't see the colors of objects, but we do see their luminance. At high light levels—with cone vision fully operational—our brain perceives and responds to brightness before it notes color. Much as we love color, humans are hard-wired to respond first and foremost to brightness.

The color quality of light is its *chrominance*. Nerve networks in the retina compare signals from the three types of cone sensors. They find the relative signal strengths and convey these differential measures to the brain, which interprets them as color. Color is therefore a measure of the *relative* amounts of energy in the three wavelength bands sensed by the cone cells. With only three stimuli—red, green, and blue—the eye can easily distinguish thousands of colors.

The visual sensation of chrominance remains constant over a wide range of luminance. People describe a red-filtered flashlight, for example, as having the same chrominance (“It looks *red* and very faint”) when it is far off and faint or close and nearly blinding (“It looks *red* and very bright”). To the eye, the blue-white star Vega appears blue-white to the naked eye and also appears blue-white in a 24-inch telescope—even though the telescopic image is 10,000 times brighter. Across a range of luminance from the threshold of color perception, to brightness nearing the pain threshold, perception of the chrominance in light is remarkably constant.

Artists, photographers, and other people who work daily with color break chrominance into two components: *hue* and *saturation*. Hue is the color component of chrominance—red, orange, yellow, green, blue, violet—while saturation is the intensity or strength of the color. Pink is a red hue at low saturation; chartreuse is a yellow-green hue with high saturation. Hue and saturation taken together comprise chrominance.

Because humans perceive the world around them in luminance and chrominance, they are potentially powerful tools in image processing.

## Chapter 20: Building Color Images

### 20.1.3 Reproducing Color

The goal in making astronomical color images is to recreate the appearance of celestial objects, with the important caveat that we want to show them as they would appear if our eyes could perceive color at low light levels. This is an ambitious and admirable goal.

When we view these images through the medium of hard copy or a computer screen, we encounter two major challenges:

1. Poisson noise and readout noise introduce distracting artifacts in the color image, especially at the lowest and highest luminance levels.
2. The range of luminance typically found in astronomical scenes is vastly greater than the range of luminance that cathode ray tubes, ink and paper, and liquid crystal displays can generate.

The wide range of luminance is a difficult problem indeed. Computer displays typically generate a luminance range between 60:1 and 100:1—but deep-sky images seldom span a luminance range less than 1000:1. Hard copy poses even greater limitations; between the black ink on paper (4% reflectivity) and the whitest paper (80% reflectivity), the displayable range of luminance is a mere 20:1. To display the full range of luminance in astronomical images, therefore, it is necessary either to:

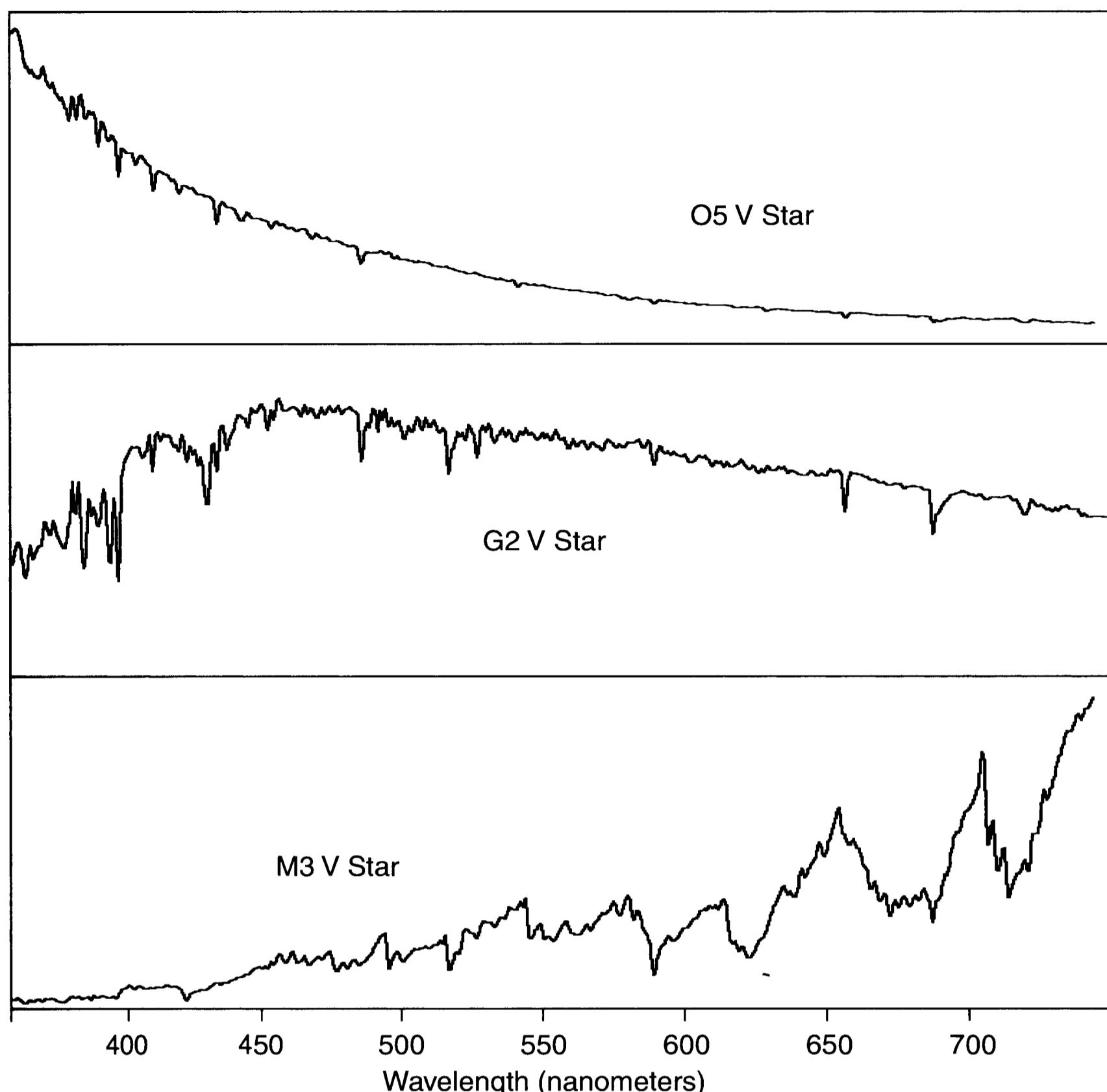
1. Discard luminance information at the dark end, the bright end, or from both ends of the luminance range, or
2. compress the natural range of luminance to a range that displays can handle.

Fortunately, discarding low and high pixel values and compressing a side range of pixel value into a smaller range without losing information are among the tasks that image processing handles best.

## 20.2 Celestial Light

Visible light consists of electromagnetic radiation with wavelengths ranging from 390 nanometers (deep violet) to 780 nanometers (deep red), or alternatively, in terms of photons with energies ranging from 1.6 electron-volts (deep red) to 3.2 electron-volts (deep violet). Within the rather narrow band of wavelengths that comprise visible radiation, three physical mechanisms generate light: black-body radiation, electronic transitions, and synchrotron emission.

**Black-Body Radiation.** In hot, dense bodies, atoms collide with one another, gaining energy that excites their outer electrons to higher energy states. When these electrons return to a lower energy state, the energy previously gained is given off as a photon. Because the atoms are tightly packed and colliding frequently, photon energies form a smooth distribution with wavelength. The filament of a light bulb is a hot, dense object, and it produces a continuous spectrum or black-body spectrum, with a distribution of photons that is continuous in wavelength.



**Figure 20.1** The energy distribution in stellar spectra differs greatly. The top panel shows the spectrum of a pale-blue class O5V star. The middle spectrum is that of a Sun-like class G2V star, white light to the human eye. The bottom spectrum is a yellow-orange class M5V star. Dips are absorption lines and bands.

Stars are also hot, dense objects, and their light is distributed across a wide range of wavelengths. However, the hot, dense layers of stars are surrounded by cooler, less dense gas. As photons pass through the stellar atmosphere, they strike atoms of this cooler gas and excite electrons from well-defined low energy levels into well-defined higher energy levels. As a result, photons that have the right energies to excite atoms are removed from the stars' light. The spectrum of starlight is therefore a continuous spectrum with gaps, or lines, and is called an absorption spectrum.

**Electronic Transitions.** In the rarefied gas between the stars, atoms seldom encounter an ultraviolet photon. When one does, however, an outer electron can be excited to higher energies. The atom may remain in an excited state for many seconds, but when it finally moves to a lower well-defined energy state, it emits a photon with a well-defined energy. Because a limited number of energy states ex-

## Chapter 20: Building Color Images

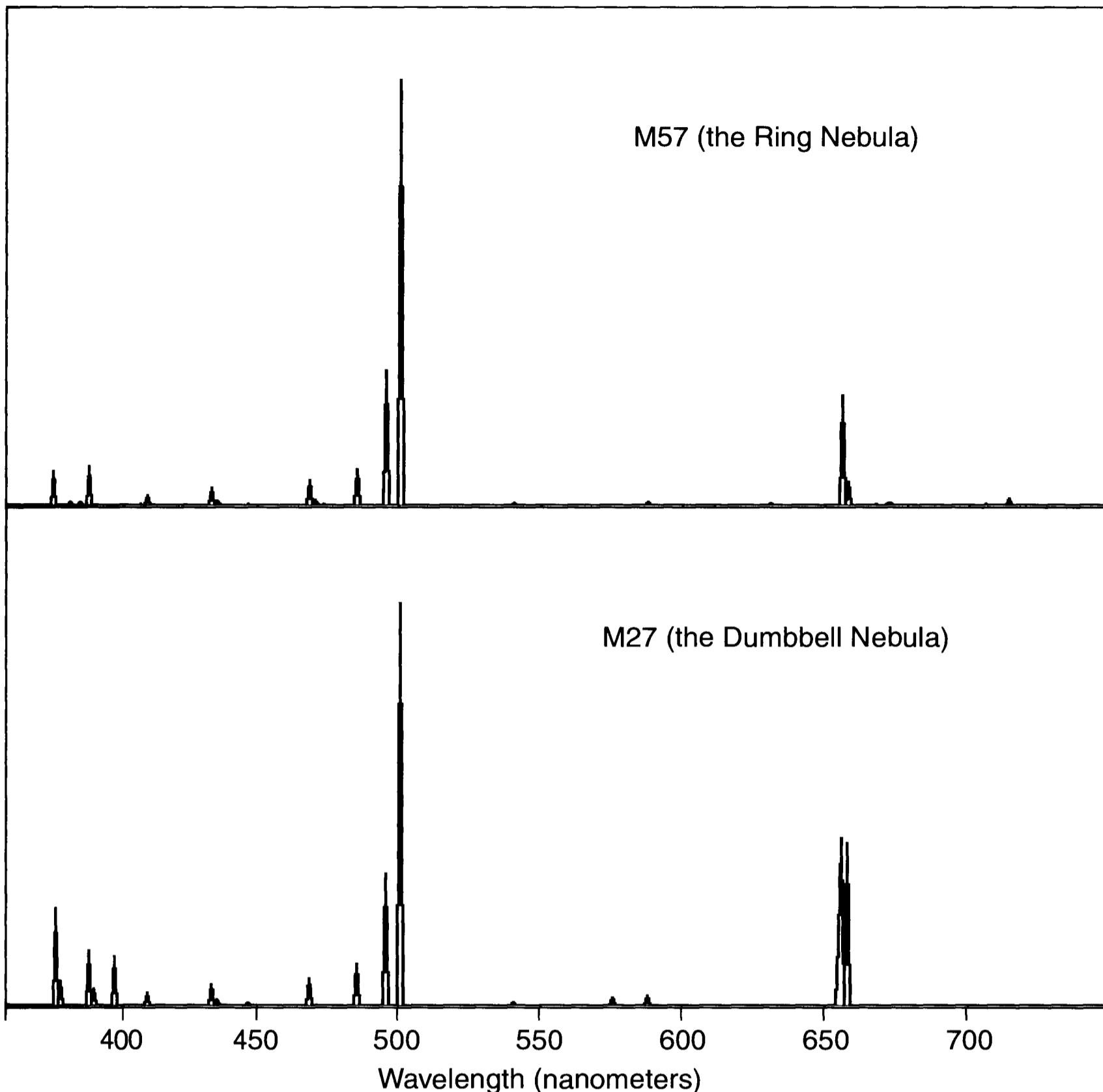
**Table 20.2 Spectral Lines in the Nebulae**

Color	Wavelength	Element	Strength
Ultraviolet	372.7 nm	[OII]	medium
Ultraviolet	386.9 nm	[Ne III]	medium
Deep violet	388.9 nm	H I	(Hε)
Violet	396.8 nm	[Ne III], H I	
Violet	410.1 nm	H I, N III	(Hδ)
Violet	434.0 nm	H I	medium (Hγ)
Blue-violet	436.3 nm	[O III]	
Blue	468.6 nm	He II	
Blue-green	486.1 nm	H I	strong (Hβ)
Blue-green	495.9 nm	[O III]	strong
Blue-green	500.7 nm	[O III]	very strong
Yellow	575.5 nm	[N II]	
Yellow	587.6 nm	He I	
Red-orange	630.2 nm	[O I], [S III]	
Red	654.8 nm	[N II]	strong
Red	656.3 nm	H I	very strong (Hα)
Red	658.3 nm	[O I]	strong
Deep red	667.8 nm	He I	
Deep red	672.6 nm	[S II]	
Near infrared	706.5 nm	He I	
Near infrared	713.6 nm	[Ar III]	

ist, the photons from the gas exhibit only a small number of wavelengths. Light from a thin cloud of hot hydrogen gas, for example, consists almost entirely of the wavelengths 656.3 nm, 486.1 nm, 434.0 nm, and 410.2 nm. Each type of atom emits a characteristic spectrum, which is how astronomers know which elements make up the nebulae. Spectra of this type are called line, or emission, spectra.

**Synchrotron Emission.** Synchrotron radiation occurs in thin, hot gases in the presence of strong magnetic fields. As the electrons are forced to accelerate in the magnetic field, photons are emitted. These conditions occur in pulsar-powered supernova remnants like the Crab Nebula. The energy distribution of photons in synchrotron emission is continuous.

The spectrum of an object influences our strategy for imaging it, particularly the choice of filters. It is straightforward to select filters to separate the continuous spectrum of a star or galaxy into three color bands, but emission spectra pose significant problems because the filters can either miss or “mis-classify” photons. In the next two sections, we examine the interesting problem of selecting filters for imaging astronomical objects.



**Figure 20.2** The spectra of gaseous nebulae are entirely unlike those of stars. Energy output occurs in narrow spectral lines corresponding to energy-level transitions in particular atomic species. The relative strengths of the lines differ, accounting for variations in the colors of these planetary nebulae.

### 20.2.1 Starlight and Continuous Spectra

Celestial objects such as stars, galaxies, and planets (which shine by reflected starlight) have well-behaved spectra. The energy we receive from them changes smoothly with wavelength. Although absorption lines are present, they do not change the gross structure of the spectrum. To a good approximation, therefore, the energy emitted at a given wavelength by a star follows the classic Planck “black-body” curve:

$$B_\lambda(T) = \frac{2hc^2/\lambda^5}{e^{hc/kT} - 1} \quad (\text{Equ. 20.1})$$

where  $B$  is the energy emitted per unit wavelength,  $T$  is the temperature of the body in Kelvins,  $\lambda$  is the wavelength, and  $h$ ,  $c$ , and  $k$  are fundamental physical constants. The distribution of energy rises rapidly from short wavelengths, peaks at a

## Chapter 20: Building Color Images

specific wavelength, and falls slowly toward longer ones. The maximum emission occurs at the wavelength:

$$\lambda_{\max} = 2.90 \times 10^6 / T \text{ [nm].} \quad (\text{Equ. 20.2})$$

The temperature of the Sun's surface is 5770 Kelvins, so the distribution of energy in its spectrum peaks at a wavelength of 503 nm, i.e., in bluish-green light. The Sun's energy output is not sharply peaked, and declines gradually at longer and shorter wavelengths. In deep blue light (400 nm) and deep red light (700 nm) the spectral energy is still about 80% of its peak value. Sunlight is rich in energy across the entire visible spectrum. Human eyes have evolved to see this particular energy distribution as white. By definition, sunlight is white light.

Other stars have surface temperatures much higher or lower than that of the Sun. Extreme class O stars have temperatures of 44,500 Kelvins, so the peak output occurs in the deep ultraviolet. In the visible part of the spectrum, the blue output of an O star is roughly ten times its output in red.

At the opposite extreme, class M supergiants have a temperature of 2600 Kelvins, so the spectrum peaks in the infrared. In visible light, the red output of an M supergiant exceeds that in the blue by more than a factor of ten.

To the eye, however, these extreme examples appear less strongly colored than you might expect. An energy ratio of 10:1 means that the red, green, and blue cone cells all receive significant signals. Furthermore, for objects above 15,000 Kelvins, we see only the ascending portion of the black-body curve, so its slope remains the same *regardless of temperature*. The light of all hot stars resembles the bluish-white light of a carbon arc lamp.

Light from the very coolest red stars is approximately as "red" as a standard household light bulb running at 2880 Kelvins. Even very red stars emit light that is only slightly more "yellow" than a household light bulb. For comparison, orange campfire light corresponds to a black body at about 1100 Kelvins.

Stars have colors that range from pale blue through pale yellow to a medium orange-red for the very coolest. Star colors appear fairly weak because a star's radiation is spread across the entire visible spectrum. Sunlight excites the rho, gamma, and beta cones with equal intensity; its color is pure white. A star such as Sirius, at a temperature of 10,000 Kelvins, excites the beta (blue) cone cells most strongly, but it also excites the rho (red) and gamma (green) cones. We perceive the star as white light mixed with extra energy in the blue and green, producing the sensation of pale blue, that is, blue diluted with white light. Likewise, the light of a cool star consists of all three primaries plus a little extra green and a fairly large amount of red. The eye perceives this as a pale orange, that is, pure orange diluted with white.

Of course, the stars do not have purely black-body spectra. Absorption of light by atoms and molecules in their outer layers causes stellar spectra to depart from the black-body curve. However, starlight is distributed across the entire visible spectrum in a way that is fairly smooth and predictable.

Galaxies consist of hundreds of billions of stars, so their light and colors are also well behaved continuous spectra. In the gassy arms of the open type Sc spirals, where stars are still forming, light is dominated by hot, blue stars. In the splayed-out arms of tightly wound Sb and Sa spirals, the bluest stars have already exploded as supernovae, so the dominant color comes from the longer-lived bluish class A and F stars plus a scattering of red giants. The resulting composite spectrum appears cool white to slightly yellowish in color. The aging stellar populations found in ellipticals and the central bulge of spirals consist of vast hordes of cool, sun-like stars and red giants, which range in color from white through pale yellow to distinctly orange.

The Moon and planets shine by reflected sunlight. Although surfaces reflect light of different wavelengths differently, most of the minerals, clouds, and gases that we see have reasonably well-behaved reflectance spectra in visible wavelengths. Only in the deep red and near infrared light from the giant planets does molecular absorption produce complex and rapidly changing absorption spectra. In visible light, the spectra of the Moon and planets are smooth modifications of the solar spectrum.

### 20.2.2 Nebulae and Emission Spectra

The spectra of nebulae consist of spectral lines; that is, most of the light is concentrated into a small number of wavelengths. (Compare the stellar spectra in Figure 20.1 with the nebular spectra in Figure 20.2.) Broadly speaking, nebular spectra fall into two classes:

- **H II Regions:** Extended, low-excitation nebulae that fluoresce because of ultraviolet light given off by newly formed stars in the nebula or nearby. The Lagoon Nebula is a good example.
- **Planetary Nebulae:** Compact, high-excitation shells of gas thrown off by aging stars and still excited by the very hot core of the parent star. The Dumbbell Nebula is a good example.

H II regions are vast, warm clouds of interstellar gas on the outskirts of star forming regions. Ultraviolet light from young, massive stars knocks the electrons from atoms in the surrounding gas clouds, and when these electrons recombine with atoms, each element produces a characteristic pattern of spectral lines. Because hydrogen is so abundant in interstellar gas, the most prominent spectral lines in H II regions are those of hydrogen, with the red H $\alpha$  line at 656.3 nm dominating, followed by H $\beta$  at 486.1 nm in the blue, and the H $\gamma$  line at 434.0 nm in the violet. These nebulae glow with a characteristic “electric pink” color, the result of mixing deep red with blue and violet.

Planetary nebulae are shells of gas ejected by red giant stars, heated to temperatures between 10,000 and 20,000 Kelvins by the light of the central star. Because they are hot, the nebular spectrum contains many lines from doubly and triply ionized atoms. In addition to the hydrogen spectrum, strong lines of helium, oxygen, nitrogen, neon, and argon are present. Because the temperature and den-

## Chapter 20: Building Color Images

sity of the gas varies from the center to the outer rim, the relative strengths of the lines vary, with the spectral lines of the more highly ionized elements dominating emission from the center, and the lines of less ionized and neutral atoms dominating the outside.

### 20.2.3 The Challenge of Celestial Color Imaging

Astronomical imaging in color faces two significant problems, one technical and the other perceptual. The technical challenge arises because light from the nebulae is concentrated in a small number of spectral lines. Small errors in filter choice can generate large changes in the final color product. The perceptual challenge arises because as amateur astronomers, we are flooded with color images from many different sources. However, the color pictures that we see are not necessarily accurate; our perception of how celestial objects “ought to look” is thus influenced by images that are often not accurate and sometimes blatantly false.

**The Technical Challenge.** The Dumbbell Nebula in Vulpecula is probably the most notorious example of an object that poses a severe technical challenge. Its spectrum is typical of high-temperature nebulae, and in the central regions of this object, the 500.7 nm line of doubly-ionized oxygen (OIII) dominates. To the eye, 500.7 nm appears neither green nor blue, but a distinct shade of greenish-blue or bluish-green called teal.

To make a color image of the Dumbbell, you need a blue filter and a green filter. Blue filters pass light between 400 and 500 nm, and green filters pass light between 500 and 600 nm. If both filters have the same transmittance at 500.7 nm, the spectral line will be recorded properly, and in the resulting image, the central regions of the Dumbbell will appear teal in color.

However, if either the blue or green filter has a higher transmittance at 500.7 nm than the other, then the corresponding filtered image will show that pass-band too strongly, and in the resulting image the center parts of the nebula will appear distinctly green or distinctly blue.

**The Perceptual Challenge.** Because the color-sensing cone cells in the retina do not respond to low light levels, deep-sky observers see a universe devoid of color. Virtually every sensation that observers perceive comes from the rod cells, which respond most strongly to blue and green light, but generate no color sensation. At levels corresponding to the very brightest nebulae, the green-sensing cone cells respond slightly. As the light level increases, the red-sensing cone cells come into play, and at higher levels still, the blue-sensing cones kick in. Observers fairly consistently report that nebulae appear gray or greenish in small telescopes, or greenish with hints of pink with large telescopes.

At the other extreme, however, amateur astronomers are confronted by glowing scarlet nebulae and highly processed enhanced-color images of galaxies showing vivid blue spiral arms studded with ruddy HII regions. These magnificent images shape our expectations: the Universe is alive in vivid color! But visual observers *know* that to the eye, deep-sky objects appear gray or pale greenish-gray.

### Section 20.3: Red/Green/Blue Tri-Color Imaging

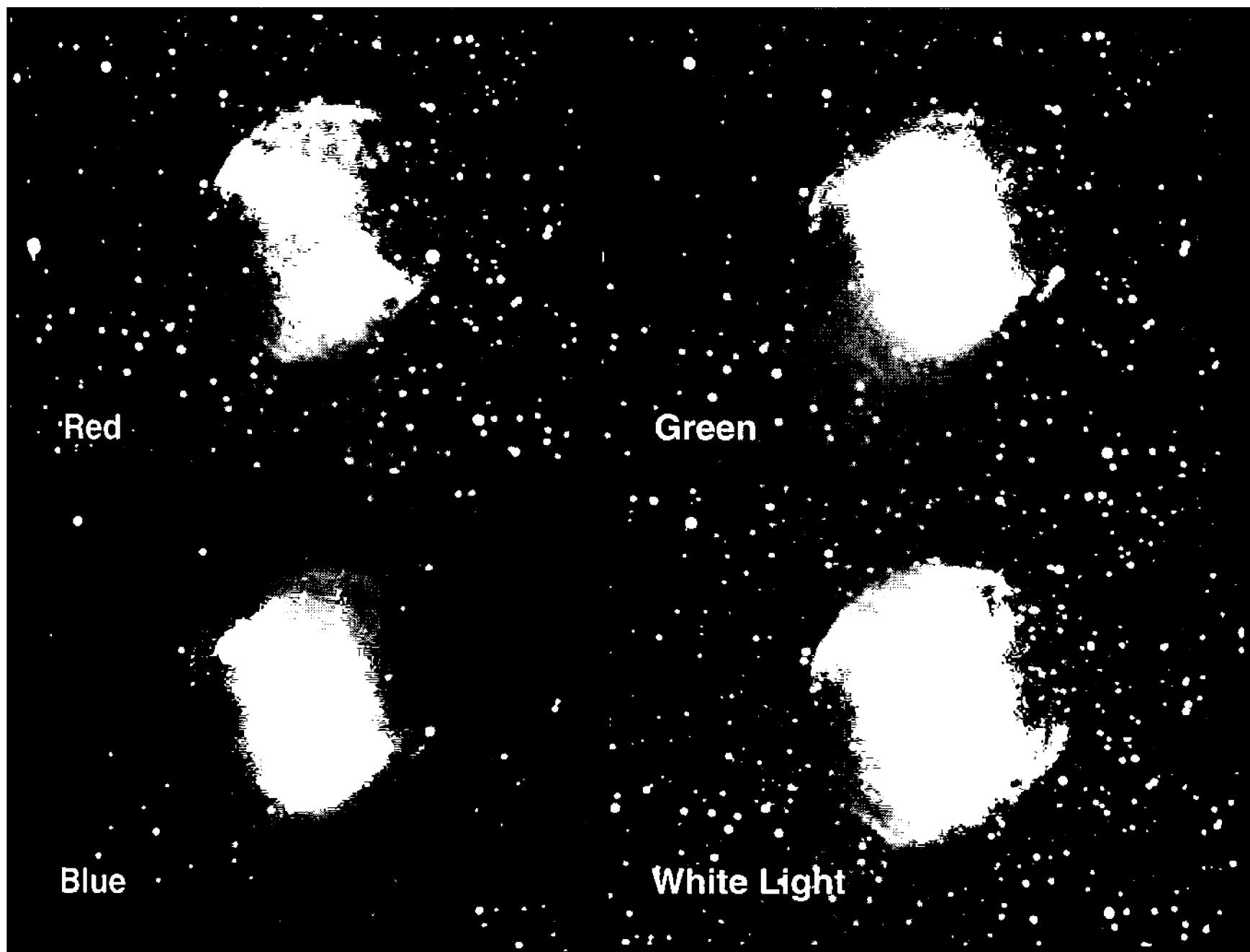


Figure 20.3 Filters pass different spectral lines, recording the appearance of objects in different parts of the spectrum. A red filter, for example, passes the red H $\alpha$  line and blocks the blue-green OIII line, while the green and blue filters both pass the OIII line and block H $\alpha$ . Images of M27 by Rob West.

That, they say, is the *real* reality!

The perceptual challenge consists of convincing people that carefully crafted astronomical color pictures can be—indeed, must be—true to nature. Galaxies glow in pastel shades of blue, yellow, and orange, while nebulae fluoresce in clear (but not Technicolor) shades of red, green, and blue. Combining knowledge and technical skill does indeed lead to accurate images of the Universe.

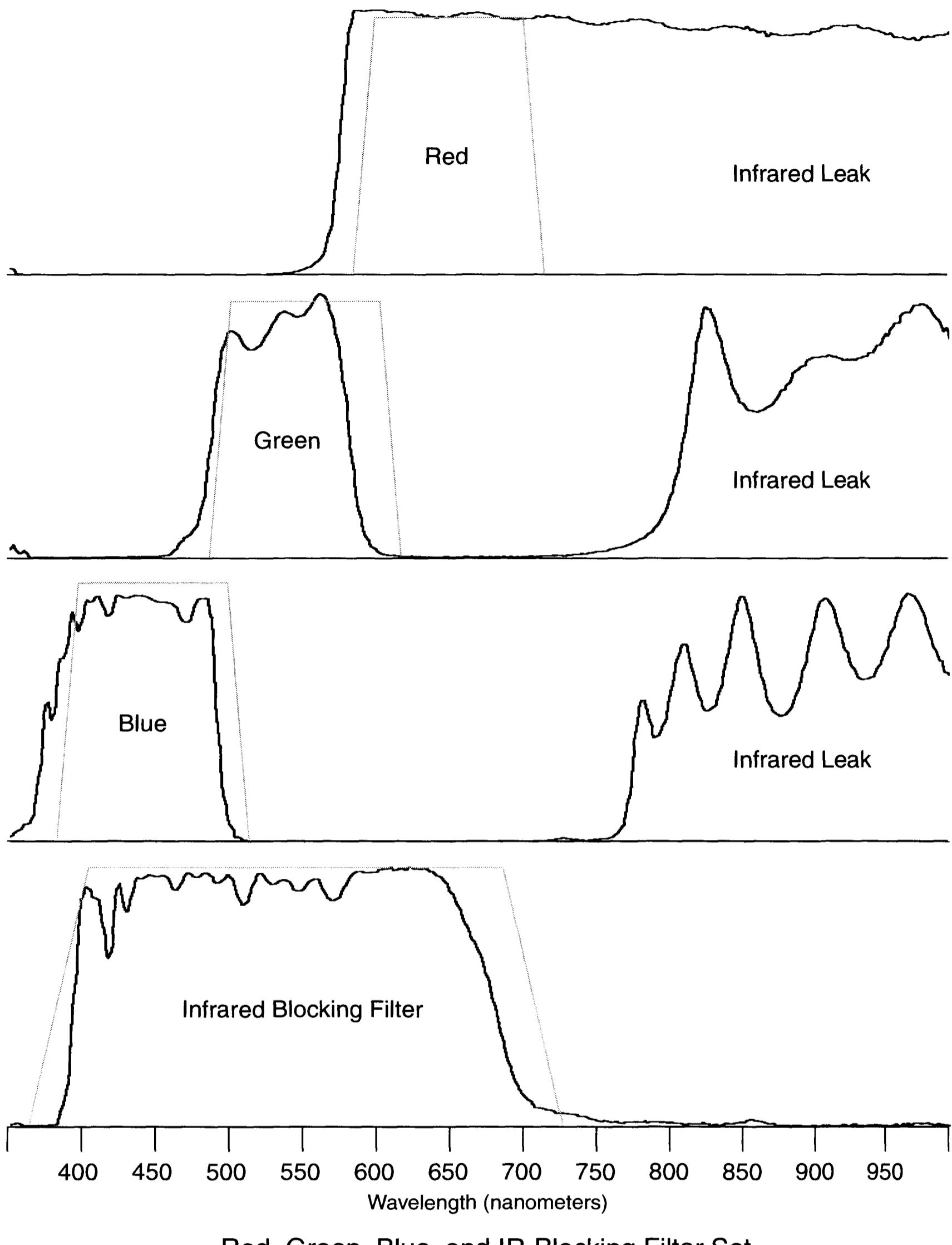
## 20.3 Red/Green/Blue Tri-Color Imaging

This section describes classic red, green, and blue color filter imaging—RGB tri-color imaging—as the basis for understanding more sophisticated color capture, correction, and display methods covered later in this chapter.

Color imaging consists of three steps:

- Capture filtered images. A basic set consists of red, green, and blue filters. Images taken through each of the color filters, and appropriate dark frames and flat frames are taken as well. After dark subtraction and flat-fielding calibration, we designate the set of filtered images as  $S_R$ ,  $S_G$ , and  $S_B$ .

## Chapter 20: Building Color Images



Red, Green, Blue, and IR-Blocking Filter Set

Figure 20.4 Color separation filters divide the white-light spectrum into well-defined wavelength regions. Solid curves show the transmittance of actual filters; the gray lines show “ideal” transmittance curves. Many red/green/blue filters sets require an extra infrared blocking filter to stop their infrared leak.

- Correct the filtered images. Filtered color images retain the effects of filter passbands and transmittances, CCD quantum efficiency, atmospheric attenuation, and background sky light.

## Section 20.3: Red/Green/Blue Tri-Color Imaging

We can measure and correct for these effects. After correction, we designate the images as  $S_R$ ,  $S_G$ , and  $S_B$ .

- Create a color image. The final stage in this process consists of scaling the corrected image values into the storage range of an image file or the display range of a computer monitor. After scaling, we designate the color display components (now usually called *color channels*) as  $D_R$ ,  $D_G$ , and  $D_B$ .

We now examine each of these steps in detail.

### 20.3.1 Step 1: Capture Filtered Images

To make color images, it is necessary to acquire images in well-defined portions of the spectrum. Although color filters do isolate portions of the spectrum, the light from the celestial object, the atmospheric transmittance, spectral sensitivity of the CCD, and background sky light also influence the signal produced in the CCD. The CCD output can be expressed by the following equation:

$$S_{\text{CCD}} = \int_{\lambda = 350\text{nm}}^{\lambda = 1100\text{nm}} F(\lambda)A(\lambda)T(\lambda)Q(\lambda)gtd\lambda. \quad (\text{Equ. 20.3})$$

In this equation,  $S_{\text{CCD}}$  is the output of the CCD, and:

- $\lambda$  is the wavelength of the light;
- $F(\lambda)$  is the flux arriving from the celestial source as a function of wavelength;
- $A(\lambda)$  is the transmittance of the atmosphere as a function of wavelength;
- $T(\lambda)$  is the transmittance of the color filter(s) used as a function of wavelength;
- $Q(\lambda)$  represents the combined transmission of the optics and quantum efficiency of the detector as a function of wavelength;
- $g$  is the gain of the amplifier, in electrons per ADU in the output image; and
- $t$  is the integration time used to capture the image.

To explore color imaging, we need to simplify this equation. We can accomplish this by lumping together factors that remain the same and integrating over each spectral band to obtain a single figure that represents the integrated characteristics.

In this simplified picture, we treat light from celestial objects as consisting of only five passbands: infrared, red, green, blue, and ultraviolet. The actual passbands do not matter except insofar as they excite the color sensors in the human retina. Infrared and ultraviolet fall outside the range of human vision, but are detectable by CCDs. These passbands correspond to the following ranges:

- Infrared consists of wavelengths longer than 700 nm;
- Red with wavelengths from 700 nm to 600 nm;

## Chapter 20: Building Color Images

**Table 20.3 Atmospheric Transmittance\***

Altitude	Red (%)	Green (%)	Blue (%)
90° (zenith)	100.0	100.0	100.0
85°	100.0	99.9	99.9
80°	99.8	99.7	99.6
75°	99.6	99.4	99.1
70°	99.3	98.9	98.4
65°	98.9	98.3	97.5
60°	98.3	97.5	96.2
55°	97.6	96.4	94.7
50°	96.7	95.1	92.7
45°	95.5	93.4	90.2
40°	94.0	91.2	87.1
35°	92.1	88.4	83.1
30°	89.5	84.7	78.0
25°	86.0	79.7	71.2
20°	80.8	72.7	62.0
15°	72.9	62.2	49.1
10°	59.1	45.4	30.6
5°	31.4	17.6	7.4

\* Transmittance typical of a clear night at a moist, low-altitude observing site.

- Green with wavelengths between 600 nm and 500 nm;
- Blue with wavelengths between 500 nm and 400 nm;
- Ultraviolet consists of all wavelengths shorter than 400 nm.

As we discuss color imaging, we will refer to these spectral regions as the *I*, *R*, *G*, *B*, and *U* passbands. With the spectrum partitioned into five passbands, a camera produces a signal that is the sum of the response in each of them:

$$S = F_I A_I Q_I + F_R A_R Q_R + F_G A_G Q_G + F_B A_B Q_B + F_U A_U Q_U \quad (\text{Equ. 20.4})$$

where *S* is the output signal from the camera,  $F_I$ ,  $F_R$ ,  $F_G$ ,  $F_B$ , and  $F_U$  are light flux from the object in the five passbands;  $A_I$ ,  $A_R$ ,  $A_G$ ,  $A_B$ , and  $A_U$  are the transmittance of the atmosphere in each passband; and  $Q_I$ ,  $Q_R$ ,  $Q_G$ ,  $Q_B$ , and  $Q_U$  are the product of the transmittance of the telescope optics and the quantum efficiency of the CCD in each.

Consider next the signal that results when you add a filter:

$$S = F_I T_I A_I Q_I + F_R T_R A_R Q_R + F_G T_G A_G Q_G + F_B T_B A_B Q_B + F_U T_U A_U Q_U \quad (\text{Equ. 20.5})$$

where  $T_I$ ,  $T_R$ ,  $T_G$ ,  $T_B$ , and  $T_U$  are the filter transmittances within the subscripted passband. The signal in each passband is the product of the flux from the object,

### Section 20.3: Red/Green/Blue Tri-Color Imaging

the transmittance of the filter, the transmittance of the atmosphere, and the quantum efficiency of the detector. The signal from the detector is the sum of the signals in each passband.

Now consider making images through a set of red, green and blue filters. An ideal red filter transmits all of the light only in the red passband, and blocks light in all of the other passbands:

$$T_I = 0, T_R = 1, T_G = 0, T_B = 0, \text{ and } T_U = 0.$$

The ideal green filter transmits only in the green passband:

$$T_I = 0, T_R = 0, T_G = 1, T_B = 0, \text{ and } T_U = 0.$$

And an ideal blue filter transmits only in the blue passband:

$$T_I = 0, T_R = 0, T_G = 0, T_B = 1, \text{ and } T_U = 0.$$

Most filters fall short of the ideal, and transmit some light outside the proper passbands. Especially insidious are infrared leaks, because many CCDs reach their peak quantum efficiency in the near infrared, so that even a small leak can generate a large signal. Although filter leakage increases the signal, the signal does not result from light in the correct passband, so the inflated signal leads to inaccurate color. It is important to remember that the more closely the filter set approaches the ideal, the better the color images they will produce.

However, suppose that we do have filters that are reasonably close to the ideal. Upon substituting their transmittances into Equation 20.5, we obtain the following signals from a celestial object:

$$S_R = F_R A_R T_R Q_R \quad (\text{Equ. 20.6})$$

$$S_G = F_G A_G T_G Q_G \quad (\text{Equ. 20.7})$$

$$S_B = F_B A_B T_B Q_B. \quad (\text{Equ. 20.8})$$

**Skylight Contamination.** We have focused on the flux coming from the astronomical source, but in so doing we have neglected an important contribution to the signal level—the light of the night sky background. Even at dark-sky sites, the sky background signal is often as large as or larger than that of huge nebulae, the outer arms of spiral galaxies, and faint clusters of galaxies. Under suburban skies, sky background represents the dominant source of signal in the filtered images. Thus, the actual signal level seen in an astronomical image is:

$$S_R = F_R A_R T_R Q_R + S_{R_{\text{SKY}}} \quad (\text{Equ. 20.9})$$

$$S_G = F_G A_G T_G Q_G + S_{G_{\text{SKY}}} \quad (\text{Equ. 20.10})$$

$$S_B = F_B A_B T_B Q_B + S_{B_{\text{SKY}}} \quad (\text{Equ. 20.11})$$

where  $S_{R_{\text{SKY}}}$ ,  $S_{G_{\text{SKY}}}$ , and  $S_{B_{\text{SKY}}}$  are signals contributed by sky background light.

## Chapter 20: Building Color Images

**Table 20.4 Bright\* Sun-Like Stars for White Balance**

Name	Catalog	$\alpha_{2000}$	$\delta_{2000}$	Sp. Type	$m_v$	Const.
	BS 9107	00 04 53.6	+34 39 56	G2V	6.11	And
	HD 1461	00 18 41.7	-08 03 04	G3	6.47	Cet
<b>9 Cet</b>	HD 1835	00 22 51.7	-12 12 34	G2.5	6.39	Cet
<b>18 Cet</b>	BS 0203	00 45 28.6	-12 52 51	G2V	6.16	Cet
	HD 4915	00 51 10.7	-05 02 23	G0V	6.98	Cet
	HD 8262	01 22 17.7	+18 40 57	G2V	6.93	Psc
	BS 483	01 41 47.1	+42 36 49	G1.5V	4.97	And
$\zeta^1$ Ret	BS 1006	03 17 46.2	-62 34 32	G2.5V	5.51	Ret
$\zeta^2$ Ret	BS 1010	03 18 12.9	-62 30 23	G1.5V	5.23	Ret
	HD 20619	03 19 01.8	-02 50 36	G1.5	7.05	Eri
$\lambda$ Aur	BS 1729	05 19 08.4	+40 05 57	G2IV/V	4.71	Aur
	HD 44594	06 20 06.1	-48 44 28	G2	6.61	Car
	HD 45184	06 24 43.8	-28 46 48	G2	6.37	Col
	HD 53705	07 03 57.2	-43 36 29	G1.5	5.56	Pup
	HD 76151	08 54 17.9	-05 26 04	G2	6.01	Hya
<b>20 LMi</b>	BS 3951	10 01 00.6	+31 55 25	G3	5.37	LMi
<b>35 Leo</b>	HD 89010	10 16 32.2	+23 30 31	G1.5V	5.97	Leo
<b>47 UMa</b>	BS 4277	10 59 27.9	+40 25 49	G0V	5.04	UMa
	HD 96700	11 07 54.3	-30 10 22	G1	6.52	Hya

\* Stars brighter than  $m_v < 7.1$  with color closely matching the Sun.

This contamination source is a significant problem. Sky background light can be strongly colored thus distorting the colors of the celestial objects you are trying to image. Furthermore, light-polluted skies are usually brighter near the horizon, so the sky might change brightness across the image. This is intensity gradient, and in color imaging, intensity gradients cause color gradients.

### 20.3.2 Step 2: Correct the Filtered Images

In the previous section, we saw that the color images,  $S_R$ ,  $S_G$ , and  $S_B$ , are contaminated by an additive signal, the sky background light, and scaled by three multiplicative factors: filter transmittances, detector quantum efficiency, and atmospheric attenuation. To achieve accurate color, we need to subtract the additive factor and to divide out the multiplicative factors.

#### 20.3.2.1 Subtract the Skylight Background

Subtracting sky background appears simple, but it is often complex. There are several difficulties, including:

### Section 20.3: Red/Green/Blue Tri-Color Imaging

**Table 20.4 Bright\* Sun-Like Stars for White Balance**

Name	Catalog	$\alpha_{2000}$	$\delta_{2000}$	Sp. Type	$m_v$	Const.
	HD 102365	11 46 31.0	-40 30 01	G3	4.89	Cen
	BS 5384	14 23 15.2	+01 14 30	G1V	6.27	Vir
	BS 5596	14 50 20.2	+82 30 43	F9V	5.64	UMi
$\psi$ Ser	BS 5853	15 44 01.6	+02 30 54	G2.5	5.87	Ser
$\lambda$ Ser	BS 5868	15 46 26.5	+07 21 11	G0V	4.42	Ser
$\beta$ Ser	BS 5911	15 53 12.0	+13 11 48	G1	6.08	Ser
	HD 144585	16 07 03.2	+14 04 16	G2	6.31	Ser
$\tau$ Sco	BS 6060	16 15 37.1	-08 22 11	G2Va	5.50	Sco
	HD 152792	16 53 32.2	+42 49 30	G0V	6.83	Her
	BS 6538	17 32 00.9	+34 16 15	G5V	6.56	Her
	HD 168874	18 20 49.1	+27 31 50	G2IV	7.01	Her
	HD 177082	19 02 38.0	+14 34 02	G2V	6.90	Aql
$\alpha$ Cyg A	BS 7503	19 41 48.8	+50 31 31	G1.5V	5.99	Cyg
$\alpha$ Cyg B	BS 7504	19 41 51.8	+50 31 03	G2.5V	6.24	Cyg
	HD 187237	19 48 00.7	+27 52 10	G2III	6.90	Vul
	BS 7569	19 52 03.4	+11 37 44	G0V	6.16	Aql
	BS 7683	20 05 09.7	+38 28 42	G5IV	6.19	Cyg
	BS 7914	20 40 45.1	+19 56 07	G5V	6.44	Del
	BS 8964	23 37 58.5	+46 11 59	G5	6.60	And

\* Stars brighter than  $m_v < 7.1$  with color closely matching the Sun.

- Sky background may not be uniform across the image, so it becomes necessary to determine the pattern (gradient) present in it,
- celestial objects may fill a large part of all of the image, making it difficult to find a spot representing the true sky background, and
- Poisson noise and readout noise mean that there is no single pixel value for the sky background, but instead a range of pixel values due to these noise sources.

When the subject of the image and the sky are clearly separate (that is, the subject is not a frame-filling nebula), it is reasonable to use median of the pixel values in star-free regions of the image as the sky background level. After subtraction of the median sky background from the image, however, the sky will have negative as well as positive pixel values. Later color processing stages must be able to deal with these physically unrealistic sky brightness values.

When the subject of the image fills the frame, even if there are no gradients present, determination of the sky background is difficult. It is necessary either to find one or more places where the subject appears to be absent and to measure the sky background in those places, to make a sky background image of an adjacent

## Chapter 20: Building Color Images

**Table 20.5 Faint\* Sun-Like Stars for White Balance**

Name	$\alpha_{2000}$	$\delta_{2000}$	$m_V$	Sp. Type	Const.
<b>SA 140-84</b>	00 03 38	-28 41 46	11.961	G?	Scl
<b>SA 92-276</b>	00 56 27	+00 41 52	12.036	G5	Cet
<b>SA 93-101</b>	01 53 18	+00 22 25	9.734	G5	Cet
<b>vB64</b>	04 26 40	+16 44 49	8.10	G2	Tau
<b>SA 92-249</b>	05 57 07	+00 01 11	11.733	G5	Ori
<b>SA 98-682</b>	06 52 16	-00 19 42	13.749	G?	Mon
<b>Rubin 149B</b>	07 24 18	-00 33 07	12.642	G?	CMi
<b>SA 101-321</b>	09 55 40	-00 18 52	12.85	G7	Sex
<b>SA 101-329</b>	09 56 19	-00 26 28	11.99	G7	Sex
<b>SA 102-370</b>	10 56 34	-01 10 40	11.229	G2	Leo
<b>SA 102-1081</b>	10 57 04	-00 13 12	9.903	G5	Leo
<b>SA 103-487</b>	11 55 11	-00 23 38	11.874	G5	Vir
<b>SA 103-204</b>	11 57 27	-00 56 53	11.189	G7	Vir
<b>SA 104-483</b>	12 44 17	-00 27 33	12.08	G5	Vir
<b>SA 105-56</b>	13 38 42	-01 14 14	9.975	G5	Vir
<b>SA 107-684</b>	15 37 18	-00 09 50	8.433	G3	Ser
<b>SA 107-998</b>	15 38 16	+00 15 23	10.436	G3	Ser
<b>SA 196-1801</b>	17 11 08	-60 06 29	12.755	G?	Ara
<b>SA 110-361</b>	18 42 45	+00 08 04	12.425	G5	Aql
<b>SA 112-1333</b>	20 43 12	+00 26 15	9.977	G2	Aqr
<b>SA 133-276</b>	21 42 27	+00 26 20	9.074	G5	Aqr
<b>SA 114-654</b>	22 41 26	+01 10 11	11.83	G0	Aqr
<b>HD 219018</b>	23 12 39	+02 41 10	7.708	G1	Psc
<b>SA 115-2688</b>	23 42 31	+00 52 11	12.487	G?	Psc
<b>SA 115-271</b>	23 45 42	+00 45 14	9.695	G2	Psc

\* Stars fainter than  $m_V \approx 8.0$  for white balance with large-aperture telescopes.

nebula-free field or to make an educated guess based on the observer's experience with similar images.

When a sky gradient is present, the values of  $S_{R_{SKY}}$ ,  $S_{G_{SKY}}$ , and  $S_{B_{SKY}}$  vary from place to place in the image. If the sky and the subject are clearly separated, the observer can apply a gradient correction (a function that is equal in magnitude but opposite in sign from the sky background) to the image.

- **Tip:** Use **AIP4Win**'s gradient correction tools to correct sky brightness gradients in your images. These tools produce satisfactory "flat" sky from a variety of different types of sky background gradient.

As a practical matter, it is okay to defer subtracting a uniform sky background until the next step, when the corrected images are converted into display values. However, image gradients must be "flattened" before attempting to make a color image.

### 20.3.2.2 Color Balance with G2V Stars

To make accurate color images, the signal,  $S$ , must be proportional only to the flux from celestial object,  $F$ , in each passband. However, an earthbound observer can only capture signals that have been multiplied by the extraneous filter and atmospheric factors.

A reliable way to accomplish color balance is to obtain filter images of a standard object of known color. In television, photography, and digital imaging, this is accomplished by adjusting the red, green, and blue signals so that gray and white objects have the same red, green, and blue pixel value—a process known as “white balancing.” Standard display devices—computer monitors and printers—are designed so that when a display receives equal color channel values, it generates gray or white on the screen or on the printed page.

In astronomical imaging, we can use main-sequence stars like our Sun, members of spectral class G2V, as white standards. Just as noontime sunlight is the accepted terrestrial standard of white light, class G2V stars comprise an astronomical standard for whiteness. Tables 20.4 and 20.5 list stars that can serve as standards for white balance.

To complete image capture, obtain images of a white standard through the same filters used for the filtered images of the astronomical object. The signal levels from a G2V white standard are:

$$S_{R_{\text{G2V}}} = F_{R_{\text{G2V}}} A_R T_R Q_R, \quad (\text{Equ. 20.12})$$

$$S_{G_{\text{G2V}}} = F_{G_{\text{G2V}}} A_G T_G Q_G, \quad (\text{Equ. 20.13})$$

$$S_{B_{\text{G2V}}} = F_{B_{\text{G2V}}} A_B T_B Q_B. \quad (\text{Equ. 20.14})$$

You can see that the “balanced” stellar flux is altered by the same multiplicative factors that compromise a set of color images: the atmospheric transmittances,  $A_x$ , the filter transmittances,  $T_x$ , and the detector’s quantum efficiency,  $Q_x$ , to yield the three signal levels:  $S_{R_{\text{G2V}}}$ ,  $S_{G_{\text{G2V}}}$ , and  $S_{B_{\text{G2V}}}$ . Note that standard techniques used in stellar photometry automatically subtract the sky background from star image signal.

The filter transmittances and detector quantum efficiencies do not change from image to image and night to night. However, because atmospheric transmittance depends on the angular distance of the object from the zenith at the time the images are made, in theory each image requires a different atmospheric correction. In practice, however, we can model the atmospheric transmission and compensate knowing the altitude at the time when the color images were made.

**Extinction Dims G2V Stars.** On several nights with skies that are typical of your observing site, you make images of standard G2V stars using the same telescope, filters, and CCD camera that you normally do, noting as you do, the zenith distance ( $90^\circ - \text{elevation angle}$ ) when you make the images. Because the G2V stars are bright, the exposure times can be quite short, and you can make these im-

## Chapter 20: Building Color Images

ages on a Moonlight night.

Next, measure the  $S_{R_{G2V}}$ ,  $S_{G_{G2V}}$ , and  $S_{B_{G2V}}$  signals produced by each G2V star. Because these measurements are made through the atmosphere, blue light will be more strongly attenuated than green, and green more strongly than red. Thus a “white” standard appears yellow or reddish relative to the color it would appear if it had been directly overhead in the sky.

The atmospheric transmittance in each passband is:

$$A_R = 10^{-0.4k_R(\sec Z_{G2V} - 1)}, \quad (\text{Equ. 20.15})$$

$$A_G = 10^{-0.4k_G(\sec Z_{G2V} - 1)}, \quad (\text{Equ. 20.16})$$

$$A_B = 10^{-0.4k_B(\sec Z_{G2V} - 1)}, \quad (\text{Equ. 20.17})$$

where  $Z_{G2V}$  is the angular distance between the standard star and the zenith, and  $k_R$ ,  $k_G$ , and  $k_B$ , are the extinction coefficients for each passband. In this formulation the atmospheric transmittance is 100% at the zenith.

Although it is possible to determine extinction coefficients for these passbands, it is entirely adequate to use typical values for the extinction coefficients and to read the atmospheric transmittances from a table.

Typical extinction values for moist, low-altitude observing sites are:

$$k_R = 0.13, k_G = 0.20, \text{ and } k_B = 0.29.$$

Typical extinction values for dry, high-altitude observing sites are:

$$k_R = 0.07, k_G = 0.12, \text{ and } k_B = 0.22.$$

To recover the signal levels that a G2V star would have had if you had observed it at the zenith, divide each observed signal by the corresponding atmospheric transmittance:

$$S'_{R_{G2V}} = (F_{R_{G2V}} A_R T_R Q_R) / A_R, \quad (\text{Equ. 20.18})$$

$$S'_{G_{G2V}} = (F_{G_{G2V}} A_G T_G Q_G) / A_G, \quad (\text{Equ. 20.19})$$

$$S'_{B_{G2V}} = (F_{B_{G2V}} A_B T_B Q_B) / A_B. \quad (\text{Equ. 20.20})$$

The atmospheric extinction terms cancel, leaving a signal that is the product of the G2V flux, the filter transmittances, and the sensor’s quantum efficiencies. Since the relative fluxes for a G2V star are, by definition, identical to white light, the flux terms also drop out:

$$S'_{R_{G2V}} = T_R Q_R, \quad (\text{Equ. 20.21})$$

$$S'_{G_{G2V}} = T_G Q_G, \quad (\text{Equ. 20.22})$$

$$S'_{B_{G2V}} = T_B Q_B. \quad (\text{Equ. 20.23})$$

### Section 20.3: Red/Green/Blue Tri-Color Imaging

The final step is to find weighting factors,  $W$ , that will correct all three standard star images to the same value. Designating the highest of the signals as  $S_{\max}$ , divide each filter signal level by this value:

$$W_R = S'_{R_{G2V}} / S_{\max}, \quad (\text{Equ. 20.24})$$

$$W_G = S'_{G_{G2V}} / S_{\max}, \quad (\text{Equ. 20.25})$$

$$W_B = S'_{B_{G2V}} / S_{\max}. \quad (\text{Equ. 20.26})$$

These weight factors correct for filter transmittance and detector quantum efficiency. With most filters and CCDs, the weight factors will be something like 1.00:0.93:0.78, that is, reasonably close to ideal values of 1.00:1.00:1.00.

- **Tip:** *AIP4Win's Color Calculator Tool makes it easy to use G2V stars for white balance. You can load these images, measure the stars and enter the altitudes at the time you made the images, and the tool will compensate for atmospheric extinction and compute suitably adjusted weighting factors for you.*

**Correct Images for Extinction.** Now the filtered images can be white-balanced and corrected for atmospheric extinction. The images of the celestial object were attenuated in the same way that the images of the white standard were:

$$A_R = 10^{-0.4k_R(\sec Z - 1)}, \quad (\text{Equ. 20.27})$$

$$A_G = 10^{-0.4k_G(\sec Z - 1)}, \quad (\text{Equ. 20.28})$$

$$A_B = 10^{-0.4k_B(\sec Z - 1)}, \quad (\text{Equ. 20.29})$$

where  $Z$  is the object's zenith distance when the image was made, and  $A_R$ ,  $A_G$ , and  $A_B$  are the atmospheric transmittance in the  $R$ ,  $G$ , and  $B$  passbands.

Finally, we are ready to apply atmosphere and white-balance corrections to each of the filtered images to obtain corrected signals:

$$S'_R = \frac{S_R}{A_R W_R}, \quad (\text{Equ. 20.30})$$

$$S'_G = \frac{S_G}{A_G W_G}, \quad (\text{Equ. 20.31})$$

$$S'_B = \frac{S_B}{A_B W_B}. \quad (\text{Equ. 20.32})$$

If there is a residual sky background component in the images, then applying these corrections will alter the sky component of the signal, but this does not matter because you can subtract the sky background as part of the process of generating a color image.

## Chapter 20: Building Color Images

- **Tip:** Given a set of weighting factors, **AIP4Win’s Process RGB Tool** will adjust a set of filtered images to the correct color balance. After they have been adjusted, the images are said to be “G2V color balanced.”

### 20.3.3 Create the Color Image

In displaying the image, the goal is to convert corrected images into display inputs. Displays usually accept an 8-bit input for the red, green, and blue primaries, so each color can be displayed in 256 levels of intensity. The three color-channel signals going to the display device, ( $D_R$ ,  $D_G$ ,  $D_B$ ), are called an *RGB triad*.

The output floor is the smallest display value, (0, 0, 0), which appears black; the output ceiling is the greatest display value, (255, 255, 255), which is white. An input of (100, 100, 100) appears as a medium shade of gray, and an input of (192, 128, 64) displays as a light yellowish-orange. An 8-bit display is capable of producing  $256^3 = 16,777,216$  colors—more than the human visual system can distinguish as separate colors.

Given that a color image contains an average sky background that should appear as black (0, 0, 0) or a very dark shade of gray, such as (5, 5, 5), and highlights that should display as bright white (255, 255, 255), the task of the display stage is to convert the pixel values in the corrected images to display values.

Creating the display begins with determining two quantities in each image:

- $S'_{R_{SKY}}$ ,  $S'_{G_{SKY}}$ , and  $S'_{B_{SKY}}$ , the pixel value of any residual sky background. This should be measured in a star-free region well away from any celestial objects in the image.
- $S'_{R_{MAX}}$ ,  $S'_{G_{MAX}}$ , and  $S'_{B_{MAX}}$ , a pixel value representing the brightest significant region or detail in the celestial object. This choice is subjective—it might be the pixel value of the brightest stars in the images, of the core of a globular cluster, of a galactic nucleus, or the brightest knots in an emission nebula.

The next step is to determine which image has the greatest range,  $R_{RGB}$ , between the maximum value and the sky brightness:

$$R_{RGB} = \max \left\{ \begin{array}{l} W_R(S'_{R_{MAX}} - S'_{R_{SKY}}) \\ W_G(S'_{G_{MAX}} - S'_{G_{SKY}}) \\ W_B(S'_{B_{MAX}} - S'_{B_{SKY}}) \end{array} \right. . \quad (\text{Equ. 20.33})$$

The conversion to 8-bit display values then proceeds as follows:

$$D_R = 255 \left( \frac{S'_R - S'_{R_{SKY}}}{R_{RGB}} \right)^{1/\gamma}, \quad (\text{Equ. 20.34})$$

## Section 20.3: Red/Green/Blue Tri-Color Imaging

$$D_G = 255 \left( \frac{S'_G - S'_{G_{SKY}}}{R_{RGB}} \right)^{1/\gamma}, \quad (\text{Equ. 20.35})$$

$$D_B = 255 \left( \frac{S'_B - S'_{B_{SKY}}}{R_{RGB}} \right)^{1/\gamma}, \quad (\text{Equ. 20.36})$$

where  $\gamma$  (gamma) is a constant that corrects the nonlinear relationship between the pixel value sent to the display device and its light output. For computer monitors, the value of  $\gamma$  is usually close to 1.8. In some computers, the graphics display card automatically compensates for monitor gamma.

The image displayed on the screen will have a black sky background; stars will show soft colors corresponding to their spectral types, and sun-like stars will appear white. Given a good set of filters, accurate measurement of G2V stars, and a good display, nebulae and galaxies appear in the precise colors they would have if you had observed them at the zenith. As a general rule, astronomical images made as described above show rather delicate colors and have an extraordinarily “natural” appearance.

### 20.3.4 Summary: White-Balance Using G2V Stars

It is possible to construct color images by making images through red, green, and blue color filters. The images can be corrected to accurate white balance by shooting auxiliary images of standard sun-like class G2V stars through the same filters.

1. In preparation, make short-exposure images of one or more G2V stars through the same color filters you plan to use for color imaging. Use the same exposure time for the filtered images, or always use the same exposure ratios among the filter exposures. Note the elevation angle of the G2V star(s) when you make their images.
2. Measure the total pixel value in the filtered star images, and compute the weighting factors,  $W_R$ ,  $W_G$ , and  $W_B$ , for the filter set you are using. **AIP4Win**'s Color Calculator Tool makes this easy. These weight factors will remain valid until you change the filters, CCD camera, or telescope.
3. Take filtered image sets. The exposure times for the filters should either be the same or have the same ratio as the G2V exposures. Note the elevation angle of the object when you take the images. Take appropriate dark frames and flat fields. Calibrate and stack to produce a high-quality image of the subject through each filter.
4. If a sky background gradient is present in the filtered images, correct the gradient. One of **AIP4Win**'s Gradient Tools can do this for you.
5. Register the three filtered images to sub-pixel accuracy, and then correct each of the filtered images using its sky background value, elevation angle, and weight factors. **AIP4Win**'s Process RGB Tool carries out these operations.

## Chapter 20: Building Color Images

6. Create a color image from the three corrected color channels. You can use **AIP4Win**'s RGB->Color function or the Join Colors Tool to meld the color channels into a single color image.

### 20.4 Using Field Stars for White Balance

Color correction using G2V standard stars is sometimes either not practical or not possible, but fortunately it is usually possible to take a shortcut to good-looking color images. This method assumes that the average color of the field stars in the image is either white or close to white. This method relies on the background population of stars in the image to serve as the color reference standard.

In the shortcut approach to RGB imaging, the observer can make a set of red, green, and blue images *without* making images of a white standard, and *without* correcting the images prior to displaying them.

Image capture is exactly the same as the G2V method. The observer takes a filtered set of images, dark frames, and flat frames, and produces a set of calibrated filtered images,  $S_R$ ,  $S_G$ , and  $S_B$ . However, except for correcting any sky background gradients, this method skips over the color correction step.

Creating the color image begins by measuring two quantities in each image:

- $S_{R_{SKY}}$ ,  $S_{G_{SKY}}$ , and  $S_{B_{SKY}}$ , the pixel value of the sky background, measured well away from any celestial objects in the image in an area free of stars.
- $S_{R_{MAX}}$ ,  $S_{G_{MAX}}$ ,  $S_{B_{MAX}}$ , a pixel value defined by histogram endpoint. Effective endpoints usually have a value between 0.995 and 0.9995 of the pixels in the image; that is, bright stars represent between 0.5% and 0.05% of the total pixel count in each of the color filtered images.

The conversion to 8-bit display values then proceeds as follows:

$$D_R = 255 \left( \frac{S_R - S_{R_{SKY}}}{S_{R_{MAX}} - S_{R_{SKY}}} \right)^{1/\gamma} \quad (\text{Equ. 20.37})$$

$$D_G = 255 \left( \frac{S_G - S_{G_{SKY}}}{S_{G_{MAX}} - S_{G_{SKY}}} \right)^{1/\gamma} \quad (\text{Equ. 20.38})$$

$$D_B = 255 \left( \frac{S_B - S_{B_{SKY}}}{S_{B_{MAX}} - S_{B_{SKY}}} \right)^{1/\gamma} \quad (\text{Equ. 20.39})$$

where  $\gamma$  is a constant—usually near 1.8—chosen to compensate for the nonlinear light output from the monitor.

The field-star method of white balance depends on the validity of the assumption that the average color of the brightest stars in the background is white.

## Section 20.5: Color Images from Filter-Matrix Cameras

Experience shows that the shortcut method works fairly well 95% of the time. Forcing the same number of bright-star pixels to become saturated usually produces images with a fair approximation of the proper color balance.

The field-star method breaks down for images containing young star clusters (because these contain large numbers of hot young blue stars), and for strongly colored objects such as bright emission nebulae, where the coloration affects the histogram endpoint pixel value.

### 20.4.1 Summary: White-Balance Using Field Stars

It is possible to construct “reasonably accurate” color images based on the normally justified *assumption* that the average color of the field stars in the image is white or nearly white. If this assumption is false, the color balance will not be accurate.

1. Take filtered image sets. You may use any exposure times for the filtered images. Take appropriate dark frames and flat fields. Calibrate and stack to produce a high-quality image of the subject through each filter.
2. If a sky background gradient is present in the filtered images, correct the gradient. One of **AIP4Win**’s Gradient Tools can do this for you.
3. Register the three filtered images to sub-pixel accuracy. **AIP4Win**’s Process RGB Tool can carry out this operation.
4. Create a color image from the three corrected color channels. You can use **AIP4Win**’s Join Colors Tool, with an automatic color balance endpoint that yields pleasing color results, to join the color channels into a single color image.

Color balancing with field stars is considerably easier than color balancing with solar-class G2V stars. However, although images color-balanced with field stars could *in theory* have color as accurate as images balanced with G2V stars, there is no guarantee that the “average color is white” assumption will be true for a particular image.

## 20.5 Color Images from Filter-Matrix Cameras

Filter-matrix cameras collect three channels of color information in a single integration. This technology depends on a sensor that has integral red, green, and blue color filters (or sometimes cyan, magenta, yellow, and green color filters) organized in a  $2 \times 2$  unit cell. Instead of capturing the color channels in sequence, the channels are captured simultaneously—at considerable cost to the versatility, sensitivity, and resolution of the sensor—but with greater convenience and ease of use.

If a filter-matrix sensor is read out directly, as a monochrome array, the resulting “raw” image looks like a checkerboard, with different squares showing different color channels. To recover the embedded color image, it is necessary to interpolate a representative ( $R$ ,  $G$ ,  $B$ ) value for each pixel from surrounding pixels. Since the interpolation is equivalent to a low-pass spatial filter, the resolution of

## Chapter 20: Building Color Images

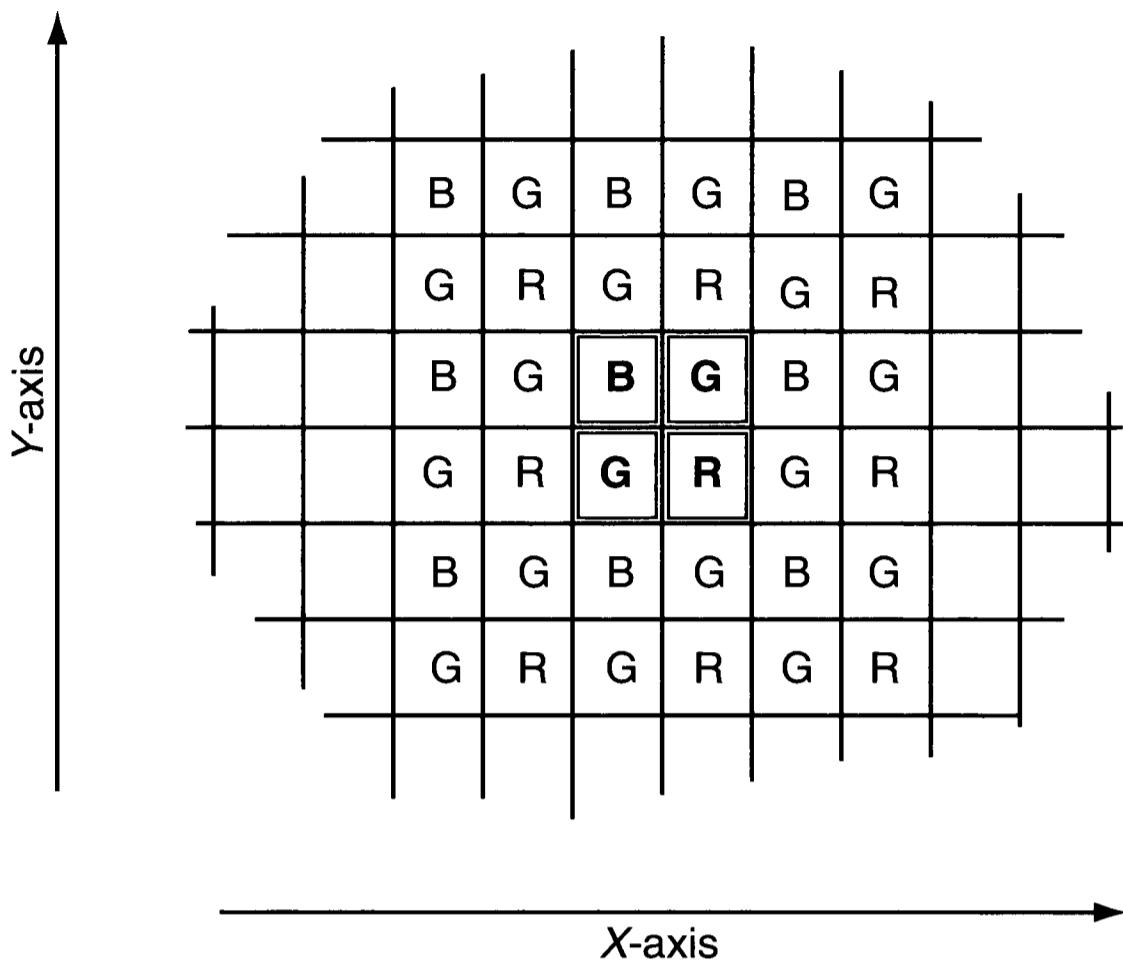


Figure 20.5 The photosites in a filter-matrix camera are masked with color filters in a regular pattern. Finding the  $(R, G, B)$  triad for each pixel requires interpolating the two missing color channels from its neighbors, trading some loss of sharpness for the simplicity of one-shot color imaging.

the array is reduced by a factor of approximately 1.6.

To see how we obtain  $S_R$ ,  $S_G$ , and  $S_B$  values for every pixel, consider the pixels in an RGB filter matrix (see Figure 20.5). Each  $2 \times 2$  unit cell contains one blue-, one red-, and two green-filtered pixels, and each of these is the center of a neighborhood comprised of eight pixels. Surrounding each blue pixel are four red pixels and four green pixels. Thus a complete  $(R, G, B)$  triad for a blue pixel at location  $(x, y)$  is:

$$S_R(x, y) = \frac{1}{4} \{ S(x + 1, y + 1) + S(x + 1, y - 1) \\ + S(x - 1, y + 1) + S(x - 1, y - 1) \} \quad (\text{Equ. 20.40})$$

$$S_G(x, y) = \frac{1}{4} \{ S(x, y + 1) + S(x, y - 1) \\ + S(x + 1, y) + S(x - 1, y) \} \quad (\text{Equ. 20.41})$$

$$S_B(x, y) = S(x, y) . \quad (\text{Equ. 20.42})$$

Similar sets of equations apply for red and green pixels. In this way, a raw filter-matrix image is converted to yield a complete set of red-, green-, and blue-filtered images,  $S_R$ ,  $S_G$ , and  $S_B$ .

Once decoded, the color channels from a filter-matrix image can be pro-

## Section 20.6: Color Space: Geometric Interpretations of Color

cessed the same way other red/green/blue-filtered image sets are processed and color balanced—providing, of course, that the camera and software that take the image do not perform non-linear operations on the raw image data.

## 20.6 Color Space: Geometric Interpretations of Color

To make color images, we capture information as an  $(x, y)$  array of  $(R, G, B)$  triads. Color images are two-dimensional displays of color points. We can, however, plot the same information as a three-dimensional display of  $(x, y)$  pixels. Such a three-dimensional plot is called a *color space*. In this section, we define three different color spaces and explore how different color spaces can be used to create and process color images.

### 20.6.1 RGB Color Space

RGB color space is a plot of pixels by their  $(R, G, B)$  value. Thus far we have considered two different RGB color spaces: the wide-open RGB color space containing image data direct from the CCD camera, and the sharply bounded RGB color space used to display color images. Converting raw image data into displayable images, the tasks described in the preceding sections, is essentially that of squeezing as much data as possible from their original wide-open RGB color space into the constrained display color space.

Normal output from a 16-bit CCD camera will have pixel values running from 0 to 65,535, and in stacked images, red, green and blue pixel values can run into the millions. The highest pixel values appear in saturated star images; in most CCD cameras, saturated pixels have values at or near the highest 16-bit integer, namely 65,535.

Consider an image showing a completely blank piece of sky, so that every pixel receives the same exposure. However, instead of a graph in which all of the pixels are clustered at  $S_{R_{\text{SKY}}}$ ,  $S_{G_{\text{SKY}}}$ , and  $S_{B_{\text{SKY}}}$ , the statistical variation in the photon count (Poisson noise) spreads sky points over the ranges:

$$S_{R_{\text{sky}}} \pm \sqrt{\frac{S_{R_{\text{sky}}}}{g}} \text{ [ADU]} \quad (\text{Equ. 20.43})$$

$$S_{G_{\text{sky}}} \pm \sqrt{\frac{S_{G_{\text{sky}}}}{g}} \text{ [ADU]} \quad (\text{Equ. 20.44})$$

$$S_{B_{\text{sky}}} \pm \sqrt{\frac{S_{B_{\text{sky}}}}{g}} \text{ [ADU]}, \quad (\text{Equ. 20.45})$$

where  $g$  is the CCD camera's gain in photons per ADU. Because of Poisson noise, a perfectly blank section of night sky will appear as an extended cluster of points centered on the coordinates  $S_{R_{\text{SKY}}}$ ,  $S_{G_{\text{SKY}}}$ , and  $S_{B_{\text{SKY}}}$ .

Figure 20.6 shows a plot of  $(R, G, B)$  color space for an image set containing

## Chapter 20: Building Color Images

### Astronomical Red/Green/Blue Color Space

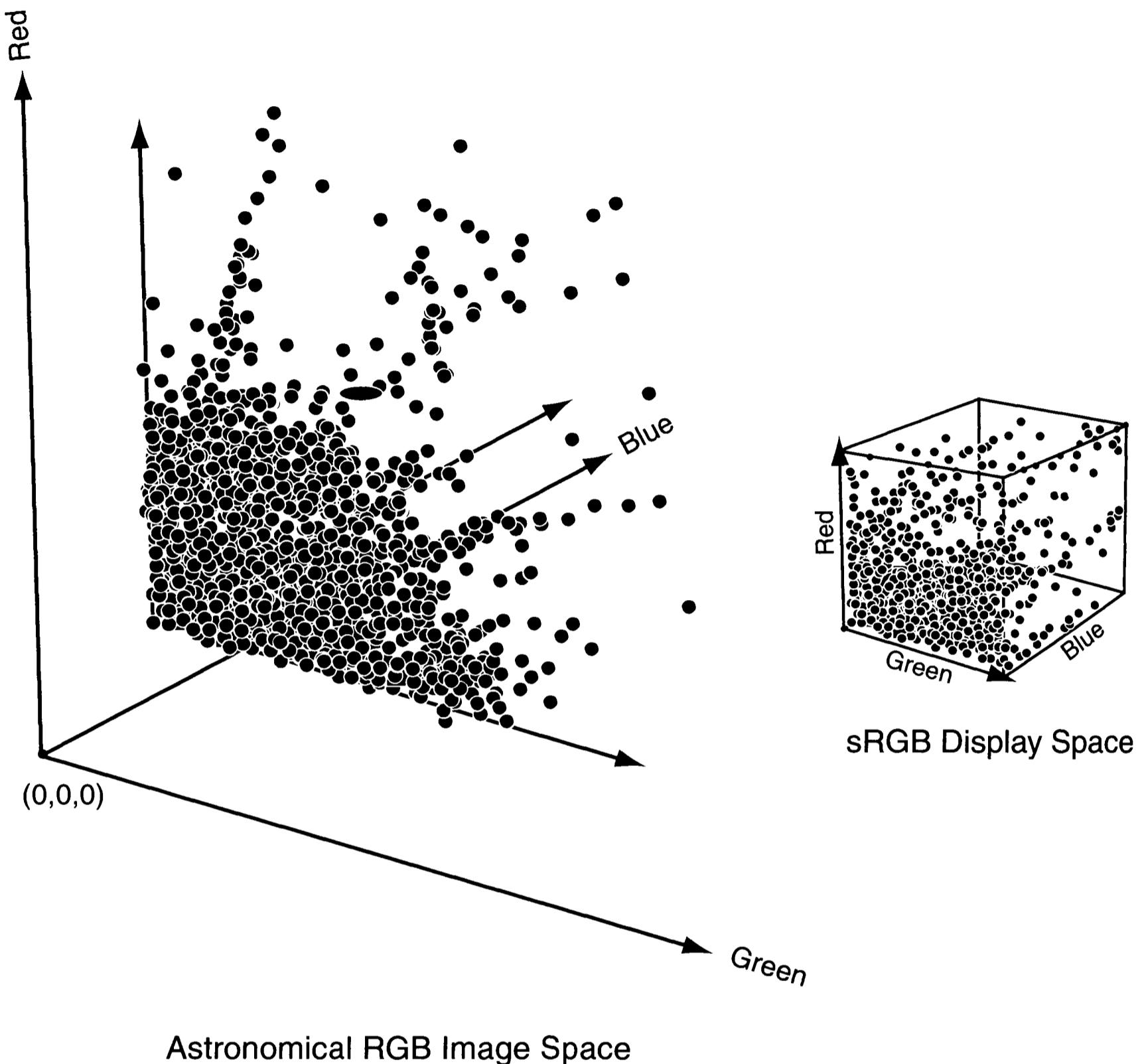


Figure 20.6 RGB image space is virtually without bounds. Even though many pixels cluster near the sky background, star pixels can attain huge values. In sharp contrast, Standard RGB display space is a cube with values in a 0-to-255 range. To make astronomical color images, it is necessary to squeeze the vast RGB image space into a tiny cube of sRGB color display space.

stars and deep-sky objects. Stars in the image appear as lines of pixels extending along a color axis. Deep-sky objects appear as clusters of pixels with values higher than the sky value; if an object appears red, the cluster is offset along the *R* axis; if it appears blue, its pixels are offset along the *B* axis, and so forth.

The pixel values in this plot extend over a very wide range, from a sky value of 100 or 200 ADUs to star images saturated at 65,535 ADUs. In such images, it is possible to pick out the faint outer tendrils of a nebula a few ADUs brighter than sky and then trace streamers to the core of the nebula at 10,000 ADUs—the plot spans a very wide range! However, by the time these data appear on your computer screen, each color must be squeezed into a brightness range of 0 to 255.

Although RGB color space might appear ideal for image processing, in ac-

## Section 20.6: Color Space: Geometric Interpretations of Color

tuality it is not. Suppose an image appears too red: in RGB color space, a logical response is to reduce the red color channel. If you subtract a constant value from red pixel values, the sky becomes green, the image gets darker, and the stars remain white. If you divide the red pixel values, the sky barely changes, the stars become green, and the whole image gets darker. To reduce an overly red image without side-effects, it is usually necessary to both subtract a constant and divide the red pixel values, and also to multiply the green and blue pixel values to maintain the original image brightness.

### 20.6.2 HSL Color Space

Although we capture images in RGB color space, people think in HSL color space, that is, they think in terms of *hue*, *saturation*, and *luminance*. Hue is the color itself: red, orange, yellow, green, blue, violet, and the colors between. Saturation is the intensity of the color: in the case of red, from zero-saturation white, to pale pink, pink, deep pink, light red, and finally 100% saturated deep crimson red. Luminance is the amount of light received, regardless of its hue and saturation.

Geometrically, HSL color space is three dimensional, and takes the shape of a cylinder. Luminance is represented by a distance along the cylinder's axis, saturation by the radial distance from this axis, and hue by an angle around the axis—as shown in Figure 20.7. The color wheel used by artists is a slice through the HSL cylinder.

It is instructive to see how HSL color space fits into RGB color space. The luminance axis of HSL color space runs from black at  $(0, 0, 0)$  in RGB color space to white at  $(255, 255, 255)$ . Every point on the line from black and white corresponds to the color gray; all grays satisfy the condition that  $R = G = B$ . On the luminance axis, saturation is zero and hue is indeterminate. As you move away from the luminance axis, the R, G, and B coordinates are no longer equal; the saturation is greater than zero, and the hue angle defines a color.

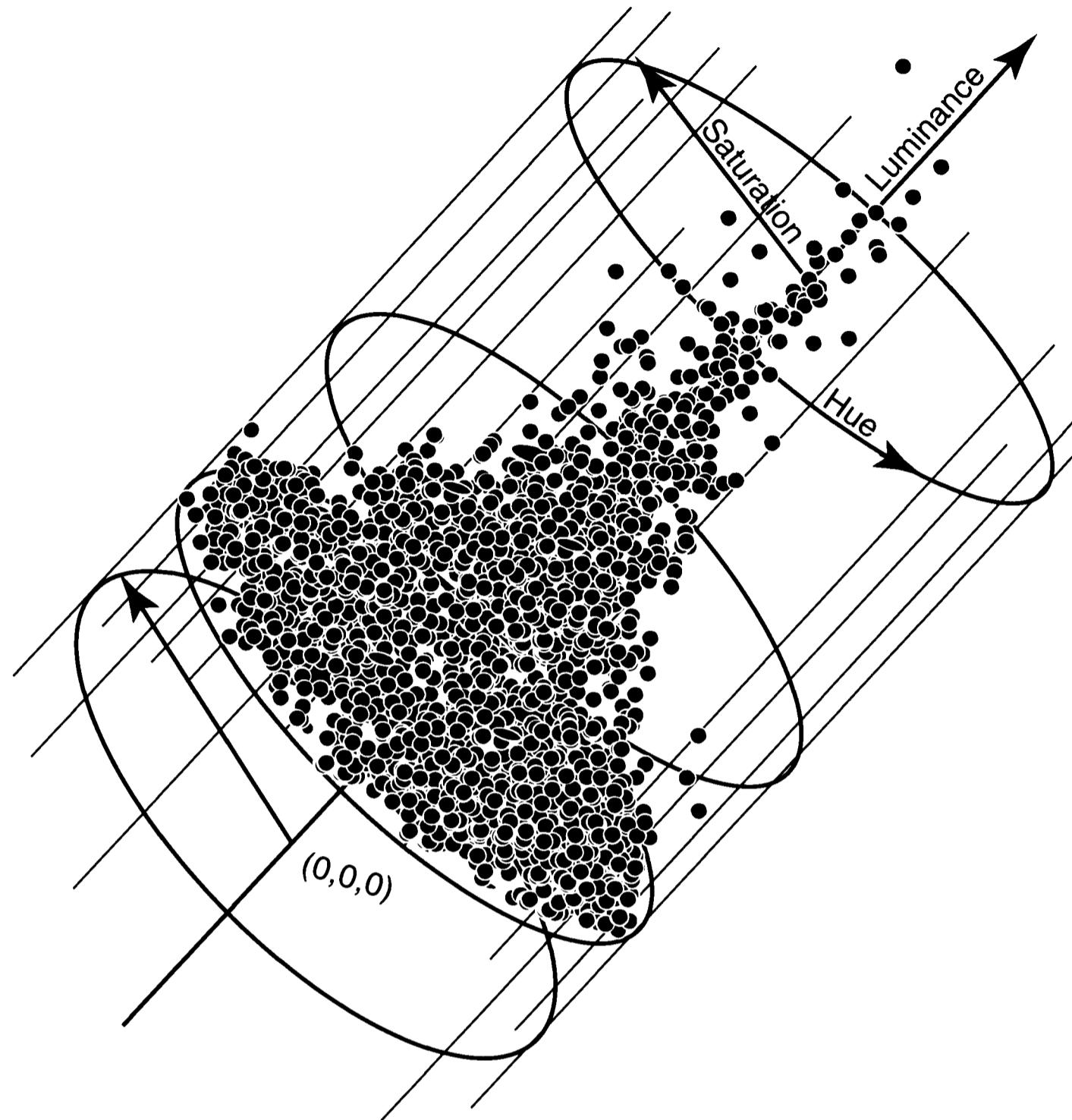
For processing astronomical color images, HSL color space is extremely useful. By converting the RGB coordinates of a pixel into HSL color space, the chrominance components (hue and saturation) are split off from the luminance. Without altering the luminance, it is possible to increase or decrease color saturation. And, without altering the chrominance components, it is possible to compress a wide range of luminance into a short color range that can be displayed on a monitor or printed on a page. Furthermore, processing a luminance channel without touching the chrominance allows you to brighten, darken, smooth, sharpen, deconvolve, or wavelet-enhance images without changing the color balance or adding unwanted artifacts.

**Transforming RGB to HSL.** Since all color spaces describe essentially the same phenomenon, it is possible to transform from one color space to another color space. Given a pixel with red, green, and blue color channels  $R$ ,  $G$ , and  $B$ , the following procedure yields hue, saturation, and luminance channels  $H$ ,  $S$ , and  $L$ :

$$L = \text{Max}\{R, G, B\} \quad (\text{Equ. 20.46})$$

## Chapter 20: Building Color Images

### Astronomical Luminance/Hue/Saturation Color Space



Astronomical HSL Image Space

**Figure 20.7** Humans perceive images as luminance, hue, and saturation. Luminance measures image brightness, while chrominance—color—describes color. Hue measures color itself, while saturation measures the color’s strength. Every color that exists in astronomical RGB image space can be transformed into astronomical HSL image space, and *vice versa*.

$$S = \text{Min}\{R, G, B\} \quad (\text{Equ. 20.47})$$

$$D = L - S \quad (\text{Equ. 20.48})$$

$$S = (L - S)/L \quad (\text{Equ. 20.49})$$

$$H = \begin{cases} R = L \rightarrow 60((G - B)/D) \\ G = L \rightarrow 120 + 60((B - R)/D) \\ B = L \rightarrow 240 + 60((R - G)/D) \end{cases} \quad (\text{Equ. 20.50})$$

where  $D$  is an auxiliary variable, and Min and Max are minimum and maximum operators, respectively.

## Section 20.6: Color Space: Geometric Interpretations of Color

Although it might appear somewhat obscure, the procedure is perfectly logical. Equations 20.46 and 20.47 find the greatest and least color channels, and Equation 20.48 finds the difference between them. Luminance is the color channel with the highest numerical value. Although it is possible to define luminance in other ways, this formulation insures that the dominant color channel will appear bright in the final image.

All three channels share the intensity found in the lowest color channel, and since all three channels share that light, the lowest value represents the white or colorless component of the ( $R$ ,  $G$ ,  $B$ ) triad. To find the saturation (Equation 20.49), divide the non-white color component by the highest color component, or luminance. Saturation can never be less than zero (when all three channels are equal) or greater than unity (when one of the color channels is zero).

The hue is an angle between  $0^\circ$  and  $360^\circ$ . Equation 20.50 selects the dominant color channel, and interprets the ratio between the strength of the greatest color channel and middle color channel as an angle. Pure red has a hue of  $0^\circ$ ; pure green,  $120^\circ$ ; and pure blue,  $240^\circ$ . Intermediate colors are assigned intermediate angles: thus the color cyan, the color complement of red, has a hue of  $180^\circ$ .

When you convert RGB display color space into HSL color space, the luminance,  $L$ , will necessarily range from 0 to 255 because  $R$ ,  $G$ , and  $B$  always lie within that range. Transforming RGB display space results in an HSL display space. However, although luminance is constrained to the range of RGB values, hue and saturation are not. Regardless of luminance, hue stays between  $0^\circ$  and  $360^\circ$ , and saturation stays between 0.000 and 1.000.

Transforming RGB image space into HSL produces an HSL image space with a range of luminance bounded only by the range of values found in the  $R$ ,  $G$ , and  $B$  color images. To convert an HSL image space into HSL display space, it is necessary to change only the luminance component so that it spans a range of 0 to 255; the chrominance components of the image remain the same regardless of changes to the luminance.

By using RGB color space for manipulating and balancing color, and HSL color space for manipulating and scaling luminance, observers can gain access to the best of both worlds.

**Luminance Swapping.** HSL color space splits the luminance and chrominance components of a color image and makes them independent of one another, thereby opening many interesting possibilities. For astronomy, replacing the original luminance image with a better one is perhaps the most important application of luminance swapping. Since the human eye/brain system responds to luminance more strongly than it does to chrominance, observers seeking superior color images can make a set of RGB images and add a deep unfiltered luminance image. The signal-to-noise ratio of the luminance derived from the RGB image set tends to be rather low, but the unfiltered luminance can be exposed deep and end up with a very high signal-to-noise ratio. Once the image is in HSL color space, the original luminance is discarded and replaced with the high-quality luminance image.

**Transforming HSL to RGB.** After carrying out an operation such as lumi-

## Chapter 20: Building Color Images

nance swapping in HSL color space, it is necessary to transform the image back to RGB color space to display it. Given a pixel with a hue, saturation, and luminance of  $H$ ,  $S$ , and  $L$ , the following procedure yields red, green, and blue color channels with values  $R$ ,  $G$ , and  $B$ :

$$i = \text{Int}\{H/60\} \quad (\text{Equ. 20.51})$$

$$f = H/60 - \text{Int}\{H/60\} \quad (\text{Equ. 20.52})$$

$$a = L(1 - S) \quad (\text{Equ. 20.53})$$

$$b = L(1 - (Sf))$$

$$c = L(1 - (S(1 - f)))$$

Then, depending on the value of  $i$ :

$$i = 0 \rightarrow R = L, G = c, B = a \quad (\text{Equ. 20.54})$$

$$i = 1 \rightarrow R = b, G = L, B = a$$

$$i = 2 \rightarrow R = a, G = L, B = c$$

$$i = 3 \rightarrow R = a, G = b, B = L$$

$$i = 4 \rightarrow R = c, G = a, B = L$$

$$i = 5 \rightarrow R = L, G = a, B = b$$

where  $i$ ,  $f$ ,  $a$ ,  $b$ , and  $c$  are auxiliary variables.

Although the process is somewhat obscure, it is logical. The auxiliary variable  $i$  takes the hue and determines whether  $R$ ,  $G$ , or  $B$  is the dominant color, and the auxiliary variable  $f$  is the fraction of the hue comprised of the second strongest color. The auxiliaries  $a$ ,  $b$ , and  $c$  hold the strengths of the middle and lowest color channels based on the saturation. Finally, depending on the dominant color channel,  $R$ ,  $G$ , and  $B$  receive their assigned pixel values.

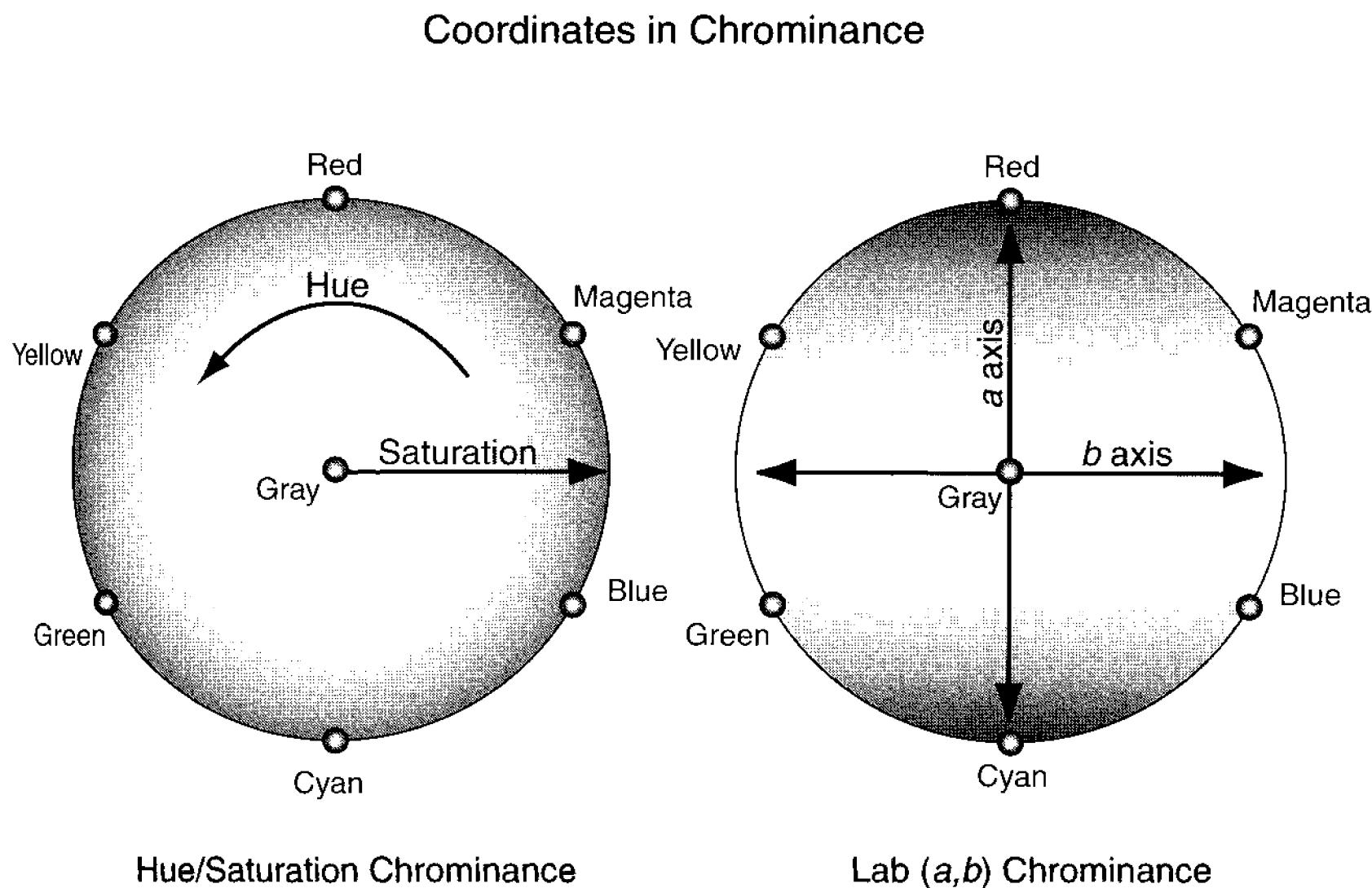
This procedure underscores the separation of chrominance and luminance. The values of  $R$ ,  $G$ , and  $B$  remain scaled fractions of the luminance, retaining unchanged their strict relationship to one another. It is clear that  $H$  and  $S$  control the chrominance—regardless of what happens to  $L$ .

### 20.6.3 Lab Color Space

Lab color space is a color geometry designed as a *uniform* color space. Geometrically, it consists of one luminance coordinate,  $L$ , and two chrominance coordinates,  $a$  and  $b$ —so Lab is not an abbreviation for laboratory, but instead a coordinate system with  $(L, a, b)$  coordinates. Geometrically, the Lab color space is similar to HSL color space, but instead of the hue angle  $H$  and the saturation radius  $S$ , a slice through Lab space reveals an  $(a, b)$  plane perpendicular to the  $L$  axis.

Lab color space corrects a bothersome problem with HSL color space: that  $H$  and  $S$  are not uniform. When  $S$  is large, a small change in  $H$  produces a big change in the color that the eye sees, and when  $S$  is small, a large change in  $H$  produces almost no change in perceived color. Because they are tightly bound together,  $H$  and  $S$  produce a nonuniform color space, a color space in which changes in

## Section 20.7: Luminance/RGB (LRGB) Color Imaging



**Figure 20.8** Chrominance values can be expressed in hue and saturation—an angle and a radius—or as two orthogonal coordinates,  $a$  and  $b$ . Changes in the hue angle and saturation radius produce erratic color changes, whereas changes in the Lab color space  $a$  and  $b$  coordinates are relatively smooth and equal.

the value of a coordinate and the change to the image are only loosely related.

In Lab color space, the  $a$  component represents a redness-greenness axis, and the  $b$  component represents a blueness-yellowness axis. The important characteristic of the  $(a, b)$  plane is that changes in the coordinates produce a more-or-less uniform change in chrominance. Increasing or decreasing the  $a$  coordinate, for example, changes the red/green appearance of saturated colors and unsaturated colors in much the same way.

In astronomical imaging, Lab color space is especially valuable for blending and smoothing irregular or noisy chrominance. In HSL color space, a weakly colored pixel can drain color from nearby saturated pixels, but in Lab color space, pixels with widely differing chrominance have equal influence on a color blend.

- **Tip:** *AIP4Win selects the most appropriate color space for each job or process, and carries out its calculations in that color space. For example, AIP4Win joins colors and adjusts color balance in RGB color space, swaps luminance in HSL color space, and smoothes away sky background noise in Lab color space.*

## 20.7 Luminance/RGB (LRGB) Color Imaging

In LRGB imaging, the observer makes a deep “white-light” image in addition to RGB filtered images. To create a color picture from these data, we combine the

## Chapter 20: Building Color Images

RGB components to create coarse chrominance and luminance data for each pixel in the image, and then replace the original luminance component with a high-quality luminance from the white-light image.

The technique of LRGB imaging works because the human eye and brain readily perceive noise in luminance, but are relatively insensitive to noise and errors in chrominance. Broadcast television exploits this quirk of human physiology by devoting 3 MHz of bandwidth to luminance information but giving only 1 MHz to chrominance signals.

In addition to gains derived from the improved quality of the luminance image, chrominance/luminance imaging gives you the ability to adjust the image color (chrominance) and the image brightness (luminance) separately. You can carefully tweak the RGB controls for perfect color balance knowing that when you adjust the image brightness, the color balance will remain unchanged.

The transmittance of an ideal white-light filter is:

$$T_I = 0, T_R = 1, T_G = 1, T_B = 1, \text{ and } T_U = 0.$$

The ideal filter blocks all light outside the visible part of the spectrum. Filter leakage in the ultraviolet and infrared passbands will increase the signal, but the human eye cannot see this light. Signals that originate outside the range of visibility have the potential to distort brightness relationships in the final color picture.

Upon substituting the filter transmittances into the filter equation (Equation 20.5) and adding the sky contribution, we obtain the following signals from a celestial object:

$$S_L = F_R A_R T_R Q_R + F_G A_G T_G Q_G + F_B A_B T_B Q_B + S_{L_{SKY}}. \quad (\text{Equ. 20.55})$$

Because the spectral response of most CCDs does not match that of the human eye, luminance images made using an “ideal” filter often show excess brightness in red and blue areas of the image. To match the spectral sensitivity of a human observer, the luminance filter should ideally be chosen so that the product of the filter transmittance and the telescope and CCD sensitivity is:

$$T_R Q_R = 0.51 T_G Q_G \quad (\text{Equ. 20.56})$$

$$T_B Q_B = 0.19 T_G Q_G. \quad (\text{Equ. 20.57})$$

However, as a practical matter, few observers want to filter their luminance exposures to match the spectral sensitivity curve of the human eye because doing so results in significant loss of light.

- **Tip:** *AIP4Win stores images internally using the HSL color space. This allows users to apply the same suite of processing tools to color images that they can to monochrome images. When appropriate, however, AIP4Win converts images to RGB or Lab color space for processing, and then returns them to HSL color space.*

## Section 20.7: Luminance/LRGB (LRGB) Color Imaging

### 20.7.1 Creating an Artificial Luminance Image

An alternative to making a luminance image at the telescope is to combine a set of filtered RGB images to create a luminance image. Combining the filtered images produces a luminance image with a higher signal-to-noise ratio than any of the individual filtered ones, resulting in a better color picture.

As we saw above, when RGB color is transformed into HSL, luminance is the maximum value in a color-balanced ( $R$ ,  $G$ ,  $B$ ) triad. This guarantees that any strongly colored object is assigned a high luminance value. However, the human retina responds less strongly to red and blue than it does to green. Strictly speaking, the ( $R$ ,  $G$ ,  $B$ ) triad should be weighted so that:

$$L = 0.299S'_R + 0.587S'_G + 0.114S'_B. \quad (\text{Equ. 20.58})$$

From an aesthetic point of view, weighting the red and blue contributions to luminance in this way makes ruddy HII regions appear too dim and blue reflection nebulae fade into the sky background. To improve the situation somewhat, the ( $R$ ,  $G$ ,  $B$ ) triad can be given equal weighting:

$$L = 0.333S'_R + 0.334S'_G + 0.333S'_B. \quad (\text{Equ. 20.59})$$

As a practical matter, however, most observers want the highest signal-to-noise and color penetration they can get. Those who are willing to sacrifice strict color accuracy will therefore prefer the maximum of a color-balanced ( $R$ ,  $G$ ,  $B$ ) triad as luminance.

### 20.7.2 Enhancing the Luminance Image

In HSL color space, the luminance image can be enhanced by sharpening, deconvolution, nonlinear brightness scaling, digital development, or any combination of techniques without altering the color balance of the color picture. Although enhancements have the potential to give the resulting color image an artificial appearance, they can also be used to improve the aesthetics of it, or to reveal features that might otherwise be difficult or impossible to see.

### 20.7.3 Creating LRGB Images

Prior to generating an LRGB color image, you must have a set of RGB images and a luminance image. To correct for atmospheric extinction and filter variations, it is best to have determined G2V standard star weight factors (see Equations 20.24 through 20.26) for your CCD and filter set.

Begin with these quantities for the RGB images:

- $S_{R_{\text{SKY}}}$ ,  $S_{G_{\text{SKY}}}$ , and  $S_{B_{\text{SKY}}}$ , the pixel value of the residual sky background in each image. Measure this in a star-free region.
- $W_R$ ,  $W_G$ ,  $W_B$ , the weight factor determined from a G2V star.

Begin with these quantities for the luminance image:

## Chapter 20: Building Color Images

- $S_{L_{SKY}}$ , the pixel value of the sky background,
- $S_{L_{MAX}}$ , the pixel value of the brightest significant feature in the image. Although subjective, it can be estimated as a histogram endpoint, typically between 0.99 and 0.9995.

Creation of the color image begins by computing, for each pixel in the RGB image, a set of coordinates in RGB color space:

$$R = \frac{S_R - S_{R_{SKY}}}{W_R}, \quad (\text{Equ. 20.60})$$

$$G = \frac{S_G - S_{G_{SKY}}}{W_B}, \quad (\text{Equ. 20.61})$$

$$B = \frac{S_B - S_{B_{SKY}}}{W_B}. \quad (\text{Equ. 20.62})$$

For each pixel in the luminance image, compute the luminance,  $L$ :

$$L = \frac{S_L - S_{L_{SKY}}}{S_{L_{MAX}} - S_{L_{SKY}}}. \quad (\text{Equ. 20.63})$$

Next, transform the image from RGB color space to HSL color space. The  $(R, G, B)$  triad you have just computed becomes  $(H, S, L)$  coordinates in HSL color space, where you can perform manipulations that are difficult or impossible in RGB color space.

First, you can replace the luminance computed from the  $(R, G, B)$  triad with the unfiltered high-quality luminance image. Note that if you wish to make the image brighter or darker, you can adjust the overall brightness of the image by gamma-scaling the luminance:

$$L' = L^{1/\gamma}. \quad (\text{Equ. 20.64})$$

Increasing  $\gamma$  increases the brightness of the image.

Next, you can control the saturation of the colors in the image. If the colors are too strong, you can reduce the saturation; if you want them more strongly saturated, you can increase the saturation. Since the saturation always lies between 0 and 1, gamma-scaling with a saturation scaling parameter,  $\sigma$ , works well:

$$S' = S^{1/\sigma}. \quad (\text{Equ. 20.65})$$

Increasing  $\sigma$  increases the color saturation of the image.

After completing these image adjustments, the modified image is transformed back to RGB color space. Since the input value of luminance was scaled between 0 and 1, the transformed values of  $R$ ,  $G$ , and  $B$  will lie between 0 and 1. If necessary, adjust these for the gamma of the monitor and scale them into the standard 0-to-255 RGB display color space:

## Section 20.8: Practical RGB and LRGB Color Imaging

$$D_R = 255R^{1/\gamma} \quad (\text{Equ. 20.66})$$

$$D_G = 255G^{1/\gamma} \quad (\text{Equ. 20.67})$$

$$D_B = 255B^{1/\gamma}. \quad (\text{Equ. 20.68})$$

LRGB color imaging is capable of excellent color accuracy, and provides very flexible controls over the final appearance of the color picture.

- **Tip:** *AIP4Win's Join Colors Tool uses the LRGB method for creating color images. If you did not make a luminance image, the software automatically generates one. The Join Colors Tool gives you independent control over the color balance and the brightness of the image.*

## 20.8 Practical RGB and LRGB Color Imaging

Good RGB and LRGB imaging differ little from monochrome imaging. The defining difference is, of course, that you must shoot the components for color images through color filters.

### 20.8.1 Selecting Filters

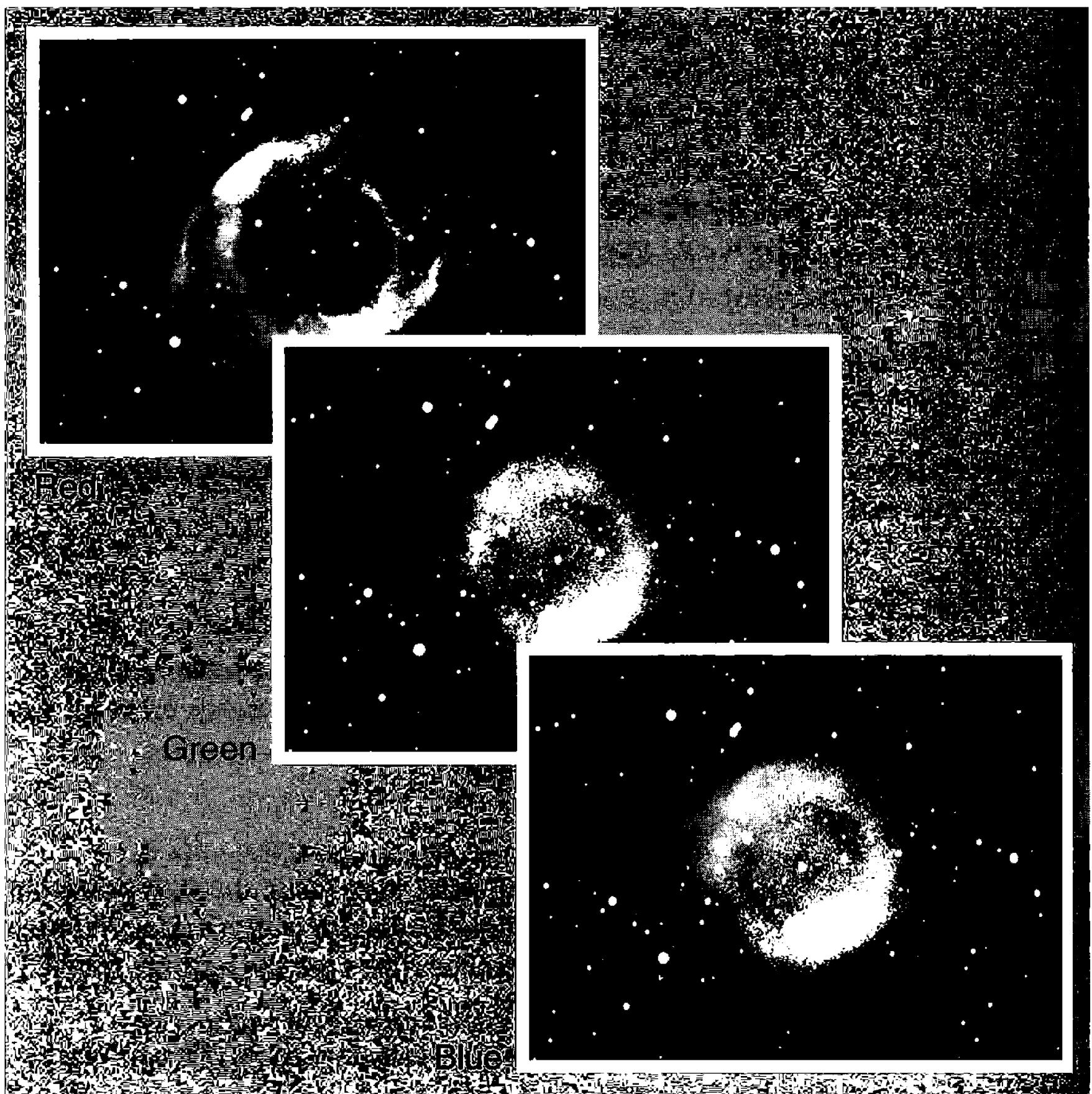
In the early days of CCD imaging, tricolor images were made through gelatin or dyed-glass filters. The classic choices for tricolor imaging were Kodak Wratten filters. Today new high-transmittance interference filters are the best bet for CCD imagers.

**Wratten Tricolor Filters.** Despite their low transmittances, classic Wratten filters used in photography for many years (W#25 red, W#57 green, and W#47 blue) perform extremely well for CCD imaging. The only significant problem with them is the miserably low transmittance values: they require integrations roughly three times as long as modern interference tricolor filters.

**RGB Interference Filters.** CCD imagers have been switching to interference filters because they offer up to 90% transmittance over their passband. Unless they incorporate infrared blocking, however, interference filters should be used with a 700 nm infrared blocking filter. For imaging objects with continuous spectra, interference filters are wonderful. Unfortunately for astronomers, there can be a significant gap in wavelength coverage between 470 nm and 505 nm, the spectrum interval that contains the H $\beta$  and O III lines so prominent in emission nebulae.

**Infrared Blocking Filters.** To the eye, an infrared blocking filter looks like a piece of clear glass. That's because these filters are designed to transmit all visible light. However, a good filter transmits less than 1% of the near infrared, that is, wavelengths longer than 750 nm, which the eye cannot see but which the CCD records all too well.

## Chapter 20: Building Color Images



**Figure 20.9** In strongly colored objects such as the Helix Nebula, the red, green, and blue color channels differ. The red channel has a ring-like structure and a dark center. The green and blue channels are similar, but examine them carefully and you will see differences. Note how the central star is brightest in blue.

**Filters for Luminance Imaging.** For accurate luminance imaging, the exposure *must* be made with an infrared blocking filter. If blocking is not used, objects that emit strongly from 700 nm to 1050 nm—such as class M stars—will record much too strongly, and appear as bright red “stoplight stars” in your final images.

### 20.8.2 Shooting RGB and LRGB Images

RGB and LRGB processes are capable of yielding excellent images; the primary caveat is that you must take exposures long enough to attain a satisfactory signal-to-noise ratio, especially if the CCD has low blue and green sensitivity.

Here are several techniques that contribute to making top-notch RGB and LRGB images:

## Section 20.8: Practical RGB and LRGB Color Imaging

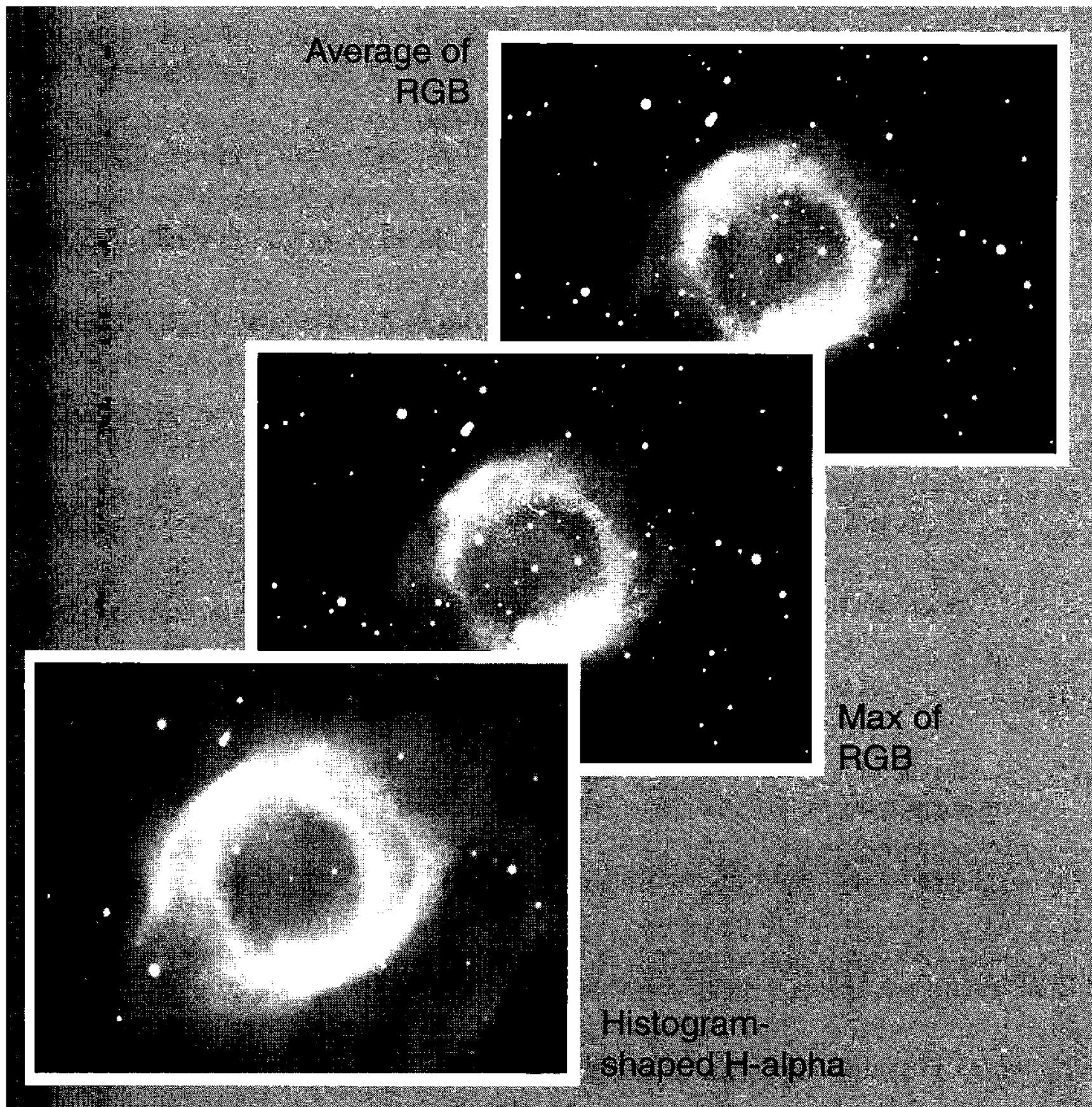


Figure 20.10 LRGB means better flexible control. The luminance can be an unfiltered exposure, or you can generate it by averaging the color channels, taking the maximum in each color channel, or by using narrowband images such as H $\alpha$  either alone or in combination with another luminance image.

- Keep a written record of the images that you take, so you can learn from mistakes and benefit from your successes.
- Don't skip taking darks and flats. Good calibration insures top-notch filtered images, which leads to quality color pictures.
- If at all possible, make your images with a CCD that has good sensitivity in the green and blue. Back-illuminated and blue-enhanced CCDs are best for color imaging.
- Use interference filters. These have high transmission in their passbands, and effective blocking outside them. Look for filters with well-matched passbands.
- Use an infrared blocking filter with your color filters. Many of the latter leak infrared, badly distorting the color balance.

## Chapter 20: Building Color Images

- Also use an infrared blocking filter for accurate luminance. CCDs are so sensitive in the near infrared that an unblocked luminance image is dominated by infrared.
- Make the images with a fast optical system. This reduces the exposure time required to reach a satisfactory signal-to-noise ratio in the sky background. With Schmidt-Cassegrains, use a high-quality focal reducer.
- For faint objects, bin the filtered images to increase their signal-to-noise ratio. To capture image detail, however, do not bin the luminance images.
- If you track-and-stack, make a small number of long exposures rather than a large number of short ones.
- Shoot color images when the subject is high in the sky. The less atmosphere you shoot through, the sharper the images and the more accurate the color.
- If your images have intensity gradients, correct them. Once a gradient is embedded in a color image, it is hard to fix.
- Shoot white balance stars every couple of months to double-check that your filter weights are still valid. Although filter weights tend to be very stable, it is a good idea to validate them periodically.

### 20.8.3 Trade-offs: RGB versus LRGB

LRGB imaging offers a significant benefit over RGB imaging: you can “skimp” on the quality of the filtered images and still recover most of what makes an image look good with the luminance image. Successful LRGB imaging involves coupling acceptable filtered images with an outstanding white-light image.

Although LRGB requires four exposures rather than three, it is more time-efficient. Shooting an RGB color image using 10 minutes in red, 25 minutes in green, and 35 minutes in blue often produces a less pleasing result than shooting 5 minutes in red, 5 minutes in green, 5 minutes in blue, and 15 minutes in white light. The total LRGB exposure time is less than half that of the RGB, and the resulting image will almost certainly look better. Another benefit is that because the luminance image will have a high signal-to-noise ratio, it can be enhanced and non-linearly scaled in brightness.

## 20.9 LLRGB (Multi-Luminance) Imaging

In LRGB imaging, the luminance image controls how strongly each of the color channels appears in the image. If you use a red-filtered image as the luminance, then red objects will appear disproportionately *bright* in the resulting color image. Likewise, replacing the luminance image with the blue-filtered image will enhance the brightness of blue objects in the image. Used with discretion, this technique is a powerful tool because enhancing the brightness of red or blue features

## Section 20.10: Extended-Range and Narrowband Color

in the image does not alter the overall image color balance, but it does permit emphasizing image features.

A useful variation on this technique is to replace the luminance channel with a deep image taken through an H $\alpha$  filter. This strongly enhances the presence of low-excitation red nebulosity. In high-excitation objects, such as supernova remnants, the OIII at 500.7 nm may be dominant in terms of total energy, but because this wavelength often falls into a gap between the transmittance curves of the green and blue filters, this astrophysically important emission will appear too dark. This can be corrected by using an OIII image as luminance.

However, replacing the luminance channel with a non-luminance image is an all-or-nothing proposition. Instead, it is possible to blend *multiple* luminance channels to allow fine control over the resulting image. If a normal luminance image and an H $\alpha$  image can be blended in any proportion, it would be easy to create color images having enhanced nebular brightness without dimming the hot blue stars that illuminate the glowing gas cloud.

Similarly, the color representation of an object that emits strongly in both H $\alpha$  and OIII can be “fine tuned” to show its red- and teal-colored components equally well, rather than strongly favoring one or the other.

- **Tip:** *AIP4Win’s Join Colors Tool accepts two simultaneous luminance channels. The different luminance images can be the same luminance image differently processed (for example, a linear processing and an exponentially histogram-shaped one), a normal luminance and the color channel of your choice, a normal luminance and one H $\alpha$ , or any other combination limited only by the user’s imagination.*

## 20.10 Extended-Range and Narrowband Color

Many people are happy to stick to the familiar and conventional, but color offers tremendous scope for creatively different approaches to displaying astronomical data. Extended-range color, for example, allows you to show the different stellar populations in a galaxy by placing infrared-, green-, and ultraviolet-filtered images into the familiar RGB color channels. Because the spectral range is wider, stars with different temperatures will be more strongly differentiated than they would be with standard red, green, and blue filters.

Variations on the extended-range theme include IRG, IGB, and GBU—but there is really no limit. Images from the Spitzer Space Telescope combine spectral bands in the deep infrared part of the spectrum, and astronomers using the Chandra X-Ray Observatory make color images from different x-ray wavelengths. In both cases, the color encodes astrophysically significant information and helps astronomers to visualize what’s going on.

Narrowband color is another technique that places interestingly different images into the standard RGB color matrix. As we saw at the beginning of this chapter, the spectra of emission nebulae are rich with lines from different elements.

## Chapter 20: Building Color Images

The relative strengths of these lines vary depending on the temperature, pressure, and abundance of the elements that produce them. An image made in the light of three different elements highlights physical features of the nebula that an H $\alpha$ RGB image, dominated by hydrogen, does not. One such combination is H $\alpha$ , OIII, and SII placed in the R, G, and B channels.

Another possibility is to replace color intensity images with color ratio images. In this technique, the red channel is replaced with an image created by dividing the red image by the green image, and the blue channel is replaced with an image created by dividing the blue image by the green image. Images made in this way enhance very small color differences, and allow professional astronomers to map differing element abundances and stellar types in distant galaxies.

- **Tip:** *AIP4Win's color tools should be regarded as just that: software tools to create new, interesting, and exciting types of color images. You can generate color images that combine the output of rank processing and histogram shaping with a linear scaling to explore the morphology of comets....you're limited only by your imagination!*

### 20.11 Color Imaging with CMY Filters

Imaging with CMY (cyan, magenta, and yellow) filters is an interesting variation on imaging with red, green, and blue filters. The techniques share much of their methodology because CMY filters are closely related to RGB ones. Each filter in a CMY set combines two passbands from the RGB set. These combination passbands are:

- Cyan = Green + Blue
- Magenta = Blue + Red
- Yellow = Green + Red.

The important point is that after taking images through CMY filters, it is possible—at least in principle—to recover red, green, and blue signals that can be used to create a color image by adding two of the CMY filtered images and subtracting the third:

- Red = Magenta + Yellow – Cyan
- Green = Cyan + Yellow – Magenta
- Blue = Cyan + Magenta – Yellow.

The reasons usually cited for using CMY filters are:

- Double passbands offer superior wavelength coverage, and
- Combining the passbands generates larger CCD signals.

Of the two, superior coverage of the spectrum is probably the more important. Especially in capturing the light from emission objects, important spectral lines may fall between the single passbands in an RGB filter set. An important

## Section 20.12: The Subjective Side of Color Images

spectral feature in nebulae, the OIII line of doubly ionized oxygen at 500.7 nm, lies right at the boundary between the blue and green passbands. Depending on the filter transmission curves, this line may be missed by one or both filters. With CMY filters, however, the OIII line lies in the middle of the cyan filter passband, where it is certain to be recorded even if neither the magenta nor the yellow filter detect it. No part of the spectrum can be missed with the CMY method.

### 20.11.1 Selecting CMY Filters

For CMY imaging, interference filters are essential. They offer well-controlled passbands and up to 90% transmittance over the filter passband. CMY interference filters should be used with a 700 nm infrared blocking filter unless the CMY filters themselves incorporate it.

The cyan filter in this set bridges the gap in coverage between 470 nm and 505 nm. This part of the spectrum includes H $\beta$  and the 500.7 nm O III lines, which are among the strongest ones in emission nebulae.

Requirements for luminance imaging in CMY are the same as those for RGB. The luminance exposure must be made through an infrared blocking filter that cuts off all light with wavelengths greater than 750 nm. Without infrared blocking, cool red stars show up as vivid red in your images.

### 20.11.2 CMY versus RGB Imaging

The greatest benefit of CMY imaging is guaranteed spectral coverage of important nebular emission lines; the greatest disadvantage is the potential difficulty in creating images with accurate color. With well-matched CMY filters, the results rival images taken with RGB filters. However, it may prove impossible to create accurate color from images taken with a poorly matched CMY filter set.

Once a set of CMY images has been converted into RGB ones, creating color from them proceeds exactly as for normal RGB images.

- *Tip:* *The Color Calculator Tool in AIP4Win takes care of the CMY color math—all the user has to do is measure and enter readings from filtered images of G2V standard stars.*

## 20.12 The Subjective Side of Color Images

Rule #1: Trust the color images that you create.

Amateur astronomers have long been bombarded with color images of celestial objects, many taken on color films under poorly controlled conditions, or digitally manipulated to look “colorful” or “pretty.” As a result, your mental image of how celestial objects look may be quite wrong. If the softness and delicacy of a color image that you have created surprises you, consider the possibility that this is what the celestial object would look like if the cone cells in your retinas responded to faint light.

## Chapter 20: Building Color Images

Rule #2: Have fun with color imaging.

Color is a powerful tool for revealing the information content of images. Experiment with the chrominance by adding different amounts of color and changing the contrast for different colors. Try pumping up the saturation to increase their intensity. Experiment with oddball processing of the luminance component. Replace the original white-light image and experiment with a copy that has been gammalog scaled, Gaussian histogram shaped, or digitally developed to bring out faint details in the white-light image.

---

# 21 Processing Color Images

---

Digital cameras, webcams, and Bayer-matrix astronomical CCD cameras capture color images directly, and after scanning, negatives and photographic prints provide a further source of images that can almost certainly benefit from digital image processing. In this chapter, we examine what to expect in digital color images, how to collect good color images, and how to process color images.

Unlike CCD cameras, digital cameras deliver a color image to you as a complete “package.” With CCD cameras, a color image begins life as a set of stacked red, green, blue, and luminance exposures that must be assembled into a color image—the previous chapter covers how such color images are “built” from scratch. By putting the color content in a ready-to-use package, digital cameras greatly simplify making astronomical color images. The first really sensitive digital single-lens-reflex (DSLR) cameras (such as Nikon’s D70 and Canon’s Digital Rebel and 10D) proved capable of exciting wide-field piggyback shots of the Milky Way, and at the focus of a telescope, could make color “snapshots” of the deep-sky favorites such as the Andromeda Galaxy and the Great Nebula in Orion.

In this chapter we explore the processing of astronomical color images. Although we focus our attention on color images from digital cameras, DSLRs, and webcams, we also address the unique technical problems encountered in processing scanned film negatives and prints.

We begin by examining the properties of color images to see how these set them apart from the monochrome FITS images generated by astronomical CCD cameras. Next, we discuss basic calibration techniques (dark subtraction and flat-fielding) that can often improve image quality, and then combine multiple exposures (a process called “stacking”) to build an image with a high signal-to-noise ratio. Finally, we cover methods of representing color images inside your computer and how these effect color balance and image enhancements such as sharpening, deconvolution, and wavelet sharpening.

## 21.1 Properties of Color Images

Digital cameras, DSLRs, and webcams perform remarkably well in astronomical applications. However, it is important to remember that these cameras were engineered for short-exposure daytime terrestrial imaging rather than low-light long-

## Chapter 21: Processing Color Images

exposure astronomical imaging. It is therefore hardly surprising that the cameras and their file formats are optimized for daytime picture-taking rather than for astronomy. With proper attention to technique, however, you can make excellent astronomical images with them.

To determine how to obtain excellent results from digital camera color image files, we will begin by examining their distinctive properties. (In Chapter 7, we performed a similar analysis for astronomical CCD cameras.) Our goal in this examination is to develop image-processing techniques to exploit the embedded color information while offsetting and overcoming undesirable artifacts.

**Bayer Array Artifacts.** In many digital cameras and webcams, the CCD or CMOS detector has an integral array of red, green, and blue color filters that separate color information into color channels. The filter pattern is called a Bayer array (shown in Figure 1.12), after its inventor. Each pixel on the detector sees only one color. To assign a complete set of color channels to each pixel, the signal processor in the camera averages obtains the other two color channels from adjacent pixels. Thus the color that you see in each individual pixel has actually been synthesized from an area containing five to nine pixels.

If you enlarge a color image from a digital camera so that you can clearly see individual pixels, you will see blocky small-scale artifacts caused by the Bayer array. In the context of a 5- to 8-megapixel image, these artifacts have little or no impact on overall quality, but on small-scale features like star images, these blocky distortions can be annoying.

**8-Bit or 12-Bit Color Depth.** Color images from digital cameras and webcams reach you as JPEG, BMP, and TIFF files, or as proprietary “raw format” files. Color information in the JPEG and BMP file formats is stored as red, green, and blue “color channels” each having an 8-bit depth, for an uncompressed total of 24 bits per pixel. These files are said to contain “24-bit color.” The 8-bit depth means that each color channel has only 256 brightness levels.

Although it seldom matters for daytime applications, the impact of the 8-bit format on astrophotography is severe. Stars and bright objects become easily overexposed, and a bright sky background can fill a significant portion of the dynamic range of the image in a few minutes’ exposure. While an 8-bit dynamic range is adequate for daytime images, it can be too limiting for astronomical work.

The proprietary “raw” formats contain a richer brew of data. Ideally, the raw format would consist of an unaltered “dump” of data directly from the camera’s analog-to-digital converter to the raw file. Most digital cameras have a 12-bit analog-to-digital converter, so the raw data begin with 4096 possible levels of brightness—sufficient range to handle astro-imaging reasonably well. However, some digital cameras appear to adjust or pre-process the raw data by altering the bias level of the data or by filtering out hot pixels, so the “rawness” status of “raw” files is uncertain: they may be “truly raw” or they may be “sort-of” raw.

To convert raw files to standard image format, the camera makers supply adjustment and conversion software. This software mimics the camera’s internal

## Section 21.1: Properties of Color Images

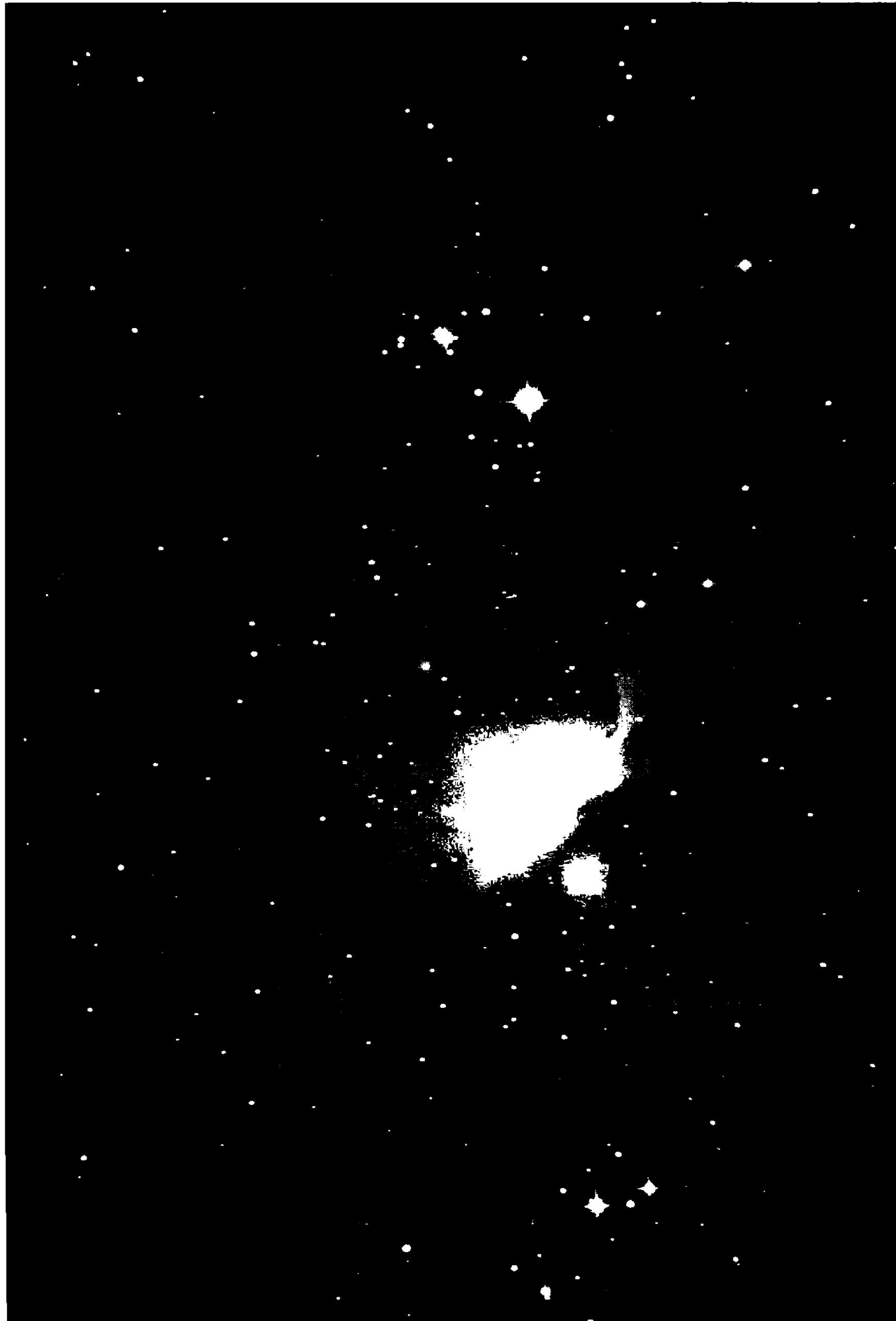


Figure 21.1 This 210-second exposure of the Orion Nebula was taken with a Nikon D70 digital SLR camera at the focus of a 6-inch f /5 Newtonian. The camera's built-in noise reduction feature (i.e., dark frame subtraction) was turned on. Although the bright core of the nebula has saturated and the sky background is somewhat noisy, the image has a pleasing aesthetic appeal.

## Chapter 21: Processing Color Images

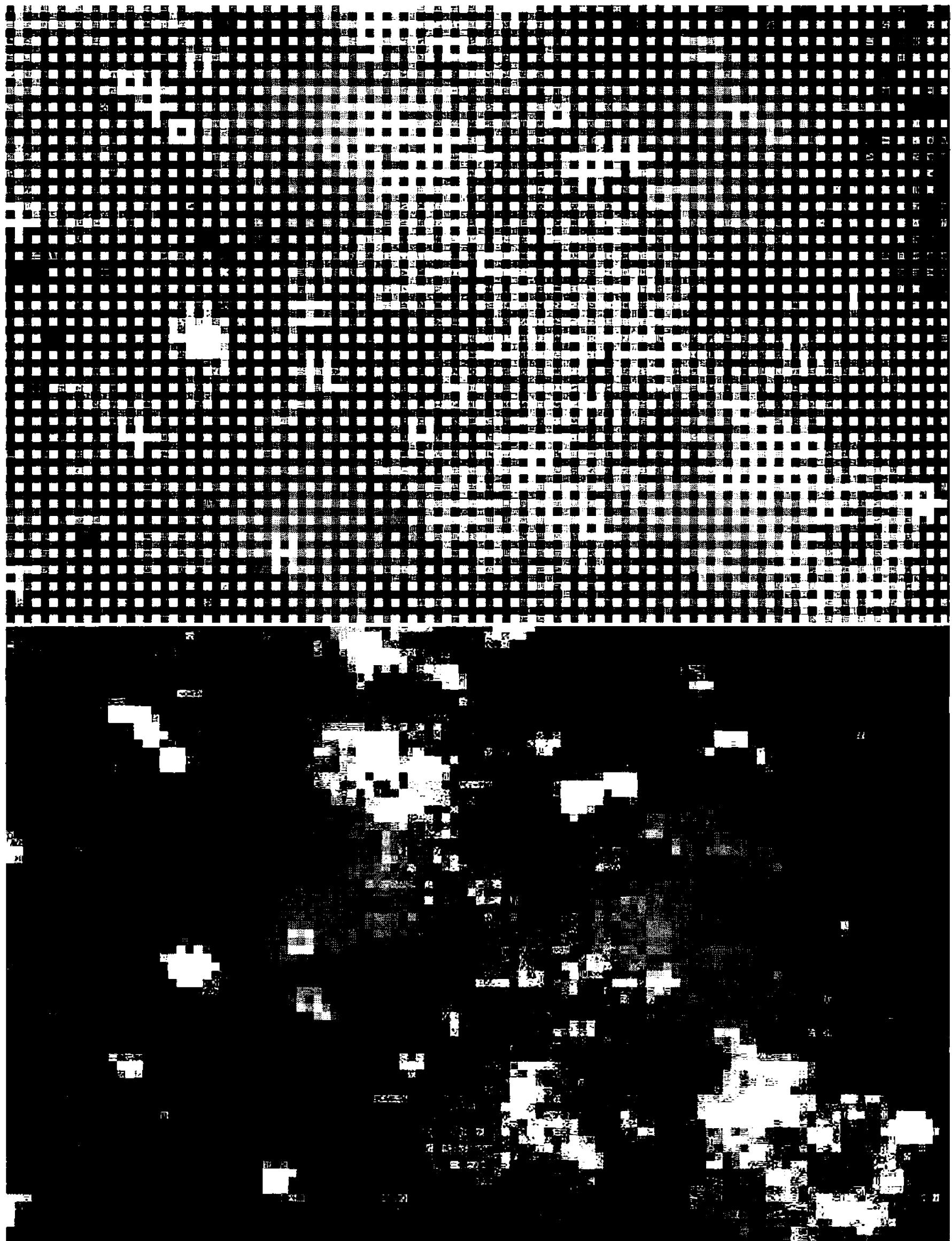


Figure 21.2 The raw image from a digital camera consists of an array of red-, green-, and blue-filtered pixels (top image). After interpolating colors from the pixels around it, each pixel can be assigned a complete (R,G,B) color triad. The full-size decoded image is shown in Figure 21.3, on the page opposite.

processing, providing exposure correction, color balancing, and image sharpening. Among the output options are TIFF files in which the red, green, and blue color channels are stored as 16-bit integers. This format allows you to export the full 12-bit range of data captured by the camera to any software that supports 16-bit or

## Section 21.1: Properties of Color Images

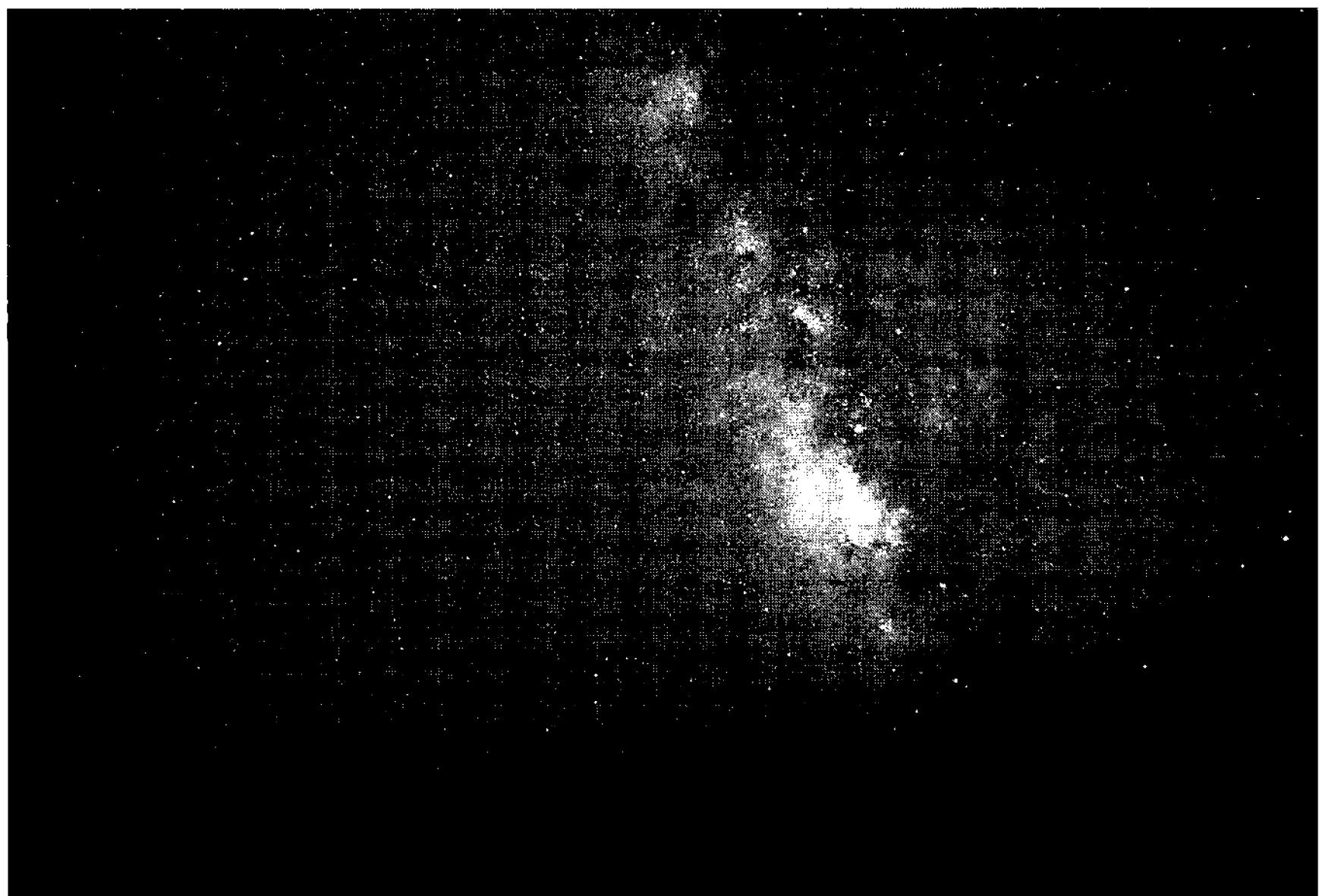


Figure 21.3 Viewed as part of an entire 6-megapixel image, a decoded Bayer array produces smooth color. The 56 x 38-pixel section shown in Figure 21.2 is located in the northern tip of the Sagittarius Star Cloud. Nikon D70 image by David Hayworth, 18 mm lens at f /3.5, 4-minutes, ISO 400.

higher integer data for further processing. Note, however, that even though the 12-bit data have been stored in a 16-bit format, they remain 12-bit data.

- **Tip:** *AIP4Win imports JPEG, BMP, TIFF, as well as proprietary Nikon and Canon raw files. When you import a color image, AIP4Win converts color data to its internal 32-bit floating-point format. This guarantees that absolutely no color or luminance information present in the original image(s) can be lost when AIP4Win processes a color image.*

**Noise and Dark Current.** CCDs and CMOS devices all produce an unwanted signal called dark current. In astronomical CCDs, cooling the sensor to -10 C, -20 C, -30 C, or even lower can reduce dark current to well under one electron per pixel per second. Because they operate at ambient temperatures, however, most digital cameras have much higher dark current levels than they would if they were cooled, and they show a scatter of “hot pixels” in the image. In well engineered cameras, however, dark-frame subtraction reduces dark current and hot pixels sufficiently well to allow multi-minute astronomical exposures with excellent results.

Figure 21.4 shows a dark frame made with a 300-second exposure. In this 8-bit image (0 to 255 ADUs), the dark current averages 4 ADUs, but a few hot pixels have values as high as 120 ADUs. The artifact in the upper left corner, peaking at 200 ADUs, is the glow of the CCD’s on-chip amplifier. The camera’s dark-frame



**Figure 21.4** This 30-second dark frame with a Nikon D70 digital SLR shows amplifier glow in the upper left, a scattering of hot pixels across the image, and weak horizontal banding. The camera’s built-in dark-frame subtraction removes the amplifier glow and hot pixels from images, but faint banding remains.

subtraction (Nikon’s term is “noise reduction”) does a good job of removing both hot pixels and amplifier glow (demonstrated by the near-total absence of amplifier glow and hot pixels in Figure 21.1).

Readout speed is prized in digital cameras, but the 10- to 20-megapixel-per-second readout rate required produces higher readout noise levels than a CCD or CMOS device would be capable of producing if it were read out slowly. In daytime photography, the intrinsic statistical variation in photon count in fully exposed images tends to mask readout noise, but in astronomical images—which consist almost entirely of underexposed sky background—readout noise and readout artifacts may become the dominant source of noise. By using a low ISO speed rating, or by saving files in the camera’s raw format, this effect will be minimized.

**File Compression Artifacts.** Most digital cameras save images in the JPEG file format. JPEG uses a lossy compression scheme, the discrete cosine transform, to reduce the size of the image file. The greater the amount of compression the greater the quality loss. However, files compressed by a factor of 4 (usually called “fine” or “best” mode) show minimal quality loss; those compressed by a factor of 8 (“normal” compression) have losses acceptable for normal scenes, and those compressed by 16 (“basic” or “coarse” compression) may display annoying artifacts.

JPEG compresses an image by dividing it into blocks 8 pixels on a side, computes Fourier components (see Figure 21.5 and Chapter 17), and discards the com-

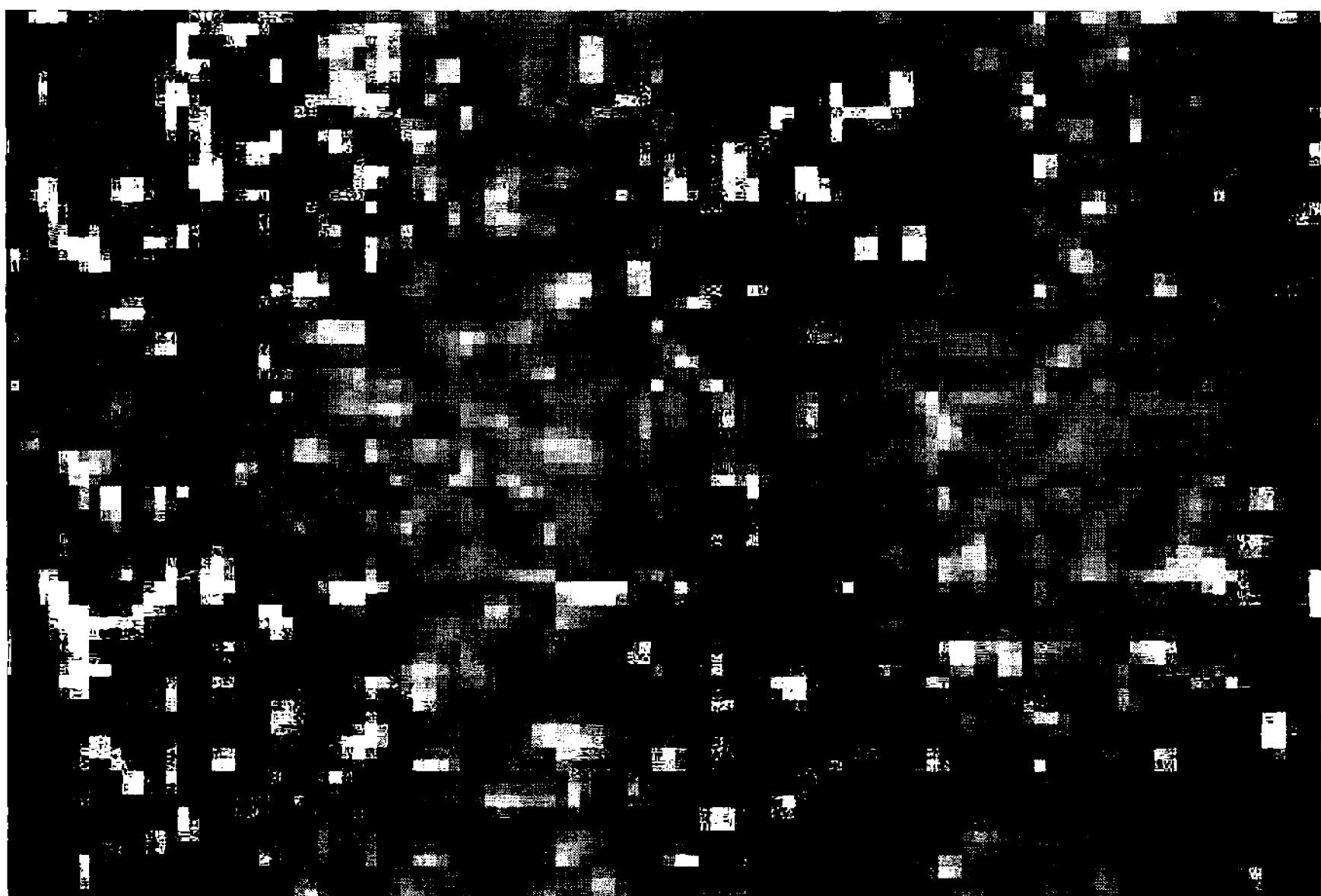


Figure 21.5 In this greatly enlarged close-up, artifacts from lossy JPEG compression appear as eight-pixel-square blocks. If your digital camera supports an uncompressed or “raw” format, save your astronomical images in raw files. If it does not, use the highest-quality JPEG option available.

ponents that have the smallest amplitudes. The greater the compression required, the greater the number of Fourier components sacrificed. Since the least influential components are cut out first, you can use a low compression with little effect on image quality.

With mild compression, JPEG artifacts usually appear as soft or blurry-looking horizontal and vertical bands eight pixels long or high. The effect is most visible in areas where the local pattern of the image noise is disrupted. With strong compression, eight-pixel blocks may have the same brightness or color.

For astronomical images, “fine” or “best” JPEG compression may be acceptable to you. If you find the compression artifacts objectionable, then you should be using your digital camera’s raw file format.

**Film Grain in Scanned Images.** Scanning films and prints is an effective way to bring older images into an image-processing environment. However, in addition to a small amount of digital noise from the linear CCD array in the scanner, you can expect film grain. Unlike the statistical variations due to photon noise where the variation in a pixel is independent of adjacent pixels, film-grain noise is correlated from one pixel to the next. Correlated noise of this type is extremely difficult to filter out, and almost always represents the limiting factor in enhancing scanned film images.

Compared to CCDs and digital camera raw files, scanned color films have a short dynamic range. Color negative materials have the longest dynamic range,

## Chapter 21: Processing Color Images



Figure 21.6 In this greatly enlarged close-up, fixed-pattern artifacts appear as aligned blobs in areas that should appear random. Artifacts can also appear as horizontal bands (see Figure 21.4) and hot pixels. Fixed pattern artifacts can usually be removed from images by dark subtraction.

followed by color transparencies. Scanned prints display a combination of severely limited dynamic range and surface artifacts (fingerprints and fine scratches).

Finally, because many scanners use a linear sensor that is scanned down the image, the images may suffer from a low-amplitude streak pattern. If your scanned images show long linear artifacts, consult the scanner manual; you may find that the scanner software includes procedures for “flat fielding” its scans.

### 21.1 Calibrate to Remove Dark Current and Vignetting

Like their CCD counterparts, images from color digital cameras contain unwanted bias, dark current, and pixel non-uniformities. By applying an image-processing technique called calibration, these artifacts can be largely removed. Section 5.5 of this book contains practical hints on calibration for CCD cameras, and Chapter 6 provides a detailed description of the theory behind image calibration.

Although calibrating color images is fundamentally exactly the same process as calibration of monochrome CCD images, it differs in important details. In this section we discuss the differences and describe taking bias, dark, and flat frames and how to use them to calibrate color images.

#### 21.1.1 Calibrating Bayer-Array Color Images

In the raw image from a digital camera, color information has been spatially en-

## Section 21.1: Calibrate to Remove Dark Current and Vignetting

coded in a Bayer array. However, because unwanted signals (like dark current), noise sources (readout noise and hot pixels), and sensor nonuniformity in the raw image are specific to the source pixel on the sensor, after the raw image has been converted to red, green, and blue (RGB) color channels, noise from the different pixels is mingled and cannot be correctly subtracted.

Instead, the bias and dark current frames should be subtracted, and flat-fielding should be applied before decoding the Bayer array into color channels. This means that to calibrate a Bayer-array image correctly, it must be loaded as a monochrome image. It must be bias corrected and dark subtracted using a bias frame and dark frame in the monochrome configuration, and then divided by a normalized flat frame, also in the monochrome configuration. Once these extraneous signals and noise have been removed, the Bayer array can be properly decoded into red, green, and blue color channels.

### 21.1.2 Calibration for Digital Cameras

Raw images, whether produced by a digital camera or by an astronomical CCD, consist of a dark frame plus a photon image containing vignetting and other non-uniformities. Calibration is an image-processing procedure that enables you to remove these degrading effects.

In this section, we examine in detail how image calibration works. We'll begin by looking at the components contained in a raw image:

$$\langle \text{RAW} \rangle_{x,y} = \langle \text{DARK} \rangle_{x,y} + \langle \text{VIGN} \rangle_{x,y} \times \langle \text{PHOTONS} \rangle_{x,y}. \quad (\text{Equ. 21.1})$$

The  $x, y$  subscripts remind us that a term such as  $\langle \text{DARK} \rangle_{x,y}$  refers to the image as a collection of pixels at every pixel location.

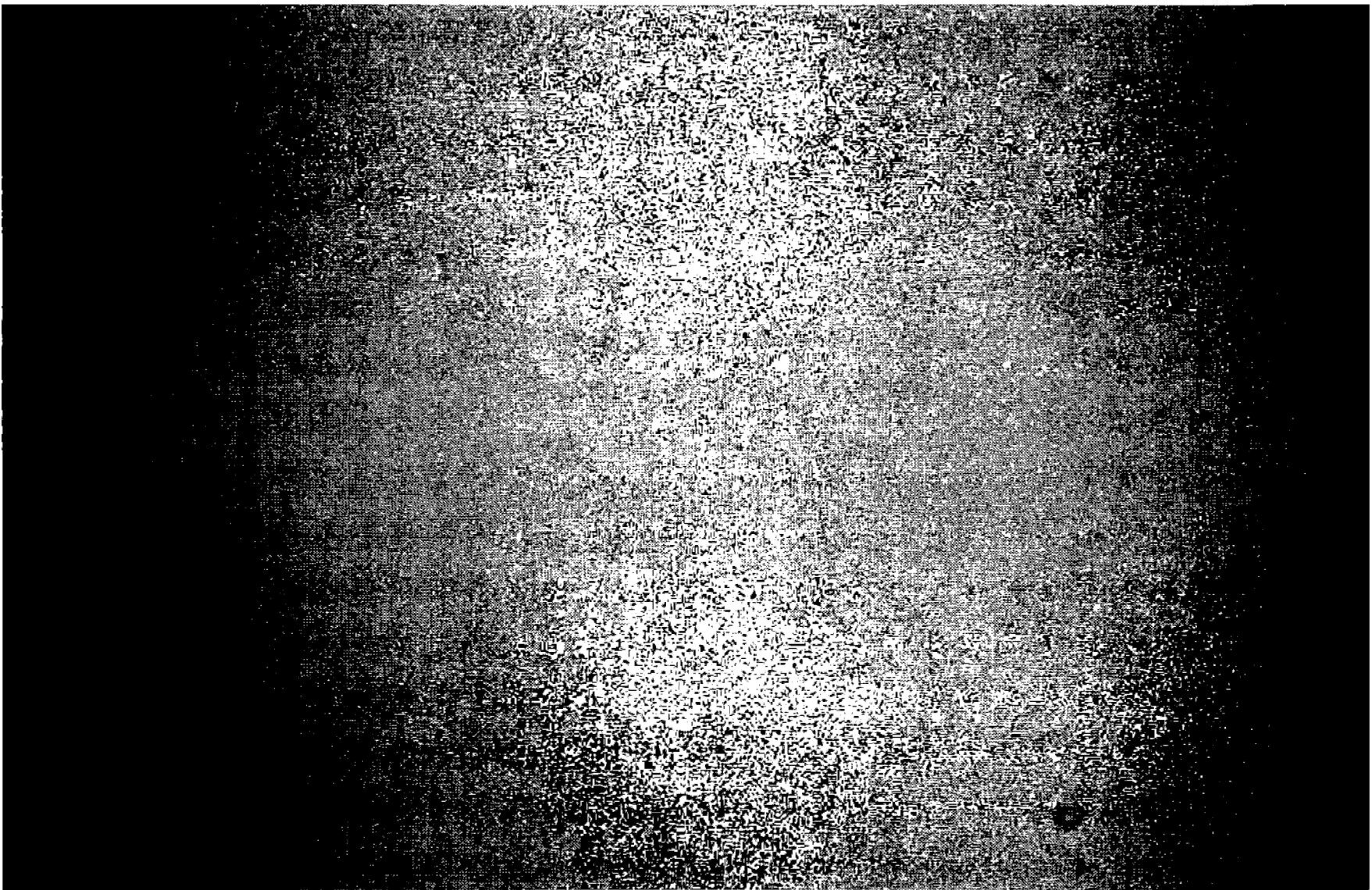
The goal of calibration is to extract the image  $\langle \text{PHOTONS} \rangle$  from the image  $\langle \text{RAW} \rangle$ . To accomplish this, we can rearrange the terms in Equation 21.1, thus:

$$\langle \text{PHOTONS} \rangle_{x,y} = \frac{\langle \text{RAW} \rangle_{x,y} - \langle \text{DARK} \rangle_{x,y}}{\langle \text{VIGN} \rangle_{x,y}}. \quad (\text{Equ. 21.2})$$

To obtain  $\langle \text{PHOTONS} \rangle$ , we need both  $\langle \text{DARK} \rangle$  and  $\langle \text{VIGN} \rangle$ . Making a dark frame is easy: you take an exposure with the shutter closed or with the lens cap on the camera. Because the amount of dark current depends on the exposure time, the exposure times you use for the image  $\langle \text{RAW} \rangle$  and the image  $\langle \text{DARK} \rangle$  must be the same.

Obtaining  $\langle \text{VIGN} \rangle$  is somewhat more involved, but it is not difficult. To record the non-uniformities, you make an image of a uniform field of light—a “flat field.” Since dust or vignetting will cause spots and dark edges, the flat field image, called  $\langle \text{FLAT} \rangle_{x,y}$ , is a map of these nonuniformities. Figure 21.7 shows a typical flat-field image.

To give a short exposure time, the uniform source should be fairly bright. However, since the image  $\langle \text{FLAT} \rangle$  is an image like any other, you must make a dark frame for it, too. This dark frame is  $\langle \text{FDARK} \rangle$ . With the addition of the flat-



**Figure 21.7** A flat-field image is nothing more complicated than a picture of an illuminated screen placed close to your camera or telescope. The dark corners are caused by vignetting, and the little “donut” at the lower right is a shadow cast by dust in the optical system.

dark frame, Equation 21.2 becomes:

$$\langle \text{PHOTONS} \rangle_{x,y} = \frac{\langle \text{RAW} \rangle_{x,y} - \langle \text{DARK} \rangle_{x,y}}{\langle \text{FLAT} \rangle_{x,y} - \langle \text{FDARK} \rangle_{x,y}}. \quad (\text{Equ. 21.3})$$

Although it is tempting to make only one dark frame, the exposure times for  $\langle \text{RAW} \rangle$  and  $\langle \text{FLAT} \rangle$  are rarely the same, so the exposure times you use for  $\langle \text{FLAT} \rangle$  and  $\langle \text{FDARK} \rangle$  will also be different. As a result, it is necessary to make two dark frames, one for the raw image and one for the flat-field exposure.

Although the formulation in Equation 21.3 works, dividing by the flat frame produces pixel values that are inconveniently small. To produce values that average around the values in the raw image, it is necessary to normalize the flat frame, that is, to divide the flat frame by the mean value of  $\langle \text{FLAT} \rangle_{x,y} - \langle \text{FDARK} \rangle_{x,y}$ , which we'll call  $\bar{F}$ , to create a normalized master flat frame,  $\langle \text{MFLAT} \rangle$ :

$$\langle \text{MFLAT} \rangle_{x,y} = (\langle \text{FLAT} \rangle_{x,y} - \langle \text{FDARK} \rangle_{x,y})/\bar{F}, \quad (\text{Equ. 21.4})$$

and then compute the calibrated image from:

$$\langle \text{PHOTONS} \rangle_{x,y} = \frac{\langle \text{RAW} \rangle_{x,y} - \langle \text{DARK} \rangle_{x,y}}{\langle \text{MFLAT} \rangle_{x,y}}. \quad (\text{Equ. 21.5})$$

Calibration for monochrome CCD images is done this way. However, with Bayer-array color images from digital cameras, Equations 21.3 and 21.5 introduce

## Section 21.1: Stacking to Enhance Signal-To-Noise Ratio

an undesirable side-effect: calibration shifts the color balance of the image.

Suppose that the uniform source that you use for the flat-field image is a tungsten lamp, and therefore yellow-orange rather than white. If the mean value of the red-filtered pixels in the Bayer array is greater than that of the green-filtered and blue-filtered mean, then when the Bayer array image is flat-fielded, the value of red pixels will decrease and those of green and blue pixels will increase, shifting the color balance of the image toward the color complement of the flat-field image—in this example toward a greenish-blue cast.

To avoid this effect, instead of normalizing all pixels in the flat-field image, it is necessary to normalize the red, green, and blue pixels separately. The mean of the red pixels would be  $\bar{F}_R$ ; the mean of the green pixels,  $\bar{F}_G$ ; and that of the blue pixels,  $\bar{F}_B$ . The normalized master flat frame is computed as follows:

$$\langle MFLAT \rangle_{x,y} = \begin{cases} \text{if R} \rightarrow (\langle FLAT \rangle_{x,y} - \langle FDARK \rangle_{x,y}) / \bar{F}_R \\ \text{if G} \rightarrow (\langle FLAT \rangle_{x,y} - \langle FDARK \rangle_{x,y}) / \bar{F}_G \\ \text{if B} \rightarrow (\langle FLAT \rangle_{x,y} - \langle FDARK \rangle_{x,y}) / \bar{F}_B \end{cases} . \quad (\text{Equ. 21.6})$$

For red pixels, normalizing uses the mean of the red pixels, and for green pixels and blue pixels, uses their means. Calibration is completed by applying Equation 21.5. This calibration procedure does not change the color balance of the image.

The value of image calibration is shown in Figure 21.9.

Indeed, calibration is so important for good image quality that digital camera makers allow serious photographers to calibrate their images. In the Nikon D70, for example, dark-frame subtraction is an optional built-in function. If you set the “long exposure noise reduction” option to “ON,” the camera automatically makes and subtracts a dark frame for any exposure longer than one second. Furthermore, you can perform flat-fielding if you make a “Dust reference photo” of an out-of-focus white object 4 inches in front of the camera lens. The dust reference photo is a flat-field frame. Nikon’s Capture Editor software uses the dust reference photo to perform a flat-field division.

- **Tip:** *Color calibration in AIP4Win uses the algorithms described in this section to calibrate single images or to calibrate multiple images during stacking. Calibration plus stacking is the key to producing high-quality images with your digital camera.*

### 21.1 Stacking to Enhance Signal-To-Noise Ratio

Even after careful calibration, an image made with a single exposure seldom possesses that satisfying sense of “depth” that makes a great astrophoto. In technical terms, lack of “depth” usually translates as a poor signal-to-noise ratio. We discuss signal-to-noise ratio at length in Chapter 2. The fundamental idea is that image quality increases as you accumulate more exposure time. Although it is not prac-

## Chapter 21: Processing Color Images



Figure 21.8 Case study: Stacking turns noisy images into great astro-images. The upper image is a single 30-second exposure; sensor noise dominates the image of M31. By stacking fifteen 30-second exposures, the noise has been defeated and a robust image of the galaxy now rules the noise.

tical to make exposures longer than a few hundred seconds, with digital images you can add (or “stack”) multiple images together to make one very “deep” image.

In astronomical imaging, stacking provides additional benefits. If your telescope does not track precisely, or if you do not have an autoguider, exposing long-

## Section 21.1: Stacking to Enhance Signal-To-Noise Ratio

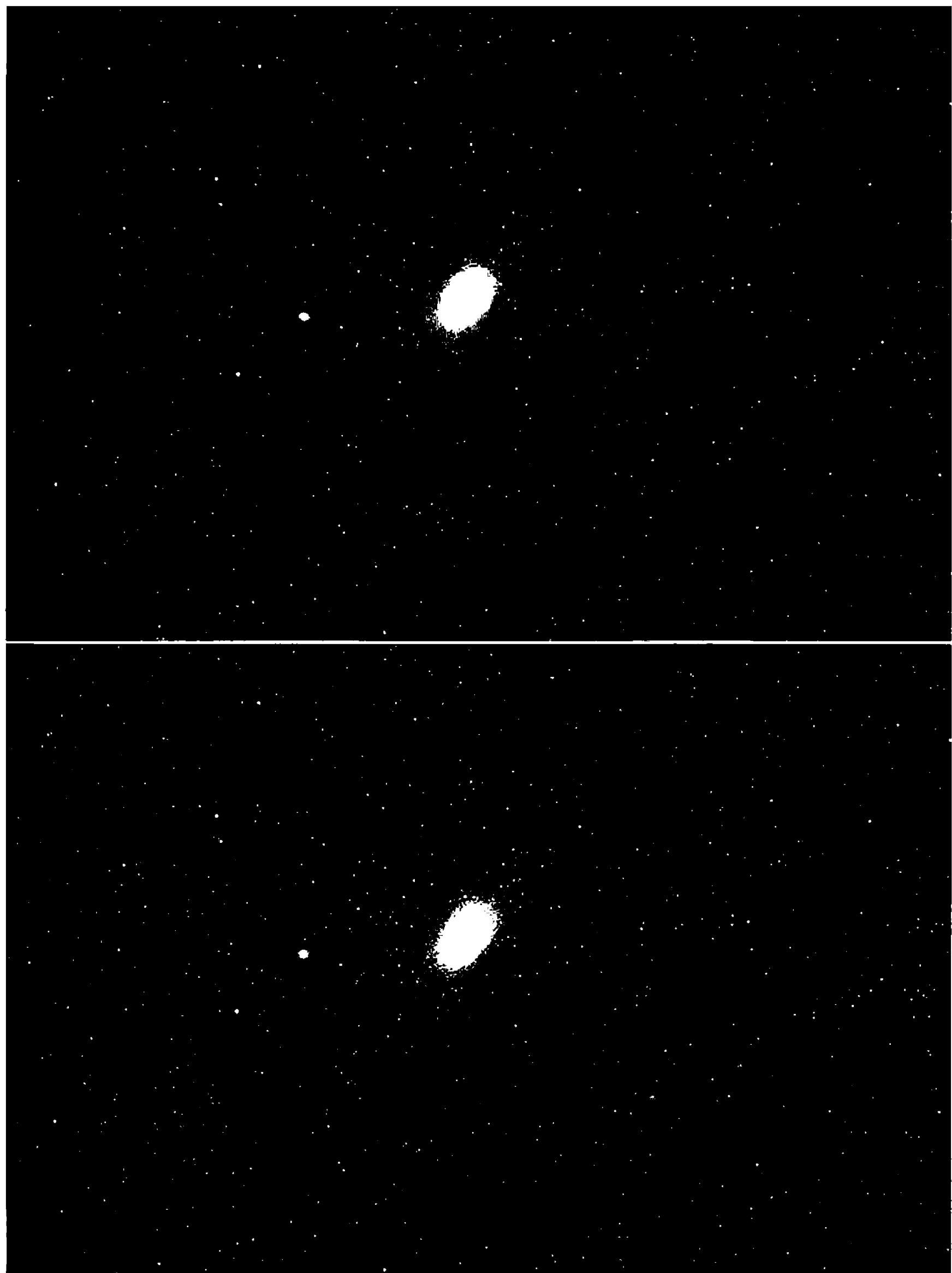


Figure 21.9 Case study: Calibration cuts dust donuts, vignetting, and dark current. The top image is a stack of fifteen 120-second uncalibrated exposures, and it is marred by dust donuts and vignetting. The bottom image was made from the same fifteen images, but this time they were calibrated as they were stacked.

er than a minute or two produces trailed images. However, in short exposures the trailing will be too small to matter. By shooting multiple short exposures and stacking them, you can make the equivalent of a long exposure without needing a telescope mounting that tracks precisely.

## Chapter 21: Processing Color Images

To make stacking practical for astronomy, however, before a set of imperfectly tracked exposures can be stacked, it is necessary to align (or “register”) them so that the same image features fall in the same location in every image. In Section 16.4 we describe how an image can be shifted, scaled, and rotated to exactly match another. Stacking a set of images usually involves selecting the ones to be stacked, choosing options such as calibrating each image before stacking it, and finally selecting one or more stars to serve as reference points. The software then automatically loads each image, calibrates it, finds the reference stars, shifts the image, and adds it to an internal buffer that is accumulating the stacked image. Figure 21.8 shows the value of stacking images, and Figure 21.9 demonstrates the importance of obtaining calibration frames and calibrating the images during stacking.

- **Tip:** *AIP4Win can calibrate and stack digital camera images from exposures stored in a variety of color image formats. Stacking images stored in the JPEG format works reasonably well, especially when you set your camera’s “long exposure noise reduction” option to “ON” when you make the images. For the very best results, you should store images in your camera’s raw file format.*

## 21.2 Making Great Images with a Digital Camera

This section summarizes techniques that lead to outstanding deep-sky images with a digital camera. Because these techniques are based on the fundamental properties of photon counting, sensor noise, and image statistics, they are valid with any and every digital camera.

**Shoot under Dark Skies.** The darker your sky, the better your pictures will be. In the sensor of your digital camera, sky background photons increase the photon event count without adding useful image information, while adding to the statistical uncertainty of the photons from your deep-sky target object.

**Shoot with a Fast Optical System.** Since noise sources like dark current increase with longer exposure times, using a fast optical system means delivering more photons to each pixel, and therefore less time required to get a fully exposed image. In the context of digital cameras, “fast” is a relative term. With a camera lens,  $f/2.8$  is not particularly fast, but with a telescope, such a low focal ratio is virtually unheard of—so a better way to approach the issue of focal ratio would be to say, “Avoid slow focal ratios.” Use the fastest optics you can consistent with the type of optical system you’re using, good image quality, and low vignetting.

**Avoid Optics with Vignetting.** Optical systems that vignette deliver less light to the edges of your images than they do to the center. The result is that you get images with dark corners. Although flat-fielding corrects vignetting, the less work your flat-frames have to do, the better your results will be. Fast optical systems tend to have worse vignetting than slower optics—not always, but generally. Try to find a good compromise between focal ratio and vignetting.

## Section 21.2: Making Great Images with a Digital Camera

**Stack your Images.** Nothing beats stacking for collecting lots of target-object photons. The reason is that you can stack and stack and keep on stacking images even if your telescope has a mediocre drive system and you cannot afford a fancy autoguider. Shooting 50 exposures that are each 2 minutes long typically yields something in the neighborhood of 40 images that are well tracked and free of airplane trails, and 10 images with defects that you can discard.

Technically, stacking makes efficient use of the limited full-well capacity of your digital camera's sensor and limited number of brightness levels captured in 12-bit images. It's the smart way to capture lots of photons with the sensor types used in digital cameras.

**Store Images in your Camera's “Raw” Format.** The standard 8-bit JPEG format of digital cameras yield excellent daytime images; for serious astronomical imaging, use your digital camera's 12-bit “raw” format. Not only does a 12-bit raw mode store the dynamic range suitable for astronomical images, but it bypasses in-camera processing that is almost certainly inappropriate for astronomical images. If your camera supports a “raw plus JPEG” mode, use it so that you can easily and quickly check the images before stacking them.

**Shoot Using a Low ISO Speed.** The high ISO settings in digital cameras multiply pixel values and noise from the sensor by the same amount; and, since noise is adequately sampled at low ISO ratings, when you shoot using the “raw” format, you gain nothing by using a high ISO setting. However, if you shoot compressed JPEG images, the dynamic range of the captured data is greatly reduced, and it will be necessary to use a high ISO speed setting.

**Make Dark Frames and Flat Frames.** If you don't shoot dark frames and flat frames when making images, you won't be able to remove dark current, fixed-pattern noise, and vignetting from your images. Treat these auxiliary images as a necessary part of your imaging routine and you'll be better able to produce exciting results with your digital camera.

**Mix Long and Short Exposures.** In your digital camera's sensor, object photons compete with dark current, readout noise, amplifier glow, fixed pattern noise, and a myriad other unwanted signals and noise. As a general rule, the longer your individual exposures, the better the ratio between object photons and everything that competes with them.

However, it is also necessary to balance the gain from long exposures against overexposing the stars and the brightest parts of your target object. In digital imaging, once a pixel saturates, its signal says nothing but “overcooked.”

As a hedge against overexposure, plan your image sessions to include a few short exposures (10 seconds or so), a few somewhat longer (20 seconds), a few mid-length exposures (60 seconds), and as many long ones (120 to 300 seconds) as you can stand making. When you stack the images, the short exposures will fill in the “overcooked” portions of the long ones, and you will capture the full range of brightness in your subject.

**Dither Tracking.** Although we didn't dwell on this when we discussed im-

## Chapter 21: Processing Color Images

age artifacts, most of the artifacts in digital camera images always occur in the same pixels. Dark current, fixed pattern noise, hot pixels, and even JPEG artifacts do not move in the image. If you make a stack with perfectly tracked images, the artifacts can add together just as well as features in the image do.

So—instead of tracking with great precision over many images, track each well, but change the telescope pointing slightly—by a dozen or so pixels—from one image to the next. When you stack your images, the processing software will register them so that image features add—but artifacts will be spread around and averaged out.

The sequence of images shown in Figures 21.8 and 21.9 demonstrates a progression in quality from a single short exposure to a stack of fifteen relatively lengthy exposures—the last image made with 60 times as many photons as the first. The bottom line to making a great image: Expose deep to collect lots of photons and calibrate the images to overcome dark current and vignetting.

### 21.3 Color Images in Color Spaces

Most digital camera images begin life as a spatially-encoded RGB Bayer array. *Spatially encoded* means that a pixel’s position on the sensor determines the color filter attached to it. In a typical Bayer array, pixels with red and blue filters alternate with pixels that have a green filter (see Figure 20.5). Although this geometry makes it possible to make color images with a single-layer CCD or CMOS sensor, the Bayer array must be *decoded* to produce a standard color image.

In standard color images, every pixel contains an RGB triad,  $\{R, G, B\}$ , that is, it contains a red intensity, a green intensity, and a blue intensity. In ordinary color images, the intensity values are represented by 8-bit unsigned integers ranging from 0 to 255—but they can equally well be stored as 12-bit integers, 16-bit integers, or floating-point values.

In the Bayer array, each pixel contains only one of the three color channels. Although the actual encoding varies from one sensor to the next, it is usually something like this:

- If  $x$  is odd and  $y$  is odd, then the pixel is the blue channel.
- If  $x$  is even and  $y$  is even, then the pixel is the red channel.
- Otherwise, the pixel is one of two green channels.

The complete  $\{R, G, B\}$  triad for each pixel is constructed by averaging the adjacent pixels. Each red pixel is surrounded by four blue ones and four green ones; the red channel is simply the value of the red pixel, and the green and blue channels are computed by averaging the adjacent green and blue pixels. Likewise, each blue pixel is surrounded by four red and four green pixels, which are similarly averaged. Around each green pixel are two red, two blue, and four green pixels; to decode them, the red and green pairs are averaged. After decoding, each pixel contains a complete  $\{R, G, B\}$  triad.

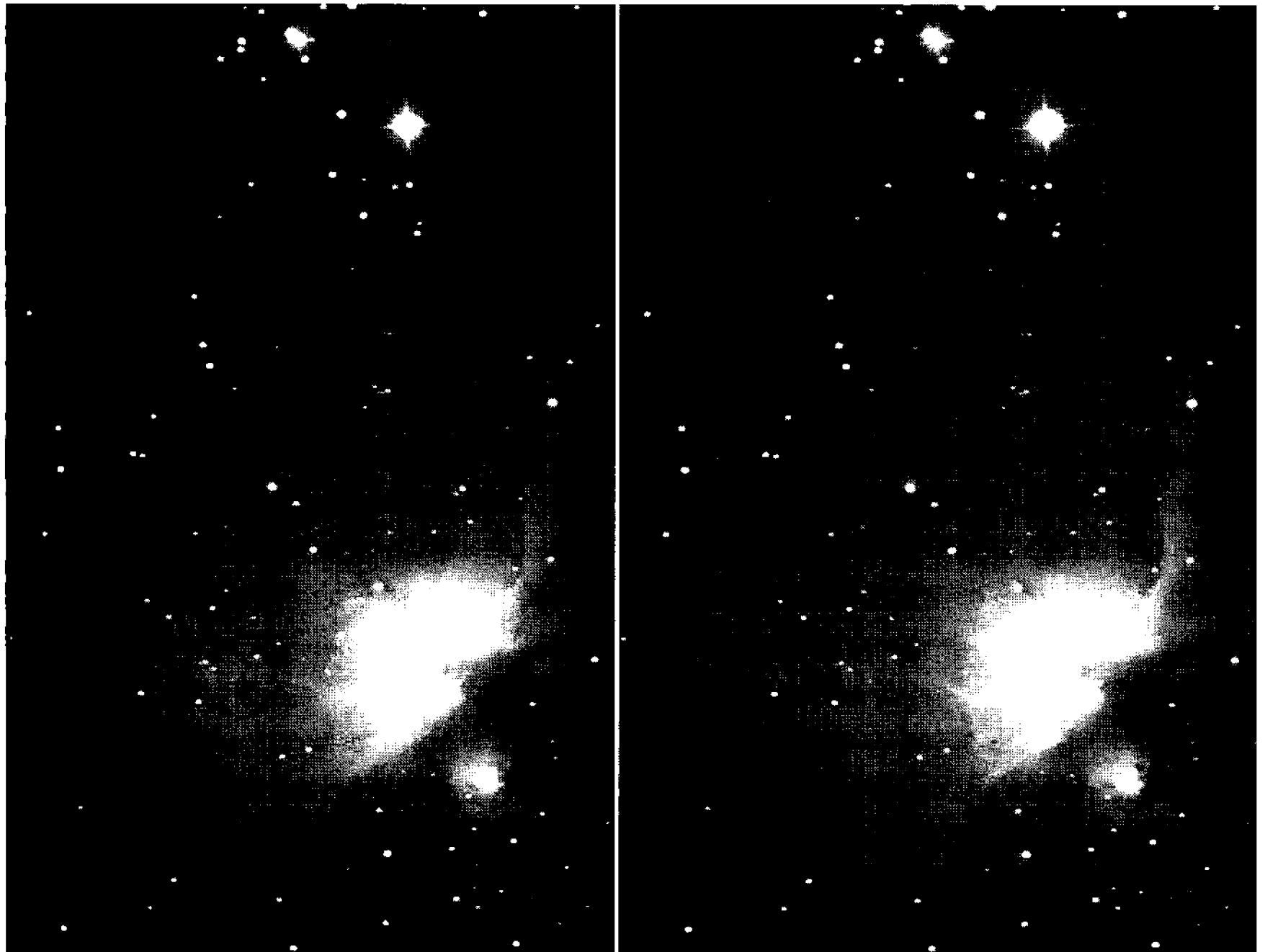


Figure 21.10 Subtle differences between the red (left) and blue (right) color channels account for the colors seen in color images. The inner regions, for example, are slightly brighter in the red channel, so they appear ruddy in the color images. The slightly lighter background makes the sky appear blue.

### 21.3.1 RGB Color Space

In color images, each pixel has five associated values:  $x, y, R, G$ , and  $B$ . The  $(x, y)$  coordinates tell the pixel's location in image space, and the  $\{R, G, B\}$  triad tells where the pixel lies in color space. Imagine that instead of plotting the pixels in an image by their  $(x, y)$  location in a image, you were to plot each pixel by its  $\{R, G, B\}$  coordinates in color space. An 8-bit image would look like a cube with one corner with  $R = G = B = 0$  and the opposite corner  $R = G = B = 255$ . Scattered through the cube you would see thousands or millions of pixels, each at a location determined by its color.

In image space, to make an image look wider, you stretch its  $y$  axis. The analogous operation in color space, to make the image look redder, is to stretch its  $R$  axis. Stretching the  $R$  axis of the RGB color space of an image is exactly the same as separating the image into three *color channels* or *color planes* and stretching the red color channel. Almost every image-processing program ever written includes controls for stretching and compressing the red, green, and blue color channels. This is conventionally implemented as three histograms with sliders that enable the user to set the black and white point of each color channel, and a third slider for gamma-scaling grayscale values between black and white. By changing

## Chapter 21: Processing Color Images

the slider settings, the user can adjust the redness, greenness, and blueness of the image.

Fortunately, RGB color space is capable of more subtle color manipulations, and in fact, more substantial manipulation is required to convert a digital camera's native color space to one of the standard color spaces, such as sRGB, used by computer monitors and color printers. Imagine that instead of simply stretching the RGB color cube, you could twist it sideways, squeeze it, and turn it around until the RGB triads in your camera's native color space lined up with the proper RGB triads in the standard color space. This is exactly the manipulation accomplished by a color matrix multiplication. Symbolically, this is represented by:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} c_{11} & c_{21} & c_{31} \\ c_{12} & c_{22} & c_{32} \\ c_{13} & c_{23} & c_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{Equ. 21.7})$$

where  $\{R, G, B\}$  is the original color triad,  $\{R', G', B'\}$  is the adjusted color triad, and the coefficients of the transform matrix are  $c_{ij}$ . A matrix multiplication takes place in your digital camera every time you make a picture.

If the values of the matrix coefficients  $c_{11}$ ,  $c_{22}$ , and  $c_{33}$  are set to 1.0, and all other  $c_{ij}$ s are set to zero, the new triad comes out identical to the original triad. But if  $c_{11}$  is greater than 1.0, the new image will be redder. The coefficient  $c_{22}$  controls green, and  $c_{33}$  controls blue. When other  $c_{ij}$ s take on non-zero values, the color channels bleed into one another, and the original RGB color cube is twisted and warped into a new R'G'B' color space. Using  $c_{ij}$  values carefully chosen by the camera manufacturer, your family pictures end up with accurate color.

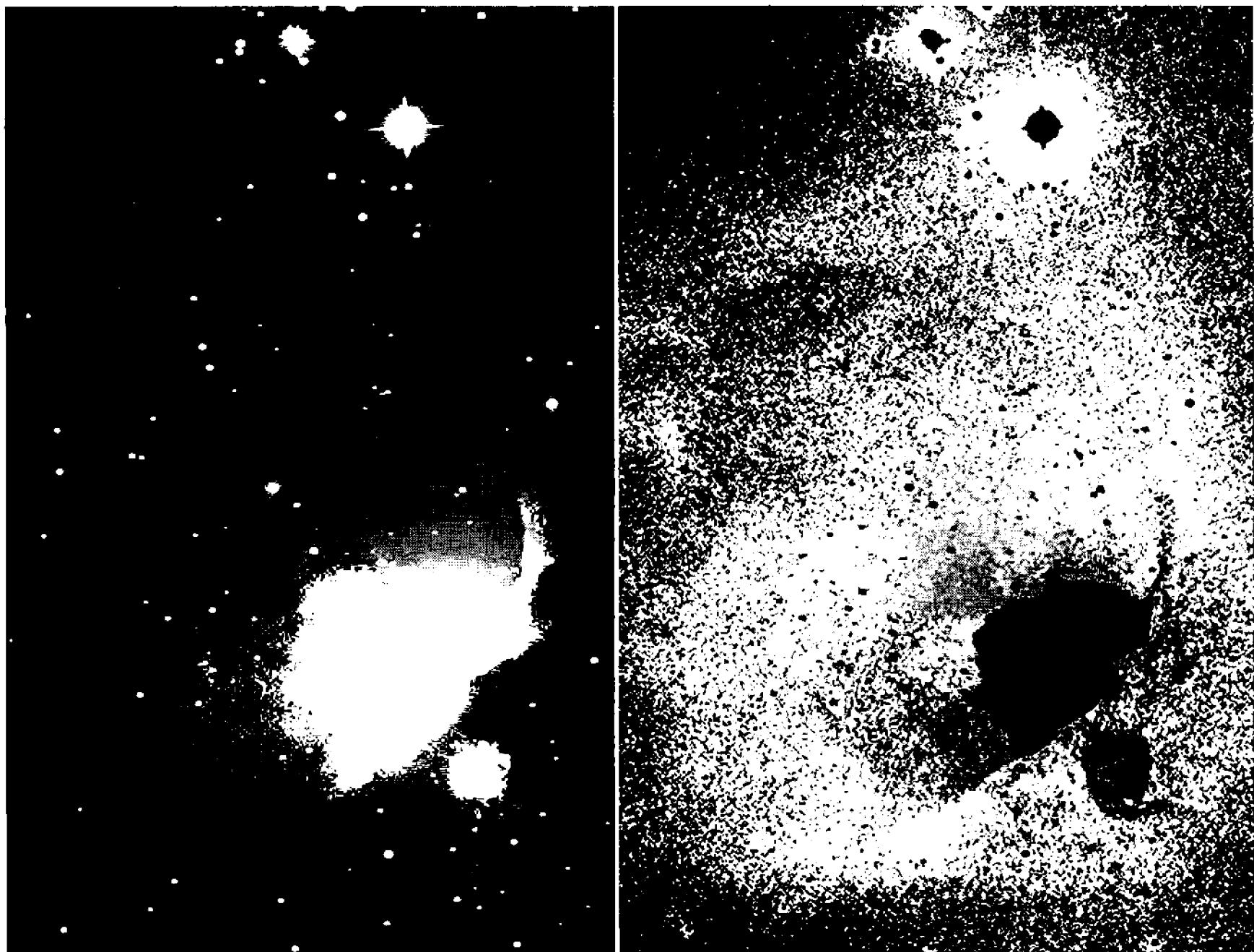
The greatest strength of RGB color space is that computers love it. Furthermore, it's easy to make digital cameras that produce images in RGB color space, and easy to display RGB output values on a computer monitor.

The big disadvantage of RGB color space is that it's difficult for *people* to use. Stretching a color channel changes many things all at once. The overall brightness of the image changes, and dark ("shadow") and light ("highlight") features can take on unwanted color casts.

- **Tip:** *AIP4Win processes images in their native RGB color space whenever offers a processing advantage or is the most appropriate color space for the task at hand. In addition, AIP4Win gives observers the ability to split an image into its red, green, and blue channels, process each individually, and then reassemble them into a new color image.*

### 21.0.1 HSL Color Space

HSL stands for Hue, Saturation, Luminance, or  $\{H, S, L\}$  color space. HSL color space contains the exact same information that RGB does, but in a different form. In RGB color space, the luminance (brightness) of the image and its chrominance



**Figure 21.11** The luminance channel (left) in HSL color space shows the image brightness, while the saturation channel (right) represents an esoteric quality of the image: the strength of its color. HSL color space is ideally suited for tasks like enhancing image contrast, image detail, and image features.

(color) are inextricably intertwined—change one and you change the other. In HSL color space, luminance and chrominance have been separated. Chrominance information is carried in the hue (color) and saturation (color strength) channels, and image brightness is carried separately in the luminance channel.

You are familiar with luminance images as “black and white.” The two components of chrominance—hue and saturation—are familiar terms, too. Hue describes the colors on the artist’s color wheel, and saturation describes the strength of the colors.

In Section 20.6.2, we discussed how software converts from the RGB to the HSL color space. Geometrically, you can think of RGB as a cube and HSL as a cylinder—but the important difference is that because humans see the world in terms of brightness and color, HSL color space mimics the human eye’s way of seeing images. Working in HSL color space, you can darken or lighten an image without affecting its color, and you can set its color balance without changing the luminance.

- **Tip:** *HSL is the native color space in AIP4Win for color images. AIP4Win accords primacy to the luminance component to accommodate the wide dynamic range found in astronomical images. Because of this, virtually every image-processing function in*

## Chapter 21: Processing Color Images

**AIP4Win** can be applied to color as well as monochrome images. When an image can best be processed in RGB color space, **AIP4Win** automatically converts the image to RGB, processes it, and returns it to HSL color space.

### 21.0.1 Lab Color Space

Like the HSL color space, Lab color space has luminance/chrominance-based color coordinates. However, in  $S$  and  $L$  axes in HSL space, color differences are not constant—sometimes a small change can make a big difference. On the  $a$  and  $b$  coordinate axes of the Lab color space, chrominance differences are uniform over the range of possible  $a$  and  $b$  values.

- **Tip:** *AIP4Win employs the Lab color space for image-processing tasks that require blending colors. For example, a noisy sky background can be averaged in Lab color space so that it appears uniform when it is returned to HSL color space. The switch from HSL to Lab and back is handled automatically.*

## 21.1 Color in Color Images

To most people, adjusting the color in an image means setting the relative strengths of its red, green, and blue channels so that it “looks right.” This definition works reasonably well for familiar daytime objects illuminated by sunlight; everyone knows how everyday things should look. In the world of astronomical imaging, however, where the human eye cannot perceive the colors of faint celestial objects, we must either base our notions of them on other images, or look for objective ways to balance the color channels in images.

### 21.1.1 White Balance

Because white balance is crucial in making acceptable images, most digital cameras come with built-in routines to handle it. If you set the white balance control to the type of scene illumination (direct sunlight, tungsten, fluorescent, cloudy, shady), you’ll get images with reasonably accurate color balance. The manufacturer has measured typical scenes in these types of light, and stored matrix coefficients in the camera’s firmware that portray a 100% reflective white object photographed under these illuminations which will appear white. In sRGB color space, “white” means that the red, green, and blue color channels will each equal 255 ADUs.

**Use the Camera White Balance.** Your digital camera’s “sunlight” or “direct sunlight” setting should yield accurate color balance for almost all astronomical images. The camera will select a set of matrix coefficients that will give a sunlike light source equal values of red, green, and blue.

**Use the Peak Values in the Image.** A fairly reliable way of getting a good-looking image is to measure the peak pixel value in each color channel, and then scale the color channels so that the peak value in each will be 255 in the finished

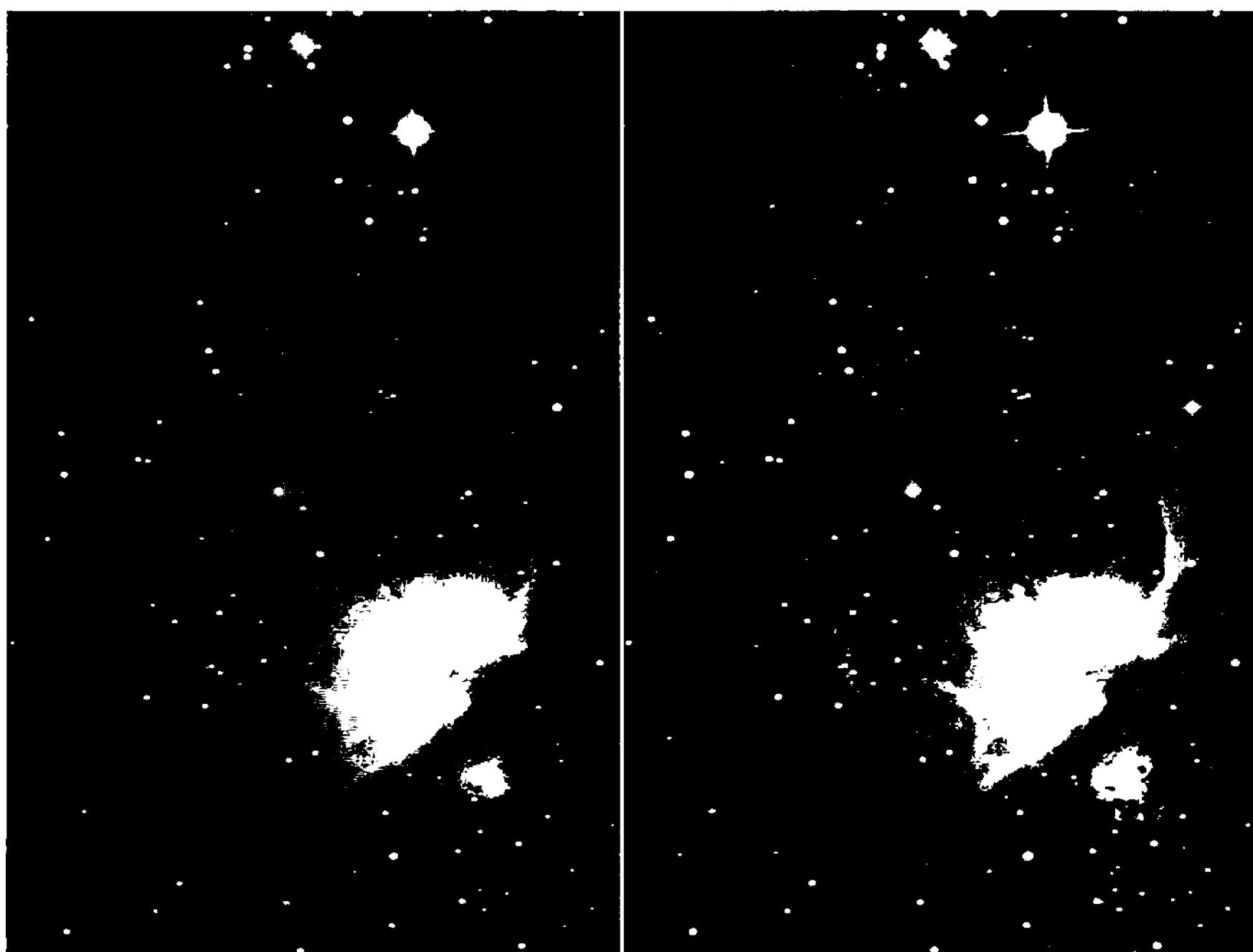


Figure 21.12 Case study: Luminance-channel processing opens color images to a wide variety of powerful image-processing functions. On left is the original image of the Orion Nebula, and on the right is same image enhanced with wavelet spatial filtering to sharpen and heighten nebular details.

image. This method works quite well for terrestrial scenes where it is reasonable to expect that the brightest object should look white. Since the orangish cast of tungsten and greenish cast of fluorescent appear in the image of a lightest-color object in the scene, this method forces it to appear white.

**Use Histogram Percentiles to Set Black and White.** In astronomical images, the peak pixels values are usually due to one or two bright stars, and the lowest pixel values are the colors of the sky background. Rather than setting the white point to the peak values, in astronomical images it is best to assume that the average color of several dozen or several hundred stars is close to white. To accomplish this, create a histogram for each color channel and set the white point to 99.5-th percentile in each color channel. Since most people want the sky background to be a very dark shade of gray, sampling a few hundred spots on a deep-sky image and finding their median value is a reliable way to determine the brightness of the sky background. The red, green, and blue color channels can each be stretched between the black and white points to give a black sky and an average star color of white.

- **Tip:** *AIP4Win's Color Image Tool includes a tab with functions that can automatically measure the histogram of a color image to find*

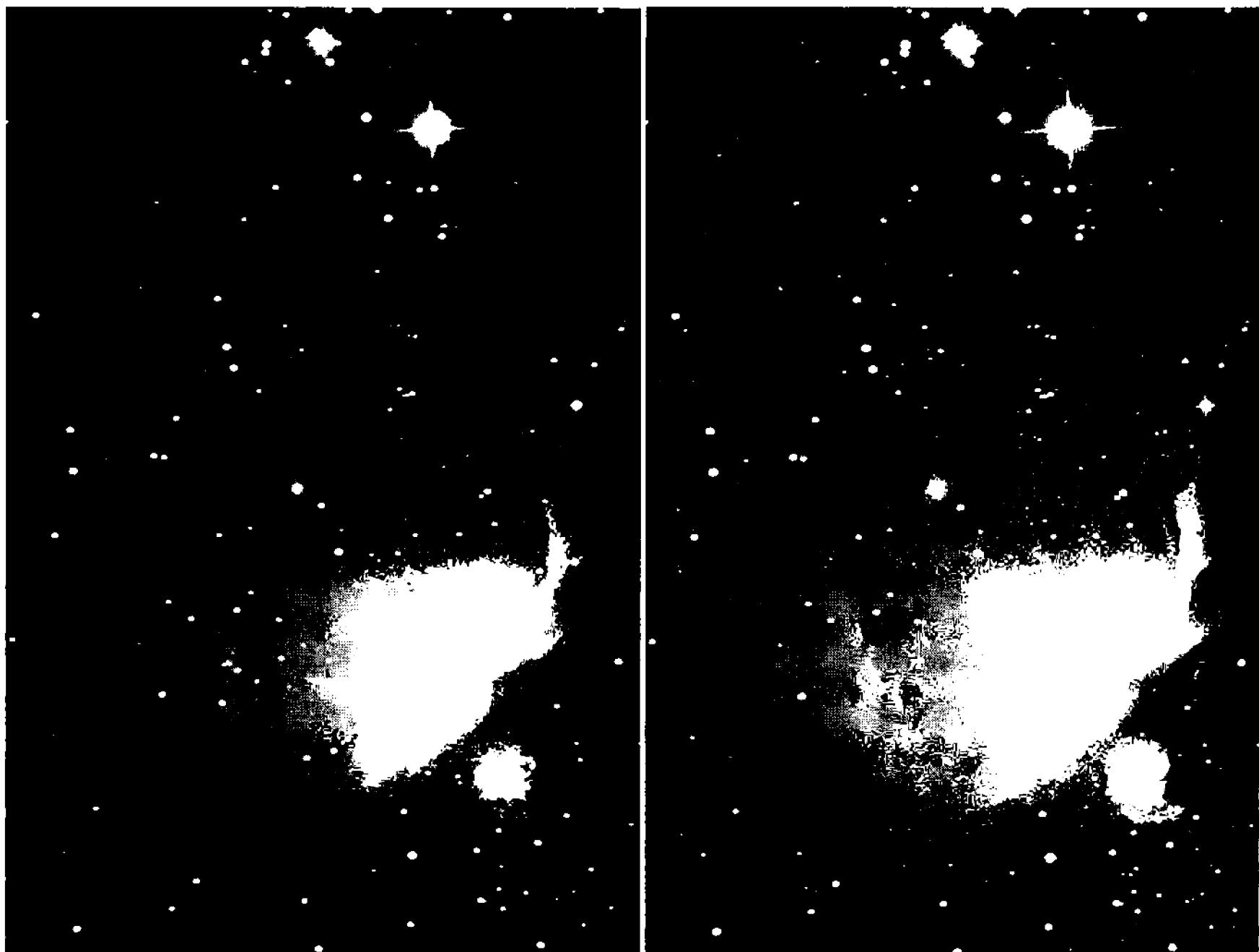


Figure 21.13 Case study: In HSL color space, you can enhance color with powerful algorithms. The left image has been sharpened from the original by 20 iterations of the Lucy-Richardson deconvolution algorithm. In the right, image luminance has been gammalog brightness scaled.

*the best value in each color channel for the sky background black point and white point.*

### 21.1 Luminance Enhancement of Color Images

Color balance is just the beginning for color image processing. In color images from digital cameras, the luminance channel always has a higher signal-to-noise ratio than chrominance channels do—enabling you to apply powerful image processing routines to the luminance channel. And because changes to the luminance channel don't alter the color balance, you can experiment freely.

**Brightness Scaling.** Important structures of galaxies and nebulae often have pixel values close to that of the sky background. Revealing these structures using simple brightness stretches usually ends up blowing out the bright parts of the galaxy or nebula. Brightness scaling offers options, such as the gammalog function, that simultaneously stretch low pixel values and compress high ones. This enables you to brighten the faint outer structures without losing the bright central arms and nucleus of your favorite galaxy.

**Histogram Shaping.** Histogram shaping is capable of converting an image with a few bright stars into one filled with swirling wreathes of interstellar gas. It

## Section 21.1: Luminance Enhancement of Color Images

forces the luminance data of an image into a histogram chosen ahead of time for superior image display properties. One of the most effective histogram shapes is exponential, a form that allows a steadily declining number of pixels into higher and higher pixel values—with the result that details from the very faintest to the very brightest appear in a single image.

**Unsharp Masking.** Unsharp masking is the classic method for revealing details in lunar and planetary images. However, if you split an image into RGB color channels, apply unsharp masking each in turn, and then combine them into a color image, color artifacts often appear around edges and bright features. By enhancing the luminance channel without touching the underlying chrominance channels, fine details appear clearly without introducing color artifacts.

**Wavelet Spatial Filtering.** Wavelet filtering has been called “unsharp masking on steroids”—but wavelets are legal! Wavelet filtering gives you the ability to selectively enhance the contrast of image details at every spatial scale, from tiny features a single pixel wide to features spanning hundreds of pixels. Figure 21.12 shows wavelet filtering on our standard Orion Nebula test image.

**Deconvolution.** Amateur astronomers use deconvolution to restore star images in images shot on nights with poor seeing. Starting with a sampled star image taken from the image itself, the Lucy-Richardson deconvolution algorithm attempts to discover the source image that most probably produced the degraded image captured by the image sensor. For sharpening images with soft star images, deconvolution is hard to beat.

**Gradient Correction.** It would be nice if the flat-fields used in calibration would correct all of the sources of nonuniformity in images, such as gradients in sky brightness, hot spots, and scattered light. However, a large class of algorithms exist that can remove these unwanted effects from digital images. The simplest gradient corrections fix images that are brighter on one side than on the other; sophisticated gradient corrections can flatten images with lumpy sky backgrounds.

**Noise Removal.** Image smoothing, noise filtering, and iterative noise removal are different approaches to making images less noisy. Image smoothing operates by averaging pixels below a threshold value; noise filtering scans an image and removes the pixels that exceed local variations; and iterative noise filtering uses wavelet analysis of noise amplitudes to remove statistically non-significant variations in pixel value. Applied to the luminance channel of a color image, these techniques give rough or grainy images a milky smoothness.

## Chapter 21: Processing Color Images

---

# Appendix A Glossary

---

Image processing has its own specialized vocabulary. In this glossary, you will find special terms defined and the abbreviations that astronomers and image processing experts use spelled out.

<b><i>à trous algorithm</i></b>	A computation scheme used to create coefficients for wavelet scales.
<b>AAVSO</b>	American Association of Variable Star Observers.
<b>aberration (astrometric)</b>	An effect in which the apparent direction of starlight changes due to the orbital motion of the Earth.
<b>aberration (optical)</b>	A characteristic of an optical system that causes light from a point source (such as distant star) to come to an imperfect focus.
<b>ABG</b>	Anti-Blooming Gate; a structure on a CCD that prevents blooming from bright sources.
<b>achromatic</b>	An optical system that brings two wavelengths of light to a common focus.
<b>ADU</b>	Analog-to-Digital Unit; a dimensionless number representing the output from the analog-to-digital converter in a CCD camera.
<b>ADC</b>	Analog-to-Digital Converter; the component in a CCD camera that converts the analog output of the CCD into numbers.
<b>air mass</b>	The thickness of the atmosphere between the telescope aperture and the target of the observation. A celestial object at the zenith is viewed through an air mass of one.

## Appendix A: Glossary

**airglow**

The natural emission from atoms and molecules in the upper layers of Earth's atmosphere, principally neutral oxygen. Airglow causes a diffuse background glow in astronomical images taken from the Earth's surface.

**Airy disk**

The bright central portion of the image of a point source of light formed at the focus of a telescope. The Airy disk in a perfect unobstructed telescope contains 84% of the light from the star; the bright disk is surrounded by a faint system of diffraction rings.

**algorithm**

A well-defined process used to achieve a desired result. In cooking, an algorithm is called a recipe.

**all-sky photometry**

Measurements of stellar magnitude made over the entire sky.

**analog-to-digital converter**

An electronic device used to convert a continuously varying voltage into a series of discrete binary numbers. Abbreviated as ADC.

**annulus**

A donut-shaped ring. In photometry, the region of sky sampled to measure the brightness of the background sky.

**anti-blooming gate**

A feature of some CCD devices that bleeds excess charge off a photosite during integration to avoid charge spill-over onto neighboring photosites.

**aperture**

In optics, the opening in a telescope through which light is admitted. In most telescopes, the aperture is the same as the primary mirror or objective lens. In photometry, the region of an image sampled to measure the brightness of a star image. The sky brightness is measured in an annulus around the aperture.

**aperture photometry**

A method of measuring star brightness that uses two regions: an aperture with the star plus background sky, and an annulus with the background sky. The contribution measured in the annulus is subtracted from that of the aperture, giving the star's brightness.

**apoachromatic**

An optical system that brings three or more wavelengths of light to a common focus.

**aspect ratio**

The width of an image or pixel divided by its height.

<b>astigmatism</b>	An aberration in optical systems that exists whenever there is a difference between the optical power of the system in the tangential and sagittal planes. Astigmatism degrades telescopic images.
<b>astrometric catalog</b>	A list containing the precise positions for reference stars for use in determining the coordinates of target stars.
<b>astrometric position</b>	The coordinates of a celestial object which have the effects of precession, nutation, aberration and parallax accounted for. It is generally formalized in a fundamental catalog of stellar positions.
<b>astrometry</b>	The science of measuring the positions of celestial objects. Because of their geometric stability, CCDs are superb tools for astrometric purposes.
<b>astronomical CCD camera</b>	A digital camera designed and optimized for astronomical imaging, with a cooled, low-noise CCD detector, and low-noise amplifier.
<b>autoguiding</b>	Use of an automated feedback system to keep a telescope pointed precisely on an object while taking an image.
<b>back-illuminated CCD</b>	A CCD which has had its back surface ground or etched to permit the passage of light. It is mounted “upside down” compared to conventional CCD devices so that incoming light does not pass through the gate structures on the front of the CCD.
<b>baffle</b>	A barrier in an optical system intended to prevent stray light from reaching the focal plane.
<b>Bayer array</b>	A pattern of color filters integrated into a CCD or CMOS detector, usually either red/green/blue or cyan/magenta/yellow/green.
<b>bias frame</b>	A CCD image with a zero-length exposure to complete darkness. In the ideal case, the bias frame contains no thermal or photonic contribution.
<b>bias voltage</b>	The voltage offset measured at a pixel in a bias frame. It is the minimum value that a pixel can have.
<b>bimodal distribution</b>	A range of values showing two prominent peaks; usually in reference to values in a histogram.
<b>binary</b>	A number system containing only zeroes and ones, used in logic circuits and digital computers.

## Appendix A: Glossary

<b>binning</b>	Reading out a CCD chip in such a way that groups of adjacent photosites are combined to create larger equivalent photosites.
<b>bit</b>	Binary digit; a zero or one value in a binary number. An 8-bit number uses 8 binary digits and represents decimal values from zero to 255.
<b>blooming</b>	Spill-over charge from a saturated CCD photosite into adjacent photosites. Blooming “tails” often extend from the images of bright stars.
<b>Cartesian coordinates</b>	A coordinate system with mutually perpendicular axes, by convention having the origin at lower left.
<b>Cassegrain</b>	A two-mirror telescope that uses a convex secondary mirror to reflect the light captured by the primary mirror back through a hole in the primary mirror.
<b>catadioptric</b>	A compound telescope that uses both mirrors and lenses to bring light to focus.
<b>CBA</b>	Center for Backyard Astrophysics, Columbia University; supports amateur observations of cataclysmic variable stars.
<b>CCD</b>	Charge-Coupled Device; an electronic light detector that converts photons to electrons in an ordered array of photosites.
<b>centroid</b>	The “center of mass” of a star image. Often gives the location of a star image to a small fraction of a pixel.
<b>charge skimming</b>	The trapping and subsequent release of signal electrons during readout by photosites containing defects or impurities.
<b>charge well</b>	The structure in a CCD’s photosite in which electrons are trapped during integration.
<b>chromatic aberration</b>	A characteristic of an optical system in which light of different colors is brought to focus at different distances from the primary.
<b>chrominance</b>	The color component of an image, characterized by its hue and saturation. Luminance is the brightness component.
<b>CMY</b>	Cyan, Magenta, Yellow; a subtractive color system used in the graphic arts, but employed as an additive color system in CCD imaging.

<b>CMY/L</b>	Cyan, Yellow, Magenta, Luminance; the CMY with the addition of a luminance channel that contains information about the image brightness. Also called CMYW.
<b>color index</b>	The difference between the photographic and visual magnitude of a star; more generally, the magnitude difference between any two photometric colors.
<b>coma</b>	An off-axis optical aberration in which the focal length varies with off-axis angle. Produces images with a flare pointing away from the optical axis.
<b>conduction band</b>	The energy level in an atom which is normally unoccupied by electrons unless a current is flowing.
<b>conversion factor</b>	The ratio between the number of electrons collected by a CCD photosite and the pixel value in ADUs; expressed in units of electrons per ADU ( $e^-/ADU$ ).
<b>Cookbook camera</b>	The do-it-yourselfer's CCD camera built from plans published in <i>The CCD Camera Cookbook</i> .
<b>coordinate reference frame</b>	A standard reference relative to which astrometric positions are determined.
<b>cosmic ray</b>	An energetic charged particle or photon from space.
<b>cross histogram</b>	A plot of the values of the pixels in one image versus the corresponding values of pixels in a second image.
<b>CRW</b>	The filename extension used for raw files made by Canon digital cameras.
<b>cylinder focusing</b>	Reflections off the sides of a tube in an optical system that cause unwanted light to be concentrated near the focal plane.
<b>dark current</b>	Electrons generated at a CCD photosite due to thermal excitation of electrons in the silicon substrate. Dark current is reduced when a CCD is cooled.
<b>dark frame</b>	A CCD image taken with no light reaching the sensor. Used in image calibration to remove the effects of dark current.
<b>declination</b>	The celestial coordinate axis corresponding to terrestrial latitude. Abbreviated as $\delta$ or Dec.
<b>deconvolution</b>	An image processing technique in which the inverse of the point-spread function of the optics and atmosphere and the optical system is applied to an image.

## Appendix A: Glossary

<b>diffraction grating</b>	An optical element with a set of very finely ruled lines. Diffraction spreads the light that passes through or reflects from a grating into a spectrum.
<b>diffraction limited</b>	A telescope or optical system that forms images nearly indistinguishable from those formed by a perfect optical system.
<b>digital camera</b>	A camera equipped with a CCD or CMOS detector designed for general-purpose photography.
<b>digital SLR camera</b>	A digital camera with a reflex mirror viewing system designed to accept interchangeable lenses.
<b>digitize</b>	To convert a measurement of an analog value to a discrete number. In CCD cameras, the current from the CCD is digitized into analog-to-digital units (ADUs).
<b>dimensionless number</b>	A value that has no corresponding units.
<b>dispersion</b>	The characteristic of a prism or diffraction grating that defines the angle through which the spectrum of light is spread. The higher the dispersion, the greater the resolution.
<b>distortion</b>	In optics, a characteristic of an optical system that causes the image formed at the focus to deviate from the normal $r \tan \vartheta$ law.
<b>DN</b>	Data Number, a unit sometimes used to express pixel value. Equivalent to ADU.
<b>drift subtraction</b>	A method used in CCD cameras, in which the bias is read from masked pixels and subtracted from each of the pixels in that row. A fixed bias value may be added to each pixel to avoid clipping negative values.
<b>dynamic range</b>	The ratio between the maximum value that a signal (such as pixel value) can achieve and the noise in the signal; sometimes expressed in decibels.
<b>electroluminescence</b>	The emission of light by a semiconductor when current is passed through it. In CCDs, the on-chip amplifier can cause a glow visible in images.
<b>electron-volt</b>	Unit of energy equal to that required to move one electron through one volt of potential.
<b>emulsion</b>	In photography, a suspension of crystals of silver halide or organic photo-sensitive dyes in gelatin, coated onto glass plates or sheets of plastic film.

<b>epoch</b>	The specific point in time to which an astrometric position is referenced.
<b>extinction</b>	The absorption of photons by the Earth's atmosphere which causes dimming of celestial objects.
<b>field flooding</b>	The illumination of the focal plane of a telescope by light that bypasses its optical system. Also called "veiling glare."
<b>field rotation</b>	The apparent rotation of an image due to the motion of the sky when a telescope's polar axis is not aligned with the Earth's polar axis.
<b>flat field</b>	A CCD image of a uniformly illuminated blank field. It is used in image calibration to remove the effects of varying pixel sensitivity and uneven illumination of the CCD chip.
<b>floating-point number</b>	A number expressed in scientific notation, as a decimal fraction times an exponent, used to express numbers that have a wide range of value. In computation, they trade numerical precision for range.
<b>focal extender</b>	Optical elements inserted ahead of focus in a telescope to extend the focal length of the system, such as a Barlow lens.
<b>focal length</b>	In a simple lens system, the distance from the objective at which light rays come to focus. In a compound optical system, the distance from a lens that forms an equivalent image.
<b>focal ratio</b>	The ratio between the focal length and the diameter of the objective; also called aperture ratio.
<b>focal reducer</b>	Optical elements inserted ahead of focus in a telescope to reduce the focal length of the system by causing the incoming rays to converge more rapidly.
<b>FOV</b>	Field Of View; the area of sky included in an image, expressed in degrees.
<b>frame-transfer CCD</b>	A CCD which has two regions, one exposed to light, the other masked. At the conclusion of an integration, the electrons in the unmasked region are rapidly shifted to the masked region from which they can be more leisurely read out. This type of CCD needs a shutter only for imaging extremely bright objects.

## Appendix A: Glossary

<b>Fraunhofer lines</b>	The prominent absorption lines in the solar spectrum first catalogued by spectroscopic pioneer Joseph von Fraunhofer around 1814.
<b>free spectral range</b>	The range of wavelengths in a spectrum; for a grating spectrum, the range of wavelengths that is free of overlapping spectral orders.
<b>frequency</b>	The rate at which a phenomenon repeats. For light, frequency is the number of wavecrests that pass a fixed point in a given time interval. For features in an image, spatial frequency is the number of dark/light cycles in a given distance.
<b>front-illuminated CCD</b>	A CCD in which the incoming light passes through the gate structure on top of the silicon substrate of the chip; most CCDs are front illuminated.
<b>full-well capacity</b>	The maximum number of electrons that the charge well in a CCD can hold before charge escapes. At full-well capacity, the CCD is said to be saturated.
<b>FWHM</b>	Full-Width at Half-Maximum; the width of a peak at half its maximum value. Can be applied to the point-spread function of a star image, or the wavelength range of optical filters.
<b>gamma</b>	A power-law transfer function which has its greatest effect on midrange pixel values.
<b>GPS</b>	Global Positioning System; a system of satellites that provides extremely accurate timing and location signals to airborne or ground-based satellite receivers.
<b>grayscale</b>	Monochrome; containing only shades of gray.
<b>guiding</b>	Maintaining telescope pointing at a star or other celestial object during image integration.
<b>Hipparcos Catalog</b>	A catalog listing astrometric positions for 118,000 stars accurate to better than 0.01 arcsecond, generated from data obtained by the Hipparcos satellite.
<b>histogram</b>	A plot showing the number of pixels in an image having a pixel value <i>versus</i> pixel value.
<b>hot pixel</b>	A CCD photosite that generates significantly higher than average dark current or thermal noise.
<b>HSL</b>	Hue, Saturation, Luminance (or lightness); a color system in which light is described in terms of its color, purity, and intensity.

<b>hue</b>	The color of light (red, orange, yellow, green, blue, violet, purple); often arranged in a “color circle.”
<b>HWHM</b>	Half-Width at Half-Maximum; half the width of a peak at half its maximum value.
<b>IAU</b>	International Astronomical Union; the official world-wide organization of astronomers.
<b>ICRS</b>	International Celestial Reference System; a standard astrometric reference frame based on long-baseline radio interferometry of cosmic radio sources.
<b>instrumental magnitude</b>	The magnitude of a star referenced only to the system used to make the measurement.
<b>integer</b>	A value expressed as a whole number; a number without a fractional part.
<b>interline-transfer CCD</b>	A CCD which has alternating columns of photosites that are masked from the light between columns that are unmasked. After an integration, the contents of the unmasked columns are all shifted to the masked columns after which they can be more leisurely read out. This type of CCD does not generally require a shutter.
<b>JPEG</b>	Acronym for Joint Photographic Experts Group, a lossy image storage format that is popular on the World Wide Web due to its ability to compress images without a great deal of information loss.
<b>Julian date</b>	The count of days since Julian day 0, which started at noon on January 1, 4713 BC; used by astronomers to simplify computation of time intervals between dates.
<b>Landolt Selected Areas</b>	A set of regions on the celestial equator, spread evenly around the sky, for which photometric reference stars (sequences) have been carefully measured.
<b>linear transform</b>	An image processing operation that can be reversed in that it has no discontinuities.
<b>linearity</b>	The characteristic of a CCD such that the number of electrons measured at a photosite is directly proportional to the number of photons falling on that photosite.
<b>lossy</b>	During image compression, a lossy algorithm loses image information. Lossless algorithm allow complete restoration with no information loss.

## Appendix A: Glossary

<b>luminance</b>	The brightness component of an image. Chrominance is the color component.
<b>magnitude</b>	The brightness of a star expressed on a logarithmic scale, with each successive magnitude 2.512 times brighter than the previous step. The magnitude increases as the object becomes fainter.
<b>mean</b>	The average of a set of values. The sum of all the values in a set divided by the number of values.
<b>median</b>	The value in a set of values for which there are as many members greater than as there are less.
<b>Messier object</b>	A deep-sky object in a list cataloged by Charles Messier, the 18th-century comet hunter. Messier apparently intended his catalog to aid other searchers, since it is a list of diffuse objects that might be mistaken for a comet.
<b>micron</b>	An informal unit of measurement corresponding to the formal unit micrometer ( $10^{-6}$ of a meter).
<b>mode</b>	The most frequently occurring value in a set of values.
<b>nanometer</b>	One billionth ( $10^{-9}$ ) of a meter. The wavelength of green light is 550 nanometers.
<b>NEF</b>	The filename extension used for raw files made by Nikon digital cameras.
<b>Newtonian</b>	A telescope configuration in which light is reflected from a parabolic primary mirror to a smaller plane diagonal secondary mirror and then to the focus.
<b>nutation</b>	A wobbling motion of the Earth as it rotates. It needs to be taken into account in establishing high-precision astrometric coordinates.
<b>Nyquist Sampling Theorem</b>	In signal processing, the proposition that to recover all the frequency information in a signal it is necessary to sample it at twice the highest frequency found in the signal. In image processing, this theory has been extended such that the smallest resolved detail must contain two samples.
<b>objective prism</b>	A prism placed in front of a telescope to spread the light of stars into its component colors.
<b>optical axis</b>	The axis of symmetry in an optical system.

<b>output amplifier</b>	An electronic circuit in a CCD camera which follows the output of the CCD chip and boosts its level to make it more easily measurable by the analog-to-digital converter.
<b>oversampled</b>	An image in which the sampling frequency exceeds the Nyquist sampling criterion; an image sampled with more than two pixels across the inner core of the diffraction disk.
<b>parallax</b>	The apparent change in position of a foreground object against the distant background due to the motion of the Earth around the Sun.
<b>periodic error</b>	Tracking errors in a telescope drive system that repeat, usually correlated with the rotation period of the worm in a worm-gear drive.
<b>photometric bit-depth</b>	The number of binary digits (bits) used to store the brightness of a star image.
<b>photoelectron</b>	In detectors such as CCDs, an electron freed by the action of a photon.
<b>photographic magnitude</b>	The magnitude of a star as measured from a blue-sensitive (orthochromatic) photographic plate.
<b>photometric curve of growth</b>	A plot of the measured magnitude of a star versus the radius of the aperture used to measure it. Used to determine the optimum radius for aperture photometry.
<b>photometry</b>	The science of measuring the brightness (magnitudes) of stars. CCDs are well-suited for making photometric measurements.
<b>photomultiplier tube</b>	A vacuum-tube detector in wide use for astronomical photometry from ~1950 to ~1995.
<b>photon</b>	The fundamental particle of light; carries an energy inversely proportional to the wavelength of the particle; i.e., short wavelengths have high energy.
<b>photovisual magnitude</b>	The magnitude of a star measured from a photograph taken through a filter transmitting light between 495 nm and 580 nm wavelength, and intended to be equivalent to the brightness seen by the human eye.
<b>pixel</b>	Short for Picture Element, a single element of a digital image.
<b>pixel coordinates</b>	The location of a pixel in an $(x, y)$ coordinate system.

## Appendix A: Glossary

<b>pixel value</b>	The numerical value of a pixel; often but not necessarily expressed in analog-to-digital units (ADUs).
<b>plate constants</b>	Coefficients of a linear transformation between the coordinates of a point on an image and the corresponding celestial coordinates.
<b>point-spread function</b>	A function describing the distribution of light in the focal plane of a telescope after the light from a point source has passed through the atmosphere and its optics.
<b>polysilicon gate structure</b>	The semitransparent electrically conductive structures built on top of the bulk silicon of a CCD chip to form the charge well that traps photoelectrons.
<b>precession</b>	A wobble in direction of the axis of rotation of the Earth (like the wobble of a child's top) that causes celestial coordinates to change with time.
<b>principal points</b>	In optical design, the locations on the axis of the entrance and exit pupils of an optical system.
<b>principal rays</b>	In optical design, light rays that pass through the principal point.
<b>progressive-scan CCD</b>	A CCD chip in which the image area is read out directly. This type of CCD typically requires a shutter to keep the image from smearing as it is clocked out of the array.
<b>proper motion</b>	The apparent motion of a star on the sky due the component of its movement through space that is across our line of sight.
<b>PSF</b>	Point-Spread Function; describes the distribution of light in the focal plane of a telescope after the light from a point source has passed through the atmosphere and the optics of a telescope; can be generalized to include the viewing or detection system.
<b>quantization</b>	The assignment of discrete values to a quantity. In a CCD camera, the ADC produces integer values when it quantizes the voltage from a CCD's output amplifier.
<b>quantum efficiency</b>	The ratio of the number of photoelectrons generated to the number of photons that fall on a detector. Often expressed as a percentage, i.e., 50%.
<b>radial profile</b>	A plot of the value of each pixel in a star image versus its distance from the centroid of the star image.

<b>random noise</b>	Noise which does not repeat from one image to another. Such noise is said to be uncorrelated.
<b>readout noise</b>	Noise (random variation) introduced during the process of reading out the stored charge in a CCD.
<b>rectilinear</b>	In optics, the mapping of points on one plane onto corresponding points on another without distortion.
<b>resample</b>	To generate a new set of samples by interpolating between pixels in an image; changes the size of an image, making it larger or smaller.
<b>residual</b>	In astrometry, the differences between the given coordinates of a reference star and the best-fitting plate constants. Small residuals imply an accurate plate solution.
<b>Ritchey-Chretien</b>	A variation on the Cassegrain telescope having a hyperboloidal figure on the primary mirror and hyperboloidal figure on the secondary mirror.
<b>RGB</b>	Red, Green, Blue; an additive color system used to capture astronomical color images.
<b>RGB/L</b>	Red, Green, Blue, Luminance; An additive color system enhanced with a luminance image, to capture high quality astronomical color images. Also called RGBW.
<b>right ascension</b>	The celestial coordinate corresponding to terrestrial longitude. Abbreviated as $\alpha$ or RA.
<b>sampling</b>	The measurement of discrete points in a signal or on an image.
<b>saturated pixel</b>	A photosite in which the full-well capacity has been reached. Additional photons do not produce an increase in the signal.
<b>saturation</b>	Reaching the full-well capacity of a photosite.
<b>scaling</b>	A general term for remapping one set of values into another. As a geometric transform, scaling maps one set of pixels onto a new set. As a point operation, brightness scaling remaps original pixel values into a new range of pixel values.
<b>Schmidt-Cassegrain</b>	A catadioptric telescope that incorporates a corrector plate with a four-order curve, a primary mirror, and a secondary mirror.

## Appendix A: Glossary

<b>semiconductor capacitor</b>	A capacitor implemented on a semiconductor chip using a gate oxide as an insulator between a doped semiconductor region and a metal or polysilicon gate layer.
<b>signal</b>	The quantity to be measured. In a CCD image, the signal is the charge sensed by the output amplifier, which is proportional to the total number of electrons generated at each photosite.
<b>signal-to-noise ratio</b>	The ratio of the signal present in a measurement to the random variation in the signal over time (i.e., the noise). There is no way, with a single measurement, to measure the amount of noise.
<b>slit</b>	In spectroscopy, the narrow opening through which the light of a star plus its adjacent sky background is admitted to a spectrograph.
<b>SNR</b>	Signal-to-Noise Ratio; the ratio of the signal present in a measurement to the random variation in the signal over time (i.e., the noise).
<b>spectral resolution</b>	The smallest difference in wavelength that can be distinguished in a spectrum.
<b>spectral sensitivity</b>	The response of a detector to light of different wavelengths.
<b>spectrograph</b>	An optical instrument for recording the spectrum of an object as an image on film or with a CCD.
<b>spectroscope</b>	An optical instrument for viewing the spectrum of an object by eye.
<b>spectroscopy</b>	The science of determining the chemical and physical properties of celestial objects using the light emitted, reflected or absorbed by them. Electronic cameras can be attached to a spectrograph for making spectroscopic measurements.
<b>spectrum</b>	The range of wavelengths of light emitted or absorbed by an object. Emitted light forms an emission spectrum. Specific dark lines in the spectrum of light passed through a gas form an absorption spectrum.
<b>speed of light</b>	The speed at which light travels; 186,282 miles/second or $2.9979 \times 10^8$ meters/second in a vacuum.
<b>spherical aberration</b>	An aberration of an optical system that occurs when light from different ray heights at the pupil comes to focus at different distances on the optical axis.

<b>spherochromatism</b>	An aberration of refracting optical systems in which the amount of spherical aberration depends on the wavelength of the light.
<b>stacking</b>	The addition of two or more images; performed to improve the signal-to-noise ratio, and hence the quality of the image.
<b>standard deviation</b>	A measurement of the range of differences in a set of values. It is the square root of the average of the squares of the differences of each of the values from the average value.
<b>stellar magnitude</b>	The brightness of stars expressed in the magnitude system. Magnitudes are determined by comparing the flux from a star of known magnitude to the flux from a standard star.
<b>STScI</b>	Space Telescope Science Institute; the organization responsible to NASA for operating the Hubble Space Telescope.
<b>tele-extender</b>	A device inserted in the optical path to extend the focal length of a system by diverging the incoming rays, such as a Barlow lens.
<b>thermal frame</b>	A CCD image taken with the aperture covered, and from which the bias has been removed. Used in image calibration to remove the effects of dark current when doing dark matching or dark scaling.
<b>TIFF</b>	Acronym for Tagged Interchange File Format, an image format popular with desktop publishing and graphic arts programs.
<b>transformation</b>	In photometry, the mathematical process of converting instrumental magnitudes to standard magnitudes.
<b>track-and-stack</b>	An imaging method in which multiple images are registered ( <i>i.e.</i> , tracked) and then summed (stacked).
<b>Tycho Catalog</b>	A precision astrometric catalog of roughly one million stars generated from data taken by the Hipparcos satellite.
<b>UBV</b>	Ultraviolet, Blue, Visual; the Johnson filter system used in astronomical photometry.
<b>UBV(RI)</b>	Ultraviolet, Blue, Visual, Red, Infrared; the extended Johnson filter system used in astronomical photometry.

## Appendix A: Glossary

<b>UCAC</b>	USNO CCD Astrometric Catalog; the Naval Observatory's most accurate astrometric catalog.
<b>undersampled</b>	An image that is coarsely sampled; technically, one that is sampled at less than the Nyquist criterion.
<b>USNO</b>	United States Naval Observatory.
<b>valence band</b>	The highest energy band in an atom that is normally occupied by electrons.
<b>valence shell</b>	The orbital shell corresponding to electrons in the valence band.
<b>vernal equinox</b>	The point on the ecliptic where the Sun crosses the celestial equator moving north; the first day of spring in the northern hemisphere.
<b>vignette</b>	Shadowing of portions of the optical field due to obstructions in the optical system.
<b>wavelength</b>	The distance between the corresponding points (such as the peaks) in a wave; in optics, the distance between corresponding points of light waves. The wavelength of visible light ranges from 380 nm (violet) to 720 nm (red).
<b>wavelet</b>	A localized function with an integrated value of zero; used to extract frequency information from digital images.
<b>zero point</b>	The reference point for an instrumental magnitude. It is set through comparison with a standard star in order to convert an instrumental magnitude into a stellar magnitude.

---

# Appendix B Resources

---

Astronomical image measurement and processing is well covered in college and university libraries. Most of these works can be obtained via interlibrary loan through your local library. The Internet offers another rich trove of information. However, because web addresses do change, we have included just a few links here. For current Internet links, go to the **AIP4Win** page on Willmann-Bell's web site at <http://www.willbell.com>.

## B.1 CCD Imaging

### B.1.1 CCD Imaging Books

Berry, Richard. *Choosing and Using a CCD Camera*. Richmond, VA: Willmann-Bell, Inc., 1992. This book's subtitle, "A practical guide to getting maximum performance from your CCD camera," describes its mission well. Written as an introductory guide for amateur astronomers, it presents topics such as field of view, calibration, integration time, guiding, tracking, and image processing in a digestible format.

Berry, Richard, Veikko Kanto, and John Munger. *The CCD Camera Cookbook*. Richmond, VA: Willmann-Bell, Inc., 1994. This how-to book describes the construction of CCD cameras based on Texas Instruments TC211 and TC245, for amateur astronomers with minimal experience with electronics. Includes test software for construction and software to operate the completed camera.

Buil, Christian. *CCD Astronomy: Construction and Use of an Astronomical CCD Camera*. Richmond, VA: Willmann-Bell, Inc., 1991. (Translation and revision of *Astronomie CCD*, from the French.) This remarkable book was years ahead of its time, and is responsible for bringing CCD imaging to the world community of amateur astronomers. Contains detailed descriptions of CCDs and accompanying electronics, software, and display devices, as well as many images.

Holst, Gerald C. *CCD Arrays, Camera, and Displays*. Bellingham, WA: SPIE Optical Engineering Press, 1998. An excellent general introduction

## Appendix B: Resources

to using CCDs as imaging devices for television and machine vision. Detailed sections on colorimetry, radiometry, array performance, camera performance, sampling theory, and image quality. Excellent all-around background for the engineer.

Howell, Steve B. *Astronomical CCD Observing and Reduction Techniques*. San Francisco: Astronomical Society of the Pacific Conference Series, vol. 23, 1992. A series of papers by professional astronomers about using CCDs for observations. Fine background material for amateur astronomers and students contemplating the use of a CCD camera for astrometry or photometry.

Howell, Steve B., *Handbook of CCD Astronomy*. Cambridge: Cambridge University Press, 2000. An excellent resource for amateurs interested in scientific imaging with CCDs. Although photometry is the primary focus, astrometry and spectroscopy are also covered, and a section on CCDs in space observatories rounds out the book.

Jacoby, George H. *CCDs in Astronomy*. San Francisco: Astronomical Society of the Pacific Conference Series, vol. 8, 1990. In the intervening years, CCDs have improved greatly, but this volume of papers presented at a conference of professional astronomers in 1989 is still useful background for students and amateurs who want to use CCDs for scientific observations.

Martinez, Patrick, and Alain Klotz. *A Practical Guide to CCD Astronomy*. Cambridge: Cambridge University press, 1997. Describes how CCD cameras work and what factors determine their performance. Lots of practical tips on using CCD cameras designed for amateurs.

McLean, Ian. *Electronic Imaging in Astronomy, Detectors and Instrumentation*. Chichester: John Wiley and Sons, 1997. Although it was written by a professional astronomer for professional astronomers and graduate students, this book will interest students at all levels as well as amateurs who are serious about astronomical imaging. Comprehensive discussion of topics from the physics of CCD detectors through telescopes and instrumentation to computers and image processing.

Ratledge, David. *The Art and Science of CCD Astronomy*. London: Springer, 1997. This book is a collection of articles by well-known observers, covering topics that range from solar imaging to high-resolution planetary imaging, to observing form the city and overcoming light pollution.

## B.2 Astronomical Optics

### B.2.1 Telescope Design

Rutten, Harrie, and Martin van Venrooij. *Telescope Optics: A Comprehensive*

*Manual for Amateur Astronomers*. Richmond, VA: Willmann-Bell, Inc., 1988. For amateur astronomers, there is probably no better volume about telescope optics. Rutten and van Venrooij discuss the design and performance of refracting, reflecting, and catadioptric telescopes, as well as eyepieces, field correctors, focal extenders and reducers.

Schroeder, Daniel J. *Astronomical Optics*. San Diego: Academic Press, 1987. This book explores the design and performance of telescopes with an eye toward the (at the time) forthcoming launch of the Hubble Space Telescope. It is a valuable reference for amateur telescope makers in search of high performance from a moderate size telescope.

Smith, Gregory Hallock. *Practical Computer-Aided Lens Design*. Richmond, VA: Willmann-Bell, Inc., 1998. Although focused on lens design, this book has lengthy chapters on Cassegrain and Schmidt telescopes, and a comprehensive discussion of the point-spread function in the presence of central obstructions, coma, and astigmatism that amateur telescope makers and CCD imagers will find valuable.

Wilson, Raymond N. *Reflecting Telescope Optics*. 2 vols. Berlin: Springer, 1996. These two volumes provide a comprehensive look at the thinking behind the designs embodied in today's large professional telescopes. These two volumes cover telescope design in the context of its historical evolution.

## B.2.2 Telescopes and Vision

Rose, Albert. *Vision, Human and Electronic*. New York: Plenum, 1973. Although it was written long before CCDs were invented, this book sheds light on the physical processes behind human vision and electronic image capture. The text is outstandingly clear in exploring the relationships between signal-to-noise ratio, image contrast, and resolution. It is a book that will never be out of date.

Suiter, Richard Harold. *Star Testing Astronomical Telescopes: A Manual for Optical Evaluation and Adjustment*. Richmond, VA: Willmann-Bell, Inc., 1994. "I'm going to tell you a little-known fact. Telescopes are easy to test...." Thus begins Suiter's treatise on testing the optical performance of amateur telescopes. Every observer (and every CCD imager) should own a copy of this book.

## B.3 Astrometry

### B.3.1 Astrometry Books and Articles

"Guide to Minor Body Astrometry," Internet article at <http://cfa-www.harvard.edu/iau/info/astrometry.html>. Description of how to carry out

## Appendix B: Resources

measurements and report observations of asteroids: “This guide is intended for those observers interested in undertaking an astrometric CCD-observing program of minor planets and/or comets.”

Everhart, Edgar. “Constructing a Measuring Engine.” *Sky and Telescope* (September 1982): 279. Describes building a two-axis precision engine for measuring positions from photographs; great insight into state of the art of amateur astrometry *before* CCDs.

Konig, A. “Astrometry with Astrographs.” In *Astronomical Techniques*. Ed. W.A. Hiltner. Chicago: University of Chicago Press, 1962. Quite possibly the clearest exposition of astrometry available; appears in a short chapter of an impeccable source book on astronomy.

Marché, Jordan D., II. “Measuring Positions on a Photograph.” *Sky and Telescope* (July 1990): 71. Instructions for doing astrometry from photographic prints; includes program in line-numbered Basic for reducing ( $x, y$ ) data using the plate-constants method.

Marsden, Brian. “How to Reduce Plate Measurements.” *Sky and Telescope* (September 1982): 284. A one-page article outlining the equations necessary to solve for plate constants and positions.

Montenbruck, Oliver, and Thomas Pfleger. “Astrometry.” In *Astronomy on the Personal Computer*. New York: Springer-Verlag, 1991. An exceptionally clear exposition of astrometry with an accompanying program in Pascal for performing astrometric reductions.

Podobed, V.V. *Fundamental Astrometry*. Chicago: University of Chicago Press, 1964. Despite hideous typesetting and muddy photographs, this is an excellent book for would-be astrometrists. It covers all aspects of the field (circa 1960) including meridian circle work and the photographic zenith tube.

Taff, Laurence G. *Computational Spherical Astronomy*. New York: Wiley-Interscience, 1981. This work addresses techniques for astrometry on rapidly moving objects and images captured on multiple plates—fast-moving asteroids and comets.

Van de Kamp, Peter. *Principles of Astrometry*. San Francisco: W.H. Freeman, 1967. Written by one of the foremost astrometrists, this classic work focuses on astrometry with long-focus refractors. The methods described are readily applicable to modern CCD astrometry.

### B.3.2 Astrometry Software

*Astrometrika*. By Herbert Raab. DOS/PC-based, shareware. Performs astrometry on CCD images. Info: <http://mars.planet.co.at/astrometrika/astrometrika.html>.

*Computer Aided Astronomy*. By John E. Rogers. Windows/PC-based, software distributed on CD-ROM. (Rogers also offers shareware *CCD*

*Astrometry*, DOS/PC-based.) Info: <http://ourworld.compuserve.com/homepages/johnerogers>.

### B.3.3 Astrometry Reference Catalogs

Bonanno, Emil. *Megastar*. Richmond, VA: Willmann-Bell, Inc., 1995. This software package includes the 15-million-star *Hubble Guide Star Catalog* on a CD-ROM.

UCAC. USNO CCD Astrometric Catalog. Catalog of some 60 million reference stars from 10th to 16th magnitude, based on CCD images made with a specially designed astrometric telescope. Precision of 0.020 arcseconds r.m.s. from 10th to 14th magnitude, 0.070 arcseconds r.m.s. at the limit of 16th magnitude. For current availability, see <http://www.nofs.navy.mil>.

USNO-A2.0 / USNO-SA2.0. Catalog of 526 million reference stars using Hipparcos-aided solutions to the original scans of POSS plates. Data for selected regions downloadable from the Internet. SA2.0: Selected catalog of 54 million stars on CD-ROM; for current availability, see <http://www.nofs.navy.mil>.

## B.4 Photometry

### B.4.1 Photometry Books and Articles

Budding, Edwin. *An Introduction to Astronomical Photometry*. Cambridge: Cambridge University Press, 1993. A general introduction to photoelectric photometry intended for graduate students and advanced amateur astronomers. Includes useful discussions on the photometry of extended objects and reduction of variable-star light curves.

Ghedini, Silvano. *Software for Photometric Astronomy*. Richmond, VA: Willmann-Bell, Inc., 1982. Once you have extracted photometric data from your images, you need to reduce the data. Although Ghedini presents his algorithms in line-numbered Basic, he provides enough theory to permit recoding reductions in Excel. Includes everything from time-of-minimum algorithms to Fourier transform period analysis.

Hardie, Robert A. “Photoelectric Reductions.” In *Astronomical Techniques*, Ed. W.A. Hiltner. Chicago: University of Chicago Press, 1962. A fundamental reference that should be familiar to anyone contemplating serious photometric observation. As a fascinating footnote in the history of astronomy, the chapter following this one describes machine language programming photoelectric reductions on the IBM 650 Drum Data Processing Machine.

Hall, Douglas S., and Russell M. Genet. *Photoelectric Photometry of Variable*

## Appendix B: Resources

*Stars*. Richmond, VA: Willmann-Bell, Inc., 1988. An excellent addition to the literature of photometry for amateur astronomers; particularly useful for tips on observing methods.

Henden, Arne A., and Ronald H. Kaitchuck. *Astronomical Photometry*. Richmond, VA: Willmann-Bell, Inc., 1990. Written for amateur astronomers engaged in photoelectric photometry, this book gives lucid explanations and examples of data taking and reduction. Includes indispensable tables of UBV standard stars.

Sterken, C., and C. Jascheck. *Light Curves of Variable Stars: a Pictorial Atlas*. Cambridge: Cambridge University Press, 1996. This comprehensive volume is a must for any amateur astronomer seriously interested in variable stars, providing some 200 light curves and a detailed list of references.

Wood, Frank B., ed. *Photoelectric Photometry for Amateurs*. New York: Macmillan, 1963. A classic in the field; inspiration to technically minded amateurs in the 60s and 70s. Well worth reading for historical perspective.

### B.4.2 Photometric Data: Extinction Stars and Standard Stars

Henden, Arne A., and Ronald H. Kaitchuck. *Astronomical Photometry*. Richmond, VA: Willmann-Bell, 1990. Contains lists of extinction stars for the Northern Hemisphere, matched pairs of type A0 stars for determining second-order extinction coefficients, a list of the Johnson standard stars, and a list of 223 Landolt equatorial standards in UBV(RI). Lists are also available on computer diskette.

Landolt, Arlo U. “UBV Photoelectric Sequences in the Celestial Equatorial Selected Area 92-115,” *The Astronomical Journal* 78, no. 9 (November 1973). A list of 642 standard stars established as photometric standards. The bulk of the stars are between 10.5 and 12.5 magnitude in V, and nearly all lie within one degree of the celestial equator, so they are equally accessible from the Northern and Southern Hemispheres.

Skiff, Brian. “LONEOS.STD.” Text file on the Lowell Observatory World Wide Web server at <http://ftp.lowell.edu/pub/starcats/loneos.std>. A continuously updated list with over 20,000 “pretty good” standard stars distributed over the entire sky.

### B.4.3 Photometry Organizations

American Association of Variable Star Observers. 25 Birch Street, Cambridge, MA 02138. (617) 354-0484. Director: Arne A. Henden. This membership organization provides many informational benefits for amateur observers and professional astronomers. Info: <http://>

[www.aavso.org](http://www.aavso.org).

#### B.4.4 Photometric Filters

Custom Scientific. 3852 North 15th Avenue, Phoenix, Arizona 85015. (602) 200-9200. Sells filters commonly used by research astronomers on a limited budget, by astronomy teachers, and by serious amateur astronomers. Most items in stock or available with a short delivery time. [www.customscientific.com](http://www.customscientific.com).

Optec, Inc. 199 Smith Street, Lowell, MI 49331. (616) 897-9351. (Prop: Gerald Persha. “Optec now offers a wide selection of high quality filters for astronomy. Photometric filters are available in sizes from ½-inch to 50-mm.” Info: <http://www.optecinc.com>.

Murnahan Instruments. 1781 Primrose Lane, West Palm Beach, FL 33414. (561) 795-2201. “UBV(RI) filters. Filter wheel holds 8 filters. Equipment for all CCDs.” Info: <http://www.murni.com>.

Schüler Astro-Imaging. PO Box 307, Sudbury, MA 01776. (978) 443-2037. (Prop: Chet Schüler) “Filters Designed Specifically for CCD Imaging. UBV(RI) and BVRI color filters.” Info: [chetschu@concentric.com](mailto:chetschu@concentric.com).

#### B.4.5 Photometry Software

*CCDIR*. Unified Software Systems, PO Box 23875, Flagstaff, AZ 86002-3875. (520) 774-8629. (Prop: Arne Henden) Software for reducing CCD and photoelectric photometry. Info: <http://www.flagstaff.az.us/~ccdir/> This site also contains lots of useful information about CCD photometry.

*EZPhot*. Automated CCD photometry reduction software. Available on the Web as shareware. Info: <http://www.mtco.com/~jgunn>.

*H&K*. Also called “Software for Photometric Astronomy,” Arne Hendon and Ronald Kaitchuck, Willmann-Bell, inc., 1992. DOS software for reducing photoelectric measurements (but it also works with data extracted from CCD images). Info: <http://www.willbell.com>.

*MUNIDOS*. An implementation of a professional data extraction program for Windows and Linux. Available on the Web as shareware. Info: <http://physics.muni.cz/~rudolfn>.

### B.5 Spectroscopy

#### B.5.1 Spectroscopy Books and Articles

Hafner, Reinholt. “Fundamentals of Spectral Analysis.” In *Compendium of Practical Astronomy*. Vol. 1. Ed. Gunter Roth. Berlin: Springer-Ver-

## Appendix B: Resources

lag, 1989. This 30-page chapter is a fine mid-level introduction to astronomical spectroscopy and instrumentation.

Jaschek, Carlos, and Mercedes Jaschek. *The Classification of Stars*. Cambridge: Cambridge University Press, 1990. This book is devoted to classifying stars from their spectra. After a general introduction, each principal stellar type is covered in a chapter of its own. For the would-be classifier, the spectrum profiles are invaluable.

Kaler, James B. *Stars and Their Spectra*. Cambridge: Cambridge University Press, 1989. This book-length discussion of stellar spectra is aimed at amateur astronomers and students of astronomy. By placing each stellar type in the overall perspective of the HR diagram, Kaler serves his readers well. Although it is fairly technical by the standards of amateur astronomy, the book is well illustrated and quite readable.

Kitchen, C.R. *Astrophysical Techniques*. Bristol: Adam Hilger, 1984. The chapter on spectroscopy deals primarily with spectrometers and spectroscopic instrumentation.

Sorenson, Brent. “An Objective Prism Spectrograph.” *Sky & Telescope*. (May 1983): 460. Describes a simple photographic objective prism spectrograph using a war-surplus 45° prism.

### B.5.2 Spectroscopic Catalogs

Jacoby, George H., and Deidre Hunter. “A Library of Stellar Spectra,” *ApJ Supplement Series* 56 (October 1984): 257–281. Spectra for 161 stars of all spectral types and luminosity classes. Each spectrum extends from 351 to 743 nm wavelength and is radiometrically calibrated. These data are available from the AURA/NOAO web site.

Kaler, James B. “A Catalog of Relative Emission Line Intensities Observed in Planetary and Diffuse Nebulae.” *ApJ Supplement Series* 31 (August 1976): 517–688. This catalog is a compilation of spectral line strengths for 600 nebulae. These data give considerable insight into the true colors of the nebulae for anyone willing to synthesize color coordinates from spectral data.

## B.6 Image Processing

### B.6.1 Image Processing Books: General

Baxes, Gregory A. *Digital Image Processing: Principles and Applications*. New York: John Wiley & Sons, 1994. An easy-to-read, largely non-mathematical introduction to image processing, with lots of examples and pictures. Includes a diskette with simple demonstration program.

## Section B.6: Image Processing

Castleman, Kenneth R. *Digital Image Processing*. Upper Saddle River, NJ: Prentice Hall, 1996. Intended as a textbook, this book offers a comprehensive view of image processing. The discussions of the histogram, point and linear operators, and filtering in the Fourier domain are especially comprehensible.

Gonzales, Rafael, and Paul Wintz. *Digital Image Processing*. 2nd ed. Reading, MA: Addison-Wesley, 1987. This volume is an update of the classic work in image processing, usually referred to by the authors' names alone: Wintz and Gonzales. The text moves rapidly through the basics to focus on processing and enhancement in frequency space.

Jahne, Bernd. *Digital Image Processing*. Berlin: Springer-Verlag, 1991. Every author brings a fresh viewpoint to a subject, even when the books have exactly the same title. The best nuggets to be gleaned from Jahne's book are in the area of linear shift-invariant operators and filter design.

Pratt, William K. *Digital Image Processing*. New York: Wiley-Interscience, 1978. This book grew from course notes by Dr. Pratt under the aegis of ARPA. It has the completeness characteristic of a survey intended for engineers and scientists eager to bring their skills up to date, and the clarity and directness not always found in books that deal heavily in the mathematical theory behind image processing. A classic.

### B.6.2 Image Processing: Science Applications

Green, William B. *Digital Image Processing: A Systems Approach*. 2nd ed. New York: Van Nostrand Reinhold, 1989. The focus of this book is image processing as applied to spacecraft imaging. Includes examples from Landsat, Voyager, and Viking. The discussions of computer hardware are long out of date.

Jahne, Bernd. *Practical Handbook on Image Processing for Scientific Applications*. Boca Raton: CRC Press, 1997. Focuses on image acquisition and analysis for scientific research. Full of useful hints for programmers.

Russ, John C. *The Image Processing Handbook*. 2nd. ed. Boca Raton: CRC Press, 1994. Probably the most comprehensive non-mathematical work on image processing in science applications in fields like materials science and biology, but an excellent source of ideas for astronomy. Heavily illustrated.

Sanchez, Julio, and Maria P. Canton. *Space Image Processing*. Boca Raton: CRC Press, 1998. This book is about the acquisition and processing of images from spacecraft, with a primary focus on sensors operating from Earth orbit. Among the topics covered are geodesy, multispectral analysis, and data formats. Includes program code in C++ and a CD-ROM with source code.

## Appendix B: Resources

### B.6.3 Image Processing: Image Restoration

Bates, R.H.T., and M.J. McDonald. *Image Restoration and Reconstruction*. Oxford: Clarendon Press, 1986. A solidly mathematical treatment, now somewhat dated by advances in computer speed.

Bertero, Mario, and Patrizia Boccacci. *Introduction of Inverse Problems in Imaging*. Bristol: Institute of Physics Publishing, 1998. An excellent and generally readable technical treatment of this complex subject. Includes examples of image restoration from the degraded Hubble Space Telescope.

Bijaoui, Albert. *Image Et Information: Introduction au Traitement Numerique des Images*. Paris: Masson, 1981. An early work on image restoration, with a typewritten text, in French, focusing on the restoration of images captured photographically with the Lallemande's image tube. An interesting perspective on pioneering work in image restoration.

Hanish, Bob. *Restoration, Newsletter of STScI's Image Restoration Project*, Summer 1993, Number 1. Baltimore: Science Computing and Research Support Division, STScI, 1993. An exceptionally valuable document covering the progress on restoring images made with the degraded optical system in the Hubble Space Telescope. Details of enhancements of the Richardson-Lucy iteration used to recover HST images, and much more.

Lagendijk, Reginald L., and Jan Biemond. *Iterative Identification and Restoration of Images*. Boston: Kluwer Academic Publishers, 1991. This book is especially useful for its discussion of the van Cittert iteration, with a detailed analysis of the converge of the van Cittert algorithm.

Magain, P., et al. “Deconvolution with Correct Sampling.” Internet publication at <http://vela.astro.ulg.ac.be/themes/deconv/engl/deconv.html>. The key point in this paper is that images must be properly sampled in order to recover the data present at the focal plane of the telescope.

Meinel, E.S., “Image Restoration Made Simple.” In SPIE Vol 627, *Instrumentation in Astronomy VI*, 1986. A concise and outstanding overview of image restoration methods; true to the promise of its title, it does make image restoration seem simple.

Puetter, Richard. “The Pixon Method: A New Way of Looking at Data,” Internet publication at <http://www.pixon.com>. Promotional materials for a commercial image restoration technology. Public domain Pixon code for private scientific use is available.

Stark, Henry, ed. *Image Recovery: Theory and Application*. Orlando: Academic Press, 1987. Focuses on the mathematical theory of image restoration, with a detailed chapter on the maximum entropy method of image recovery.

#### B.6.4 Image Processing and the Fourier Transform

Bracewell, Ron. *The Fourier Transform and its Applications*. New York: McGraw-Hill, 1965. This is the classic work on the Fourier transform. It combines clear exposition with well drawn diagrams. Although not directly relevant to astronomical imaging, good background for anyone who needs to understand Fourier space.

Brigham, E. Oran. *The Fast Fourier Transform*. Englewood Cliffs, NJ: Prentice-Hall, 1974. Republished in 1988. Written shortly after the introduction of the FFT, this book strives to make both the FT and the FFT comprehensible. Includes FFT source code in Fortran and Algol 60.

Ramirez, Robert W. *The FFT: Fundamentals and Concepts*. Englewood Cliffs, NJ: Prentice-Hall, 1985. Written for electrical engineers and other science-technical workers who need to understand the FFT, this book is a model of clarity in explaining the one-dimensional FFT for signal analysis. Includes Basic source code.

Zonst, Anders E. *Understanding the FFT*. Titusville, FL: Citrus Press, 1995. This nifty little book was written for the technicians, engineers, and computer programmers who want to understand the FFT in a truly hands-on way. Develops “program-it-yourself” software for the FFT as it progresses.

———. *Understanding FFT Applications*. Titusville, FL: Citrus Press, 1997. In the sequel to *Understanding the FFT*, Zonst develops application software (in line-numbered Basic) to illustrate spectrum analysis, audio fidelity, and simple image applications. The software is available on diskette.

#### B.6.5 Image Processing and Wavelets

Starck, Jean-Luc, et al., *Image Processing and Data Analysis: the Multispectral Approach*. Cambridge, Cambridge University Press, 1998; reprinted in 2000. An excellent and practical introduction to wavelet theory and image processing using wavelets.

Starck, Jean-Luc, and Fionn Murtagh. *Astronomical Image and Data Analysis*. New York, Springer-Verlag, 2002. This book, Starck’s second on wavelet theory for image and data analysis, is strikingly similar to his first. Nevertheless, enough new insights and clarifications emerge from cross-reading the two books to justify owning both.

#### B.6.6 Image Processing Algorithms

Kabir, Ihtisham. *High Performance Computer Imaging*. Greenwich, CT: Manning, 1996. Covers all basic topics in image processing from the standpoint of a programmer looking for efficient ways to code them.

## Appendix B: Resources

Especially good coverage of color spaces.

- Klette, Richard, and Piero Zamoeroni. *Handbook of Image Processing Operators*. Chichester: John Wiley and Sons, 1996. Describes dozens of operators, with algorithms and sample code in Pascal. Well worth reading for the practical programmer.
- Pavlidis, Theo. *Algorithms for Graphics and Image Processing*. Rockville MD: Computer Science Press, 1982. A fascinating look at image processing from a pioneer in the field; includes instruction for printing gray scale images by multiple character strikes on a line printer! Solid material despite its age.

Schwaderer, W. David. *Digital Imaging in C and the World Wide Web*. Wordware Publishing, 1998. Covers a broad range of basic image processing, with a good section on color models. Includes a chapter on the social issues raised by altering digital images. CD-ROM includes C source code.

Stephens, Rod. *Visual Basic Graphics Programming*. New York: Wiley Computer Publishing, 1997. The key to processing and displaying images using Microsoft's Visual Basic programming language, especially for limited computer graphics cards with 256-color palette capabilities. CD-ROM includes source code.

### B.6.7 Image Processing Software

**AIP4Win 2.0.** By James Burnell and Richard Berry. Windows/PC-based. Your copy of **AIP4Win 2.0** is included on the CD-ROM bound into this book. **AIP4Win 2.0** is a versatile and powerful package of software tools for calibrating, analyzing, enhancing, compositing, and saving monochromatic and color images from CCD cameras, scanned astrophotography, and the Internet. Updates available to registered users. Info: <http://www.willbell.com/aip4win/aip.htm>.

**Astroart.** By Marino Nicolini. Windows/PC-compatible processor required; features a wide range of image-acquisition plug-ins. Info: <http://www.sira.it/msb/astroart.html>.

**CCDSofT.** By Software Bisque (Tom and Steve Bisque). Windows/PC-based, commercial. “*CCDSofT* is an astronomical image processing and CCD image acquisition program that supports the full line of SBIG CCD camera and Apogee Instruments cameras. Written and tested by experienced astronomers, *CCDSofT* combines power with ease of use. By using *CCDSofT* in conjunction with *TheSky*, you can control both computer-driven telescopes and CCD cameras from your computer, creating an unmatched system for astronomical observations.” Info: <http://www.bisque.com>.

**Computer Aided Astronomy (CCDAst).** By John Rogers. Windows/PC-

## Section B.6: Image Processing

based, commercial. Performs astrometry on CCD images. Info: <http://ourworld.compuserve.com/homepages/johnerogers/>.

*CCD Image Processing*. By Frank Jan Sorenson. Windows/PC-based shareware, Cookbook oriented. Info: <http://home4.inet.tele.dk/frank.s/ccdprog.html>.

*EZPhot*. By Jerry B. Gunn. Photometry software. Info: <http://www.mtco.com/~jgunn>.

*Hartmann-Mask Analysis Software*. By Santa Barbara Instrument Group. Windows/PC-based, commercial. Info: <http://www.sbig.com/sbwht-mls/hartmann.html>.

*Image Scientist*. By Mike Gutzwiller. Windows/PC-based, commercial. Originally named *Deep Sky*. Written to analyze data coming from automated CCD cameras. “Look at, process, or build any image size.” Many processing/enhancement features. Info: <http://www.imagescientist.com>.

*Images Plus*. By Mike Unsold. Windows-based general-purpose image processing software, featuring support for digital SLR cameras. 4193 Tallmadge Road, Rootstown, OH 44272 (330) 325-0765 [www.mlunsold.com](http://www.mlunsold.com).

*IRAF (Image Reduction and Analysis Facility)*. By the IRAF Working Group. Unix-based, open source, public domain. *IRAF* is the package that professional astronomers use for processing *their* images. Definitely a high-end program, with an infamous user interface. Info: <http://iraf.noao.edu/iraf-homepage.html>.

*IRIS-Prism*. By Christian Buil. Windows/PC-based, freeware. This interesting hybrid features a command-line/script driven interface embedded in a GUI software environment. Many powerful and specialized commands and a well done on-line user manual. Info: <http://www.astrosurf.org/buil/iris/iris.html>.

*MaxIm DL*. By Cyanogen Productions, Inc. Windows/PC-based, commercial. “*Maxim DL* includes a variety of image processing and image enhancement tools including flexible kernel and FFT filtering, unsharp masking, nonlinear and histogram-based stretching, and color processing.” Demo version available as a free download. Info: <http://www.cyanogen.com>.

*Megafix*. By Bruce Johnson Computing. Protected-mode-DOS/PC-based, commercial. Supports 8-bit and 16-bit CCD cameras, 480 x 640 and 800 x 600 256-color VESA modes, with access to high memory. Info: <http://members.aol.com/BJohns7764/BGfix.html>.

*MIRA*. By Axiom Research (Mike Newberry). DOS and Windows versions/PC-based, commercial. “Astronomical image processing software for the Intel platform.” Oriented toward sophisticated scientific im-

## Appendix B: Resources

age analysis. “*MIRA* offers such features as curve of growth aperture photometry using circular or elliptical apertures, elliptical isophote flitting and isophotal photometry, image animation, automatic computation of the translational and rotational transforms for image registration, convolution with linear and nonlinear operators,” and much more. Info: <http://www.axres.com/>.

*NIH Image*. By the National Institutes for Health (Wayne Rasbud). “*NIH Image* is a public domain image processing and analysis program for the Macintosh.” An excellent program for scientific image processing (and free, too!) Info: <http://128.231.98.16/nih-image/about.html>.

*Pises Atlas Prism 98*. By F. Colas. Windows/PC-based, commercial. Combines image pre-processing and processing module, a sky-map module, a reference file of images taken at the Pises Observatory, and image acquisition software for SBIG and HiSys CCD cameras. Shareware version available on floppy disk. Info: <http://pages.pratique.fr/~ccavador/Features.html>.

*Quantum Image*. By Christer Strandh. Windows/PC-based. Quantum Image is “a single multi-faceted image processing program, but one which will in the future be divided into a lighter version and a more fully developed professional version.” Limited demonstration version available for download. Info: <http://www.quantumimage.com>.

*WinMiPS*. By Christian Buil and E. Thouvenot. Windows/PC-based, commercial. “*WinMiPS* is open and evolutionary. It is a platform up to answering quickly the observer’s specific needs; its concept goes far beyond. The current version owns instruments for automatic pre-treatment, photometric analysis, Gaussian and Moffat synthesis, and also astrometry.” Info: <http://www.bdl.fr/52P/pises.html>.

## B.7 Color Imaging

### B.7.1 Color Imaging Books

Committee on Colorimetry. *The Science of Color*. The Optical Society of America, 1963. Compiled by color experts from science and industry, this volume remains among the most comprehensible expositions on colorimetry available.

Fairchild, Mark D. *Color Appearance Models*. Reading, MA: Addison-Wesley, 1998. This book deals with measuring color as humans see it, including discussion of viewing conditions, chromatic adaptation, and color appearance modeling. An excellent source for amateurs seriously interested in accurate color rendition.

Hunt, R.W.G. *The Reproduction of Colour in Photography, Printing, and Television*. 4th ed. Tolworth, England: Fountain Press, 1987. This

## Section B.7: Color Imaging

volume is the most comprehensive discussion of color we have found. The sections on television are relevant to producing color on computer systems. It covers everything from color fundamentals to the matrix algebra needed to transform color between different color spaces.

Giorgianni, Edward J., and Thomas E. Madden. *Digital Color Management, Encoding Solutions*. Reading, MA: Addison-Wesley, 1997. For anyone interested in reproducing digital color on computers, this is *the* book. The authors, with the blessing of Eastman Kodak, describe color capture, color encoding, and color display for the digital age, from scanned film to output on color monitors. Detailed technical appendices cover topics from colorimetry to transforms between color spaces.

## Appendix B: Resources

---

# Appendix C Tutorials

---

We have written these tutorials to help you master the **AIP4Win** image-processing software program included with this book. Each tutorial walks you through activities designed to clarify and reinforce the concepts that you have learned in the book. For the best learning experience, we recommend that you first follow each tutorial exactly as we have written it. Then repeat the tutorial while experimenting and trying out new processing variations using your own CCD images.

We find the best way to use these tutorials is to run **AIP4Win** on your computer as you follow the instructions in the book. You will find test images on the accompanying CD-ROM in the Tutorials directory. Each tutorial covers a specific set of program features and builds a specific set of image-processing skills; and, of course, each builds on the skills you have acquired in the previous ones. The tutorials are listed in Table C.1.

**How to Install AIP4Win.** If you have not installed **AIP4Win** on your computer, you must do so before you can run any tutorial. To install **AIP4Win** on your computer, remove the CDROM from its pocket in the back of the book. Insert the CDROM in your CDROM reader and the installation package will run automatically. If it does not run automatically, use the Windows Explorer to browse to the CDROM and double-click on the file “Setup.exe” and the installer will start.

The first time you run AIP4Win, the program will ask for your name plus the serial number. You will find your serial number on a sticker in the back of the book, below the CDROM. You MUST register your copy of **AIP4Win** within 30 days. To register, go to the Willmann-Bell website (<http://www.willbell.com>) and browse to the **AIP4Win** webpage. You will be asked to enter your name and serial number as well as the Installation Code provided by **AIP4Win**. You will receive a Registration Code that you should enter into the program. **AIP4Win** will run for up to 30 days if it has not been registered. After that it will not run until you have registered it.

**Important:** If you purchased the first edition of this book and installed **AIP4Win 1.x** on your computer, there is no need to remove it. You can keep the earlier version while installing and running **AIP4Win 2.x**. This allows you to learn **AIP4Win 2.x** without losing the familiar **AIP4Win 1.x** environment.

**Using the Tutorials.** To make the tutorials easier to follow, the typography highlights menu items and control buttons. Menu selections are printed in an *italic typeface*, and buttons are shown in **sans-serif bold** to make them stand out.

## Appendix C: Tutorials

Whenever we suggest that you select an item on a drop-down menu, we write it as *Menu|Submenu*, meaning that you should select the *Submenu* item in the *Menu*. To open a file, we'll say, "click on *File|Open Image*." This means that you should click on *File* in the top menu bar, and then click on the *Open Image* item that appears on the drop-down menu. For deeper menu levels, we write, "*Menu|Submenu|Submenu*."

When you select menu items in **AIP4Win**, you often call a tool window. We think of tool windows (or "tools," for short) as the basic units of **AIP4Win**, just as each item in a carpenter's tool chest performs one of the basic functions of carpentry. The tool windows in **AIP4Win** operate on the currently selected image, using the settings that you make using the text boxes, sliders, check boxes, and buttons on the tool window. To make the names of buttons and other controls stand out clearly, we refer to them like this: **Select Dark(s)**. Used in concert, the tool windows provide a powerful yet flexible software package.

Clicking on a menu item or a tool window button often brings up a dialog box, such as the familiar Windows dialog for opening a document or file. You must enter the information requested and then click the **Okay** or **Cancel** button. When a process is running, **AIP4Win** sometimes displays a progress bar to let you know how much longer it will take, or a message box with information that you should be aware of. If you are familiar with the Windows operating system, these are undoubtedly already familiar features.

Many of the tool windows that you can access from the menus are also available on the toolbar. Although we tell you to make menu selections in these tutorials, you will probably want to learn and use the icons on the toolbar as well.

**Before You Start...** remember that **AIP4Win** is unique. It is a program conceived, designed, and optimized for processing *astronomical images*. **AIP4Win** therefore offers unique tools and powerful functions for calibrating, track-and-stacking, image enhancement, and processing images in the spatial domain. **AIP4Win** was designed to handle both the dynamic range of deep-sky images and the restricted contrasts of planetary images. Its suite of measurement tools includes astrometric, photometric, and spectroscopic functions that are specific to celestial images. If you have used image processing programs designed for photographers and graphic artists, set aside your preconceptions and prepare to enter the world of *Astronomical Image Processing*.

**About the images used in these tutorials.** The images have been chosen to illustrate specific features of the program. All of the images were all taken by amateur astronomers using modest equipment. None were taken using expensive, high-end gear. Our goal in doing this was to give novice CCD imagers a feeling for what they can expect to accomplish with equipment that a novice might own. You will also notice that some of the images are on the small side. This was done so that your time can be spent learning from the tutorials rather than watching your computer crunch numbers in an 11 megapixel image.

**Table C.1**

Section Number	Subject(s) covered by tutorial
C.1	<b>Basic Skills:</b> Learn how to open images and control their appearance on the screen of your computer. If you've never done image processing before, this is a real eye-opener! Run this tutorial before you try anything else.
C.2	<b>Calibration:</b> Demonstrates how you can subtract dark frames and apply flat-field frames either manually or automatically. Calibration is an essential step in the creation of every CCD image. For background information, consult Chapter 6.
C.3	<b>Image Evaluation:</b> Dig into the numerical nuts and bolts of images using <b>AIP4Win</b> 's comprehensive suite of software tools for measuring and evaluating images. For background, see Chapter 7.
C.4	<b>Astrometry:</b> Make precise measurements of right ascension and declination for objects in your images. See Chapter 9.
C.5	<b>Photometry:</b> Learn how you can derive precise magnitudes from your CCD images, and make light curves for variable stars. For background reading, check out Chapter 10.
C.6	<b>Spectroscopy:</b> Discover hands-on what astronomers see when they classify stars by spectral type. See Chapter 11.
C.7	<b>Image Enhancement:</b> Discover how to extract detail from otherwise bland images. More than producing "pretty pictures," the techniques you will demonstrate to yourself enhance details and show structures that, due to their low contrast, might otherwise be invisible. These techniques are covered in Chapters 13, 14, 15 and 18.
C.8	<b>Fast Fourier Transform:</b> Experiment with the power of spatial filtering to enhance and analyze your images. See Chapter 17.
C.9	<b>Multiple Image Processing:</b> Here is power at your fingertips! Calibrate an entire observing session's worth of images at one time automatically. Align and enhance a set of images in preparation for creating a movie. Align and combine a group of images to create a single, "deeper" image. Process hundreds of planetary images.
C.10	<b>Image Registration and Blinking:</b> Registration and blinking are key tools in searching for asteroids and patrolling for supernovae. In this tutorial, you'll see how it's done. See Chapter 16.
C.11	<b>Deep-Sky Images:</b> Learn the best ways to process a wide variety of deep-sky images, including the calibration and enhancement of a typical track-and-stack one.
C.12	<b>Planetary Images:</b> In this tutorial, you process an outstanding image of Jupiter using brightness scaling, unsharp masking, and deconvolution tools.
C.13	<b>Building Color Images:</b> Learn how the <b>AIP4Win</b> color tool helps you to create stunning color images—hassle-free—from sets of red/green/blue filtered images. For background, see Chapter 20.

## Appendix C: Tutorials

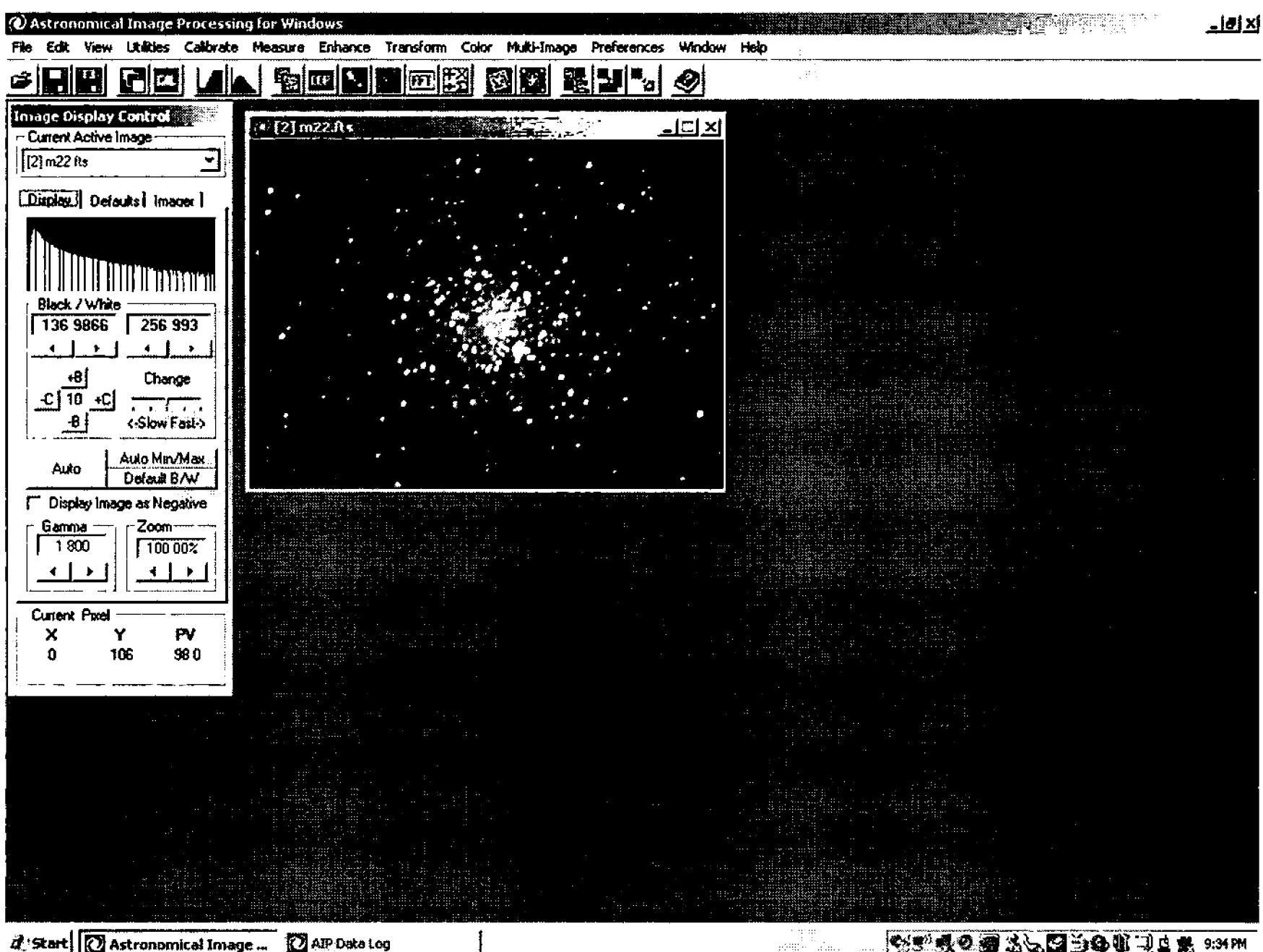


Figure C.1 As you load an image, the Image Display Control appears at the upper left side of the screen. This tool never changes your image data, it controls every aspect of how your images appear on the computer screen—their size, shape, brightness, contrast, and display mode.

### C.1 Basic Skills

In this tutorial you will load, display, and save images using **AIP4Win**. This one is the longest of all because it covers most of the basic functions of the program. You will learn these specific skills:

- How to load and display CCD images and other images using standard PC file formats,
- how to control the way an image is displayed,
- how to customize some of the features of **AIP4Win**,
- how to save images in FITS format, and
- how to export images using standard PC file formats.

The files for this tutorial are found in the Tutorials directory on the CD-ROM under the Using **AIP4Win** subdirectory.

**Step 1: Starting the Program.** Start **AIP4Win** by either clicking on the desktop icon or by running **AIP4Win** from the Start Menu. The program will load and display its introduction panel.

**Step 2: Loading an Image.** We'll begin by loading a CCD image that has been stored in the FITS format. Besides FITS, **AIP4WIN** supports all well-documented proprietary CCD camera formats. Click *File|Open Image...*, a dialog box

will pop up. Use it to select your CD-ROM drive and navigate to the Basic Skills subdirectory of the Tutorials directory on the CD-ROM, select the file “m22.fts,” and click **Open**. The file should display on the screen.

**Step 3: Auto-Stretch.** If the image appears all black, click the **Auto** button on the Image Display Control on the left side of the screen. The image should now display in a continuous range of gray tones. The auto-stretch feature automatically determines the range of gray levels best suited to display an image by examining the histogram of the image. It then stretches the image linearly to display the greatest contrast. To “unstretch” the image, click **Auto Min/Max** on the Display Control. The image should now appear mostly black, with only the brighter stars visible. Click **Auto** again and the image will reappear. By default, images are stretched when they are loaded to make them visible. The **Default Display Mode** buttons on the **Defaults** tab can turn the automatic stretching feature on and off. You can also change the default to one of the other stretch modes using these buttons. It is important to note that none of the controls on the Display Control window affect the original image, they only affect the way it is displayed on the screen.

**Step 4: Display Controls.** The Display Control window contains a group of controls which give you complete flexibility in how an image is displayed. The **Black/White** control determines the lower cutoff value, below which any pixel is displayed as black and the upper cutoff value, above which any pixel is displayed as white. Any pixels between these two values will be linearly stretched between black and white. Above these controls is a small display of the portion of the image histogram that is currently viewed. This display shows you in real time what changes you are making as you adjust the controls.

As you increase the **Black** control, dimmer stars and the sky background will start to disappear; and at the same time you will see the histogram stretch out off the left end of its display. As you decrease this control, faint stars, sky background and noise will start to appear. Decreasing the **White** control will cause brighter stars to saturate the display, burning in the center of the image. You will see the histogram stretch off the right end of its window. Increasing the **White** control causes the display to dim. Brightness and contrast of the image can be varied using these two controls.

The **Change** slider is used to adjust the sensitivity of these controls. Moving it toward the fast end makes the controls increase in large steps when the up and down arrow buttons are clicked. Moving the **Change** slider toward the slow end allows you to make changes in small steps.

You can also adjust the brightness and contrast using the **+B**, **-B**, **+C** and **-C** buttons. These buttons affect both the black and white values, and their sensitivity is also controlled by the setting of the **Change** slider.

Finally, you can type numerical values into the **Black** and **White** text boxes. When you press the enter key, the image display is updated using the values you have entered. When you have finished experimenting with the Black and White setting, click the **Auto** button.

**Step 5: The Gamma Control.** Try clicking the up and down buttons on the

## Appendix C: Tutorials

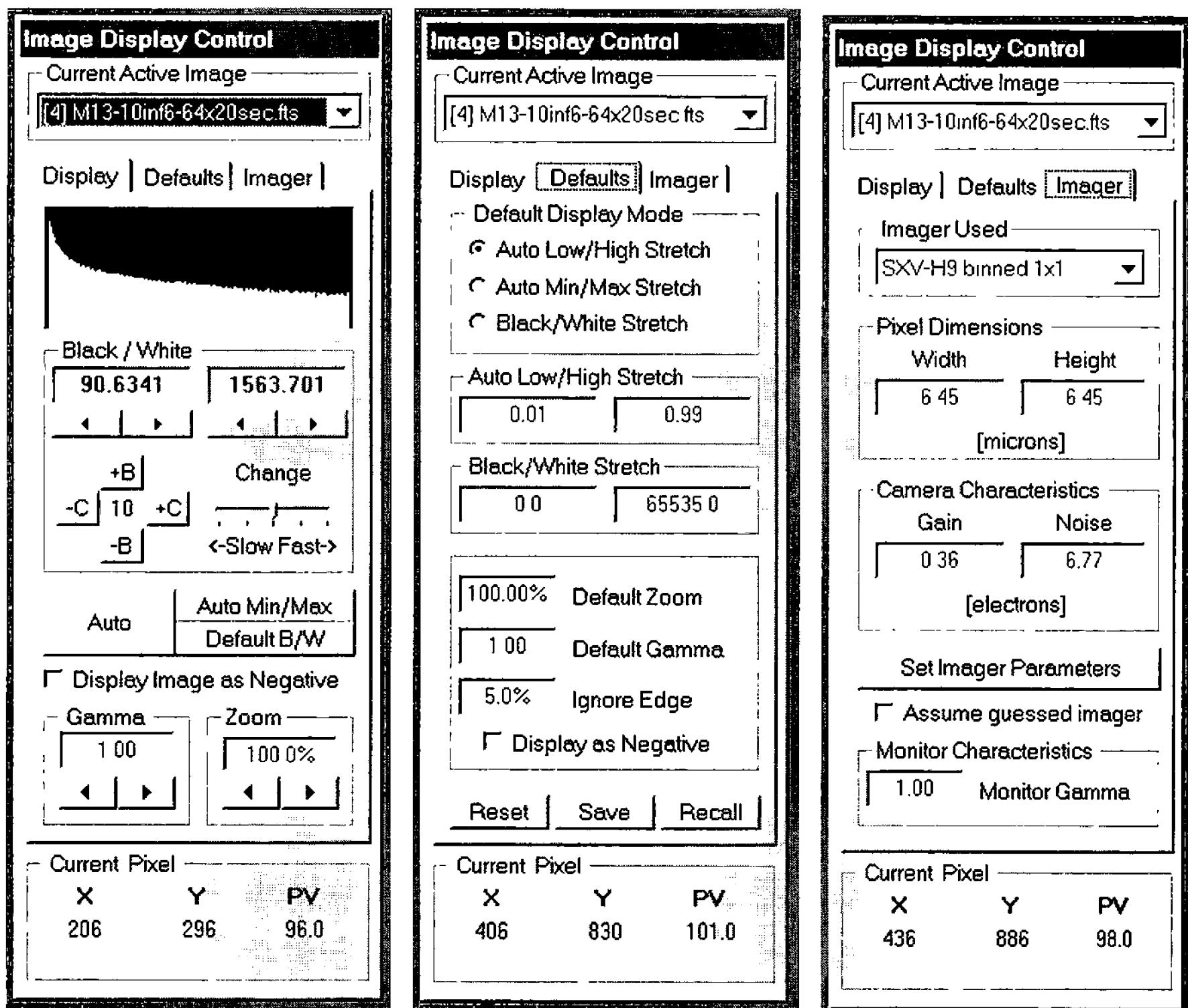


Figure C.2 The three tabs on the Image Display Control control different aspects of the image. The Display tab controls how an image appears right now, the Defaults tab controls the initial settings when you open a new image, and the Imager tab sets the pixel size and camera characteristics.

**Gamma** control. It initially defaults to 1.00, and as you decrease its value, the dimmer stars in the image start to disappear. Increase it and more stars will appear. The **Gamma** control makes the stretch between the minimum and maximum pixel values non-linear, either boosting or dropping the mid-tone values using a power-law scaling. This control is useful for searching out faint objects in an image, as it increases the brightness of faint details while leaving the existing brighter areas alone.

You can set the default Gamma value using the control on the **Defaults** tab of the Display Control.

**Step 6: The Zoom Control.** Try clicking the **Zoom** control. As expected, the image is displayed larger or smaller. You can also type in a desired zoom value (with or without the “%” sign) and the image will zoom to that ratio. The range of the zoom is from 16.67% to 1600%. As with the other controls, the default value can be set on the **Defaults** tab. Reset the Zoom to 100% and click the **Auto** button to return the image to its original state. Now go to the **Defaults** tab.

**Step 7: The Defaults Tab.** For a more complete description of this tab, just

hit the F1 key while the mouse is in this window to bring up the help system. **AIP4Win** provides a complete, context-sensitive help system. At any time, hitting the F1 key will bring up the page in the help file corresponding to the currently selected tool or control.

The **Default Display Mode** is used to control the way an image is displayed when it is first loaded. You have a choice of stretching the image automatically between a low point and a high point, stretching it between the minimum and maximum values found in the image, or between a set of user-specified values.

The **Auto Low/High Stretch** settings are a pair of text boxes where you enter the fraction of the image pixels used to set the black value and the fraction of the image pixels used to set the white value used by the **Auto** button.

The **Black/White Stretch** settings are used to specify a fixed black and white value, used by the **Default B/W** button on the **Display** tab.

The **Default Zoom** textbox is used to set the magnification of the image when it is first loaded into **AIP4Win**.

The **Default Gamma** textbox is used to set the gamma of the image when it is first loaded into **AIP4Win**.

The **Ignore Edge** textbox contains a percentage of the image border that is to be ignored when the statistics are collected to determine the auto low/high stretch. This is often necessary with CCD images that contain artifacts around their borders that can throw off the display calculations.

You can save your favorite values for these defaults when you click the **Save** button to store your settings in the Windows Registry for when you next start-up **AIP4Win**. If you mess up your settings, and want to return to your previously saved defaults, click the **Recall** button to retrieve them from the Windows Registry. Clicking the **Reset** button returns the values in the settings to the original **AIP4Win** defaults, but does not affect the settings in the Windows Registry.

**Step 8: Loading Images.** In addition to FITS and proprietary CCD camera formats, **AIP4Win** opens over 40 different PC graphics formats, plus raw files from a number of different digital cameras.

**AIP4Win** reads the image files and loads them into an internal 32-bit floating point storage array. CCD camera files are in formats such as FITS, STx, Cook-book, etc., which preserve all the image data read from the CCD chip. Almost all PC image file formats can record, at most, only 256 gray levels. Even if the format is capable of representing 16 million colors, only 256 gray levels are available. This means that information is lost when a 14- or 16-bit CCD camera stores an image in one of these formats. Regardless of the source of the image, **AIP4Win** promotes the incoming image to its internal format.

For those formats that provide color information, **AIP4Win** also stores the color channels using its internal floating point format. To indicate to a user whether an image contains color data, the icon in the upper left corner of the image window is shown in color. Grayscale images show a grayscale icon.

To load an image, you need only to select *Open Image...* from the *File* menu,

## Appendix C: Tutorials

or click the folder icon on the toolbar. The image will load, and the appropriate menu items and toolbar buttons will be enabled.

**Step 9: Saving Images.** Click on the “m22.fts” image to select it. Clicking on an image makes it the “active” or “currently selected” one. To save this image as a FITS file, to preserve its full range of data, click *File|Save as FITS...*, this will bring up the Save FITS File dialog box. Select a directory (not on the CD-ROM) in which to save the image and type in a name for it, with an extension of “.fts” or “.fit,” the two standard file extensions for FITS files. Click **Save** and the Save as FITS File window will appear. This window provides a means for you to select the FITS format you want to use, as well as the means to “coerce” the numbers in your image to fit into any formats which cannot handle the range of values contained in it. The default format is 32-bit floating point (the data format that **AIP4Win** uses internally), but you also have the option to save images in a variety of other FITS data formats. Take care to store your images in a format that supports the range or numerical resolution of their data; if you do not, information data will be lost when you save the image. By saving your images using 32-bit floating point, your data will be preserved.

It is important to remember that other image processing programs, even some that support FITS, may not support 32-bit floating-point FITS files; so it is wise to make sure before using this format to share your files with them. The most universal implementation of FITS is the 16-bit signed integer format; so if in doubt use this format if you have to. Whatever format you select will be remembered for the next time you save an image.

Another feature that is provided when saving FITS files is the ability to edit the file header. To use this capability, click the **Examine/Edit FITS Header** button, and the FITS Header Editor window will pop up. This window is a simple editor that will allow you to view the Fixed Lines (those which contain data that, if modified, could render the file unreadable), and a group of User Editable Lines. **AIP4Win** will have pre-loaded a set of lines for you, which you can modify as you see fit. If you change any lines, or insert your own, make sure you follow the FITS specification and maintain the keyword usage if you want to be able to read the data back in the future. In general, you can type anything you want after a COMMENT or HISTORY keyword, as long as the line does not exceed 80 characters (the width of the FITS Header Editor box).

**Step 10: Exporting Images.** Most graphics arts programs such as PhotoShop and PaintShop Pro can only read PC format files (unless special plug-ins are installed). Most internet web browsers read just a limited variety of files, none of them CCD camera or FITS files. If you want to publish images or send them to friends, you will need to export them in a PC file format such as JPEG, TIFF, etc. **AIP4Win** supports over 40 PC file formats. To output a file in one of these, select the image of the M22 globular cluster we have been using and select *File|Export*. This will bring up the Export File dialog. Select the directory and type in the name you want to export the image as with an extension of “.jpg.” Click **Save** and the JPEG Options window will pop up. Here you will be able to select the compression

and image quality you want. Click **OK** and the file will be stored in JPEG format, a good one for web publishing. Click *File|Open Image...* and select the image you just saved from the Open Image dialog, and the image will be loaded and displayed.

**Conclusion.** This tutorial has shown you how you can use **AIP4Win** to load and display several types of images and how to control the way they are displayed on the screen. You have also seen how to save images in several file formats. In the next tutorials you will see how we can use **AIP4Win** to calibrate, analyze and enhance images.

## C.2 Calibration

This tutorial demonstrates how to use **AIP4Win** to calibrate CCD images. The supporting images for it are contained in the Calibration subdirectory of the Tutorials directory on the CD-ROM.

A frame produced by a CCD camera contains a lot more than an impression of the light that fell on it. As described in Chapter 6, the frame also contains noise and artifacts. Using **AIP4Win**'s calibration tools we can remove most of this noise and generate an image with scientifically useful information, which is usually more pleasing to the eye as well.

**Step 1: Opening and Examining a Raw CCD Image.** From the Calibration Tutorial directory, open the file “m13.fts.” Use the *File|View Header...* menu item to examine the header of this FITS file, which contains useful information about the image. In this case, you can see that it is a 15-second exposure of the globular cluster M13. Close the FITS header display when you are finished.

Click the **Auto** button on the Display Control if the image appears all black. To automatically stretch each image as it is loaded, just check the **Auto Low/High Stretch** button on the **Defaults** tab. This tutorial and those that follow will assume that this button is checked.

Examine the image and note all the single-pixel “stars.” Use the Zoom function in the Display Control to see them better. These are hot pixels in the CCD frame that are brighter than the rest because of the higher rate at which they accumulate thermal charge.

**Step 2: Opening and Examining a Dark Frame.** From the calibration tutorial directory, open the file “dark00001.fts.” An examination of the FITS header for this file will show that it is a 15-second dark frame, taken around the same time as the “m13.fts” image frame. A dark frame should be taken for the same duration as the image frame and should be taken around the same point in time in hopes that the CCD will be at the same temperature for both exposures.

Click the **Auto Min/Max** button and examine the dark frame. Notice that it appears as a basically black frame with a scattering of white pixels. This particular dark frame was taken with a Cookbook CCD camera using the Low Dark Current feature. The Cookbook camera with the LDC feature shut off and other CCD cameras often produce dark frames with a salt-and-pepper look.

## Appendix C: Tutorials

Notice the **Black/White** boxes on the display control. Click the *Measure Statistics|Image...* menu item to display the statistics for this image. (If the Data Log window pops up, minimize it using the – button on its titlebar.) The Image Status window shows that this dark frame has a small dynamic range, from 95 to 124, just 30 ADUs. The background is right around 97 ADUs. This is the image bias, plus a small amount of dark current. As you pass the cursor over the frame, the pixel value under it is shown at the bottom of the Display Control, along with the pixel coordinates. As you move the cursor around the dark frame, notice how the values change. Dismiss the info window by clicking the **X** button on its title bar.

Each hot pixel in the dark frame corresponds to a hot pixel in the image frame. An image accumulates “on top” of the dark frame. That means that the dark frame can be simply subtracted from the image. When you do this, you are also subtracting the bias.

To subtract the dark frame from the image manually, click the *Multi-Image|Image Math...* menu item and select “m13.fts” for the top text box, “minus” for the middle text box and “drk00001.fts” in the bottom text box. Click **Perform Math Operation** and a new image is created, which is the M13 image with the dark frame removed. Notice how the hot pixels are gone. What you have done here is to calibrate your image manually. As you will see, **AIP4Win** provides a much more sophisticated way of performing basic calibration.

Close the “drk00001.fts” image and the calibrated image you just created.

**Step 3: Basic Calibration.** **AIP4Win** provides an automated tool, AutoCalibrate, for calibrating your images. This tool needs to be initialized before you apply it to an image. There are three kinds of calibration protocols provided by **AIP4Win**. In this step we will use the most basic protocol, which simply subtracts a composite dark frame from an image.

Select the *Calibrate|Setup...* menu item and the Calibration Setup window will appear. This window can also be invoked using the Setup Calibration button on the toolbar. The floating text that appears over each button when you pass the mouse cursor over it will show you which button is which.

The Calibration Setup Tool has a drop-down listbox at the top with which you select the calibration protocol you wish to use. For this exercise, select Basic.

The Basic Calibration tab has three buttons of which only the **Select Darkframe(s)** one is enabled. It also has a pair of radio buttons that allows you to select **Average Combine** or **Median Combine** of dark frames. Next to the **Select Darkframe(s)** button is a red light. This light will turn green once you have selected one or more dark frames. A text box below this button shows you how many dark frames you have selected.

Click the **Select Darkframe(s)** button. When the dialog box opens, navigate to the Calibration Tutorials directory and click on the file “drk00001.fts.” Hold down the shift key and click on the file “drk00008.fts.” Now click the **Open** button; the dialog box will disappear, and the Basic Calibration window will display a green light next to the **Select Darkframe(s)** button and show that eight dark

frames have been selected.

Combining multiple dark frames allows you to reduce the noise in the “master” dark frame that you will subtract from your image (see Chapter 6). **AIP4Win** provides two ways of combining these images: averaging them together or taking a median of the corresponding pixels in each frame. Of the two methods, median-combining them is more processing-intensive, but it will remove artifacts that appear in only a few frames, such as cosmic ray hits. Average-combining dark frames is quicker, but random noise is not completely suppressed and may leave artifacts in your calibrated images. Since we only have eight dark frames for this tutorial, we will use median combine. Click the **Median Combine** radio button.

Now click the **Process Dark Frame(s)** button. You will see a status bar that will display the progress of the median operation. When the operation is done, a green light appears next to the **Process Dark Frame(s)** button, and the **Save as Master Dark...** button is enabled.

**AIP4Win** allows you to save your master dark frame so that you don’t have to median-combine or average-combine a set of dark frames all over again when you want to re-use them. You can save your master dark frame as a FITS file and just load it as a single file when you click the **Select Darkframe(s)** button. This can save a lot of time, especially when you have median-combined a large number of dark frames.

For now we will not save the master dark frame. When you created it, it was automatically loaded into the dark-frame buffer maintained by the program. Whenever you want to see what frames constitute the master dark, just select the *Calibrate|Master Frames|Master Dark Contents* menu item, and it will display a list of the current contents of the master dark buffer. Dismiss the Calibration Setup window.

Now that the setup has been completed, we can calibrate the image. Click the *Calibrate|AutoCalibrate* menu item or click the **AutoCalibrate** button on the toolbar. A new image will be displayed, which is a calibrated version of the M13 image. Notice that the hot pixels have been removed.

**Step 4: Opening and Examining a Flat Frame.** As described in Chapter 6, not all pixels on a CCD chip respond to light and build up charge at the same rate. Also, dust and vignetting in the optical system can cast shadows on the CCD chip. Taking a flat-field exposure allows you to calibrate out these effects.

Open the file “flat00001.fts” in the calibration tutorial directory. When this image is stretched, you will see a lighter area in the center of the field, gradually darkening towards the edges, and a bunch of variously sized black donuts. The darkening around the edge is due to vignetting, and the donuts are dust specks on various optical surfaces in the system.

**Step 5: Standard Calibration.** This protocol is very similar to Basic Calibration, but adds the step of dividing the image by the flat field. Click the *Calibrate|Setup...* menu item and the Calibration Setup window will appear. This time, select Standard in the dropdown box.

## Appendix C: Tutorials

In addition to the **Dark** tab you saw when Basic Calibration was selected, there is now a **Flat** tab that has a set of similar controls used to manage the flat frames. As with the dark frames, you can average- or median-combine multiple flats to reduce noise, and you can also subtract a special flat-dark frame from the flat frame in order to remove the dark current and hot pixels from it. Note that this flat-dark corresponds to the flat frame and not to the image frame. Don't get them confused.

The **Dark** tab should still be set up from before; if it isn't, select the same dark frames you used above using the **Select Darkframe(s)** button and process the dark by median-combining those frames. Now go to the **Flat** tab and select the files "flat00001.fts" through "flat00008.fts" for use as flat frames and select **Median Combine** for them. Click the **Subtract Flat-Dark** checkbox and the **Select Flat-Darks** button will be enabled. Select the files "flatdrk00001.fts" through "flatdrk-00008.fts" for use as flat-darks, and select **Median Combine** for them.

Now click the **Process Flat Frame(s)** button, and **AIP4Win** will calibrate your flat frames, combine them and load them into the flat buffer for you. As with the master dark frame, you have the option of saving it as a FITS file for later use. Dismiss the Calibration Setup window.

Now click on the raw M13 image and use the **AutoCalibrate** button on the toolbar to create a new image that has been calibrated using both dark subtraction and flat-fielding.

**Step 6: Advanced Calibration.** This calibration method builds on the previous protocols, but it allows you to use a single master dark frame for all of your images, given that they are taken using the same camera operating mode and that the dark frame is at least as long in duration as the longest image it will be used to calibrate (preferably at least 5 times longer). The availability of this feature means that you can take one set of dark frames at any time during the night and use it on any of your images taken on that night, or, possibly, any other night (hot pixels tend to be pretty constant, though it does pay to check your dark frames from time to time to see if they have changed).

Click the *Calibrate|Setup...* menu item, or click the **Calibration Setup** button on the toolbar, and the Calibration Setup window will appear. Select Advanced in the drop-down box. Similar to the Basic and Standard Calibration protocols, there are tabs for each type of calibration frame.

The **Bias** tab contains a set of controls used to manipulate the bias on the calibrated image. It has three bias options selected by mutually-exclusive radio buttons:

1. **No Bias Removal.** Selecting this button will disable the bias functions and the dark-matching and dark-scaling capabilities.
2. **Use Constant.** Selecting this button will allow you to enter a fixed bias value to be used instead of a bias frame.
3. **Use Bias Frame.** Selecting this button will allow you to select one or more bias frames which can be averaged or median-combined to create a master bias frame.

For this tutorial, select the **Use Bias Frame** button.

Load the bias frames by clicking the **Select Bias Frame(s)** button and the bias frame selection dialog box will appear. Navigate to the calibration tutorial directory and select the files “bias00001.fts” through “bias00008.fts” and click **Open**. The green light next to the **Select Bias Frame(s)** button will turn on and the text box will indicate that eight bias frames have been selected. Click the **Median Combine** radio button for the bias frames. Now click the **Process Bias Frame(s)** button, and the program will median-combine the eight bias frames into a single master bias frame. You have the option of saving this frame as a FITS file for later use, just as in the case of master dark or flat frames.

On the **Dark** tab, dark frames are handled just like in the Basic and Standard calibration protocols, but you now have options to scale the dark frame to match it to the image frame. Select the dark frames “dark00001.fts” through “dark00008.fts,” and select **Median Combine** for them. For this tutorial, we will use automatic dark matching, so click the **Automatic Dark Matching** radio button. Now click the **Process Dark Frame(s)** button, and the program will load your dark buffer. You will notice that it runs through a series of processing steps, including identifying the hot pixels in the dark frame. When it is finished, the green light next to the **Process Dark Frame(s)** button will turn on. Again, you have the option of saving this master dark frame.

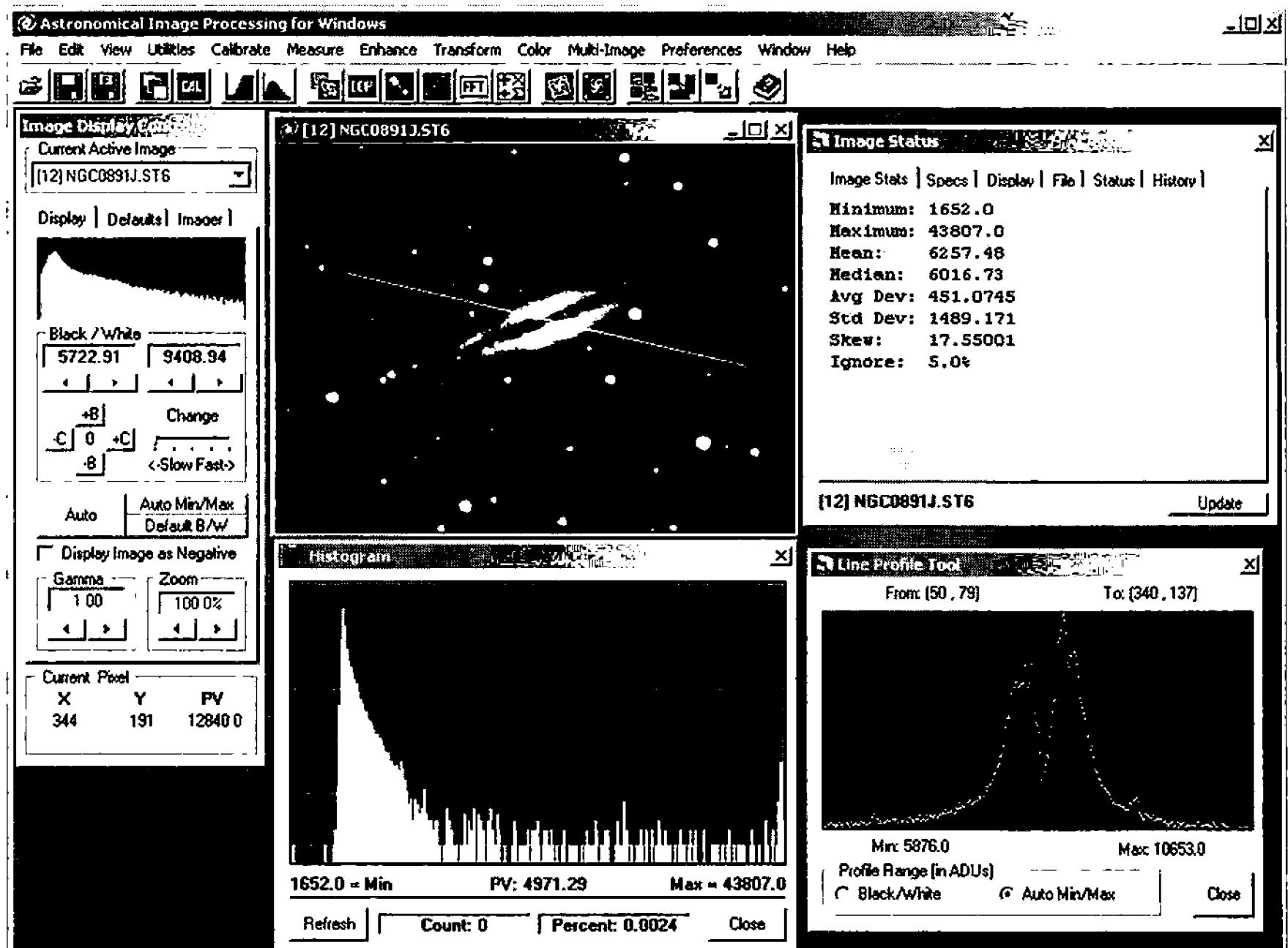
If the **Flat** tab is not already set up from earlier, select the files “flat00001.fts” through “flat00008.fts” as your flat frames and select Median Combine. You also have the option here of using Normalize Median Combine, which is meant for processing sky flats taken from a twilight sky. For that type of flat frame, each image has a widely varying average pixel value, as the twilight sky dims rapidly. Normalize Median Combine allows you to scale all your flats to the same average level before they are combined. It doesn’t hurt to combine dome flats (or flatbox flats) this way, but the extra processing can take time.

Check the **Subtract Flat-Dark** checkbox and select the files “fltdrk00001.fts” through “fltdrk00008.fts” as your flat-darks. Select Median Combine for them and click the **Process Flat Frame(s)** button; the program will load the flat buffer for you, which you have the option of saving.

Select the **Defect** tab, and you will be presented with a single button with which you can select a defect map. A defect map is a specially prepared image that identifies each defect in your CCD image and provides a code that tells **AIP4Win** how to deal with it. For the purposes of this tutorial, we will not perform defect correction. Because a defect map was not selected, the **Correct Defects** checkbox at the bottom of the form is not checked and is grayed out. However, all the other checkboxes should be checked. At this point the advanced calibration is set up and ready to go to work, but don’t dismiss the Calibration Setup window yet.

If not already loaded, load the “m13.fts” image and then click the **AutoCalibrate** button on the toolbar. A new, calibrated image of M13 will appear, and in the titlebar of the new image the display “Dark Coeff = 1.008” is shown. This value is the scaling applied to the master dark frame after the bias has been removed

## Appendix C: Tutorials



**Figure C.3** Tools in the *Measure* menu produce information about an image, from its statistical properties, processing history, and histogram to brightness profiles through the image. Because effective image processing begins with measuring image characteristics, you'll often apply these measuring tools.

and before it is subtracted from the raw image. Had you selected Constant Dark Scaling when setting up the dark frame for this calibration, that is the fixed value you would have entered in the **Dark Scaling Constant** text box.

**Step 7: Partial Calibration.** At the bottom of the Calibration Setup form you will see four checkboxes, one for each step in image calibration. When the corresponding box is checked, each calibration step will be taken on your image. If you have your calibration set up, and decide not to perform one or more of these calibration steps, just uncheck the corresponding box. You can turn it back on again later by opening this form and checking the box. If a particular step has not yet been set up, its box will be grayed out. You can re-enable it by setting up the corresponding bias, dark or flat frames, or the defect map.

This completes the image calibration tutorial. If you want more practice, you will find a variety of additional images in the **Images\Berry**, **Images\Burnell** and the **Images\McMickle** subdirectories on the CD-ROM. Be sure to read the file “**About\_These\_Images.txt**” in each directory to see what calibration frames are associated with what images.

## C.3 Image Evaluation

In this tutorial, you will learn how to use **AIP4Win**'s measurement tools for evaluating image characteristics. The images for this tutorial can be found in the Image Evaluation subdirectory of the Tutorials directory on the CD-ROM.

**AIP4Win** provides a series of tools used to examine and evaluate your images. The tools calculate image statistics, graphically display characteristics of an image, and provide a means of comparing two images.

**Step 1: Reading Individual Pixel Values.** Let's start by opening an image for evaluation. Open "ngc0891j.st6" in the Image Evaluation subdirectory. This is an image of NGC 891 taken by Tim Puckett using an ST6 CCD camera.

The most basic image evaluation capability is to be able to read the value of a specific pixel at a specific location in it. To do this, just move the cursor over the image to the point you wish to measure, and the Current Pixel frame in the Image Display Control window will display the X and Y coordinates of the pixel you are over, along with its value. If this were a color image, the red, green and blue values would also be displayed, while the PV indicates the luminance value.

**Step 2: Image Statistics.** Click on the *Measure|Statistics|Image* menu item. The Image Status window will be displayed. The information that appears in the Info window is also stored in the **AIP4Win** Data Log. The Data Log is a separate window of the **AIP4Win** program and keeps a running log of all the measurements you make using the measurement tools. It appears in the icon tray at the bottom of your screen when **AIP4Win** starts up. It can be minimized by clicking the minus (-) button in the upper right corner, and it will retire to the taskbar where it can be recalled later. This window is editable, it can be cleared, and it can be saved to a file. The tools that write to this window do so in a format that makes the contents easily parsed into a spreadsheet program such as Excel or Lotus. This permits you to perform further analysis on data you extract using **AIP4Win**. For now you can just minimize the Data Log. If you should accidentally close it, no problem; it will be restarted the next time it is needed.

The Image Status window provides a list of statistics measured from the currently active image. If another image is loaded, the Image Status window can be updated by clicking the **Update** button. Some processing operations will generate a new image and will automatically update the Info window.

The basic statistics measured from the image are the minimum and maximum pixel values present, and the mean, median, average deviation and standard deviation, and skew of the pixel values.

A second tab of statistics tells you what image and pixel dimensions **AIP4Win** thinks the image has. The pixel dimensions are fundamental to correctly displaying the image on the screen, and are very important in astrometry and photometry. **AIP4Win** recognizes the image dimensions of most of the popular CCD cameras and makes a best guess at the pixel dimensions based on the file type and the number of pixels on each axis. The Preferences dialog even has a tab that allows you to enter the pixel size for your own camera, if it is not one of the models

## Appendix C: Tutorials

**AIP4Win** knows about. You can override **AIP4Win**'s choice using the **Imager** tab on the Display Control window. Select your camera type using the drop-down list and click **Set Imager Parameters**, and the image will be automatically redisplayed with the correct aspect ratio. If yours is not listed, just enter the pixel size directly in microns. You can also edit the file “Cameras.ini” located in the directory where **AIP4Win** is installed. This file can be edited with any ASCII text editor to add your own camera, if it is not there already, or to remove cameras you are not interested in. Directions for editing the file are included in its file header.

Close the Info window now using the **X** button in the upper right corner.

**Step 3: Image Histogram.** Click on the *Measure|Histogram* menu item and the histogram of the image will appear. This is a graph of the pixel value versus the number of pixels of that value in the image. For convenience it also displays their minimum and maximum values, and as you pass the cursor over the graph, the number of pixels at each value is displayed, along with its percentage of the total pixel count. Note that due to the wide range of possible pixel values afforded by floating point data, this histogram is binned proportionally to the range of pixel values in the image, so that the value you see is not the actual pixel value, but rather the pixel value associated with the selected bin. The histogram for this image shows a peak around 6000, dropping off rapidly and then peaking again at a much lower level around 43800. This is a pretty typical shape for a deep-sky object, in which most of the pixels show the sky background, with a few bright stars at the upper end. Close the Histogram window now using the **X** button in the upper right corner.

**Step 4: The Profile Tool.** Click on the *Measure|Profile Tool...* and the Profile Tool window will appear. Take the mouse and press the left mouse button on one edge of the image and drag a line across the image and release the button. The view in this window will plot the relative pixel values crossed by the line you drew. This view represents a slice taken through the image. A pair of radio buttons controls how the profile view is scaled to fit its window. Try clicking on each one to see how it affects the display. This tool is useful for exposing detail which might otherwise be hidden in very bright or very dark areas of an image. It also gives a view into the structure of an object. Dismiss this window by clicking on the **Close** button.

**Step 5: The Star Image Tool.** This is a very basic tool that allows you to extract information about a star in an image, including the ability to determine the exact center of a stellar centroid to  $\frac{1}{100}$  of a pixel. Open it by clicking on the *Measure|Star Image Tool...* menu item. The Star Image Tool window will appear.

This is a good time to mention that when you have a preferred location where you wish to see a tool window appear when it is invoked, you can save that location by dragging the tool window to where you want it and clicking on the *Window|Save Current Tool Window Positions* menu item. The next time you invoke this tool it will appear in the same place. Try it by dragging the Star Image Tool window to the upper right corner of the screen and then click on the *Window|Save Current Tool Window Positions* menu item. Close the Star Image tool and then re-

invoke it from the menu. It will appear in the place where it was when you saved its position.

The Star Image Tool contains three tabs, **Result**, **Shape** and **Settings**. Select the **Settings** tab and you will see a trio of text boxes with associated up/down arrows. This is a good time to introduce you to the concept of a photometer window. These three textboxes and their associated up/down arrows control the radii of three concentric circles that will be drawn around any star you select in the image. The inner circle defines the aperture, a region containing light from the star plus the sky background. The annular (doughnut-shaped) region between the middle circle and the outer circle is used to measure the brightness of the sky background. To find the brightness of the star, **AIP4Win** totals pixel values in the aperture and then subtracts the contribution form the sky based on the light in the annulus.

To get accurate measurements, you must set the radius of the aperture so that it completely contains the image of the star you are interested in. For a CCD image with reasonably good sampling, a radius of 6 pixels for the central circle works quite well. The sky annulus must be larger than the aperture and include a reasonable number of sky pixels. Radii of 9 and 15 pixels work well for most images.

Move your cursor over a star in the image and click the left mouse button. A set of circles will appear. You do not have to click the exact center of the star; the program will locate it, determine its centroid, and position the circles automatically. Click on the **Results** tab, and the star's X, Y centroid coordinates appear in the **Position** box. This information is also written to the Data Log window for later use. Other data, such as the size, peak pixel value and brightness are also displayed and saved in the Data Log. Try selecting a series of stars one at a time. Change the radii and see how this affects the program's ability to detect and measure centroids. If you set the radii large enough to contain several stars, **AIP4Win** will compute the “center of mass” of the group of stars.

Click on the **Shape** tab, and you will see a plot of the star's profile. This plot is generated in the program by determining the centroid of the star and then sweeping radially around it, plotting the pixel values. It is a good way to assess the “cleanliness” of the star image. Also displayed on this tab is information about the star's elongation.

This tool is very simple, but it forms the foundation for the rest of the measurement tools. When you are finished, dismiss the Star Image Tool by clicking the **Close** button.

**Step 6: The Distance Tool.** This tool builds on the basic capabilities of the Centroid Tool, providing the ability to measure distances and angles between any two stellar centroids in an image.

Open it by clicking on the *Measure|Distance Tool...* menu item, and the Distance Tool window will appear. It contains several text boxes used to set up the measurements.

The first is the **Telescope Focal Length** text box. This is used to enter the focal length of the telescope used to acquire the image, and must be expressed in millimeters. This information is used to measure distances at the focal plane in mi-

## Appendix C: Tutorials

crons. The NGC 891 image was taken with a 12-inch  $f/10$  telescope having a focal length of 3,048 mm. Enter 3048 mm in the **Telescope Focal Length** text box.

The next text box is the **Parallactic Angle** text box and is used to indicate the deviation, in degrees, from true north for a column on the CCD image, increasing in the clockwise direction. In the case of this image, north is  $90^\circ$  to the left, so enter 90 in the **Parallactic Angle** text box. This value will be used to determine the position angle for pairs of centroids.

Now select a pair of stars in the image using the cursor. A circle will appear when the left mouse button is pressed, just as with the Centroid Tool. Select the upper of a pair of equally bright stars just above the center of the image, at  $X = 152$ ,  $Y = 71$  and click the **Star 1** button. Next, select the star at  $X = 149$ ,  $Y = 87$  and click the **Star 2** button. The Distance and Position Angle data will be calculated and filled in showing a distance of 16.97 pixels = 456.16 microns = 30.87 arc-sec. These data were calculated knowing the pixel size and the telescope focal length. The position angle is calculated at 278.56 degrees, based on knowing the parallactic angle. Type in a different value in the **Parallactic Angle** text box and see how the value for the position angle is recomputed.

You can click the **New Stars** button and select several pairs to experiment with. Every time the data are calculated, they are written to the Data Log for later use.

To dismiss the Distance Tool, click the **Close** button.

**Step 7: The Pixel Tool.** The last of the measurement tools we will be covering in this tutorial is the Pixel Tool. This provides a means for gathering the statistics of a selected region of interest (ROI) in an image. It forms the basis for a set of photometry tools we will cover in a later tutorial.

Click on the *Measure|Pixel Tool...* menu item to open it. The first thing you will notice is that there are two radii available, an outer radius and an inner one. Click anywhere on the image window and a single circle will appear with the default outside radius of 6 and inside radius of 0. Change the outside radius to 12 with the **Out** slider and the inside radius to 6 with the **In** slider. When you click on the image, a “bullseye” pattern will appear. In this case, the ROI is the area between the inner and outer radii. This is useful for gaining information about the background sky immediately surrounding a star. Center the bullseye on a star and click, and the Region Statistics frame will display the data for that region. Note that unlike the Centroid and Distance Tools, the cursor does not “snap” to the centroid of the star. The shape of the ROI can also be changed from a circle to a square, which can be useful for some types of measurements.

As an example for how to use the Pixel Tool, select a circular region of outside radius 6, inside radius 0, centered on  $X = 305$ ,  $Y = 107$ . It shows a mean pixel value of 6024.23 and a standard deviation of 46.94. The signal-to-noise ratio (SNR) is defined as the mean divided by the standard deviation, giving an SNR of 128.34 for this specific region of the image. Other regions will tend to give different results, depending on the how much light struck the pixels there.

When you are finished, dismiss this window by clicking on the **Close** button.

**Step 8: The Data Log.** Examine the contents of the Data Log window and see how all the measurements you made were recorded there. If you wished to import the contents of this log to a spreadsheet, you would just click the *Save to File* menu item in this window and a dialog box would appear asking you for a file name. The file is saved as ASCII text and can be edited with any text editor.

To import your data into an Excel spreadsheet, load the file into Excel as fixed-width ASCII text. The labels will appear in one column and the data will appear in the next column. It is probably easiest to save the Data Log after each type of measurement and then clear it before adding new data of a different type. This makes it simpler for the spreadsheet to parse it into columns, since each tool outputs a different type of data.

This concludes the general image evaluation tutorial. In later tutorials we will cover specific types of measurements used in photometry and astrometry.

## C.4 Astrometry

This tutorial demonstrates how to use **AIP4Win** to make precise positional measurements of objects in your CCD images. The files for it are located in the Astrometry subdirectory of the Tutorials directory on the CD-ROM.

In recent years a number of ever more accurate astrometric databases has become available, making it easy to precisely determine the celestial coordinates of objects in electronic images. **AIP4Win** supports several CD-ROM databases including:

- The Hubble *Guide Star Catalog (GSC)*, the first of the CD-ROM databases to become widely available.
- The *United States Naval Observatory A2.0* database, which was created by scanning the Palomar Sky Survey (POSS) plates, and precisely locating the stars on each plate using data from the Hipparcos satellite.
- The subsampled version of the *A2.0* database, the *SA2.0*.
- The *MegaStar* CD-ROM, which is based on the *GSC*.

Knowing the coordinates of the center of your image, you can generate an overlay of known stars from any of these databases and, once the overlay is registered to the image, determine the precise coordinates and stellar magnitude of any object in it.

You can also use *MegaStar* to find the coordinates of your image and save a list to disk of all the stars in its immediate vicinity for use as an overlay.

In this tutorial, you will load a CCD image in which there are several asteroids, and then load a pre-generated reference star data file corresponding to that image. You will then generate and align the overlay to it, select a set of reference stars from the overlay, and then determine the coordinates and magnitude of the asteroids on the image.

**Step 1: Load the Image.** If any images are currently open, click the

## Appendix C: Tutorials

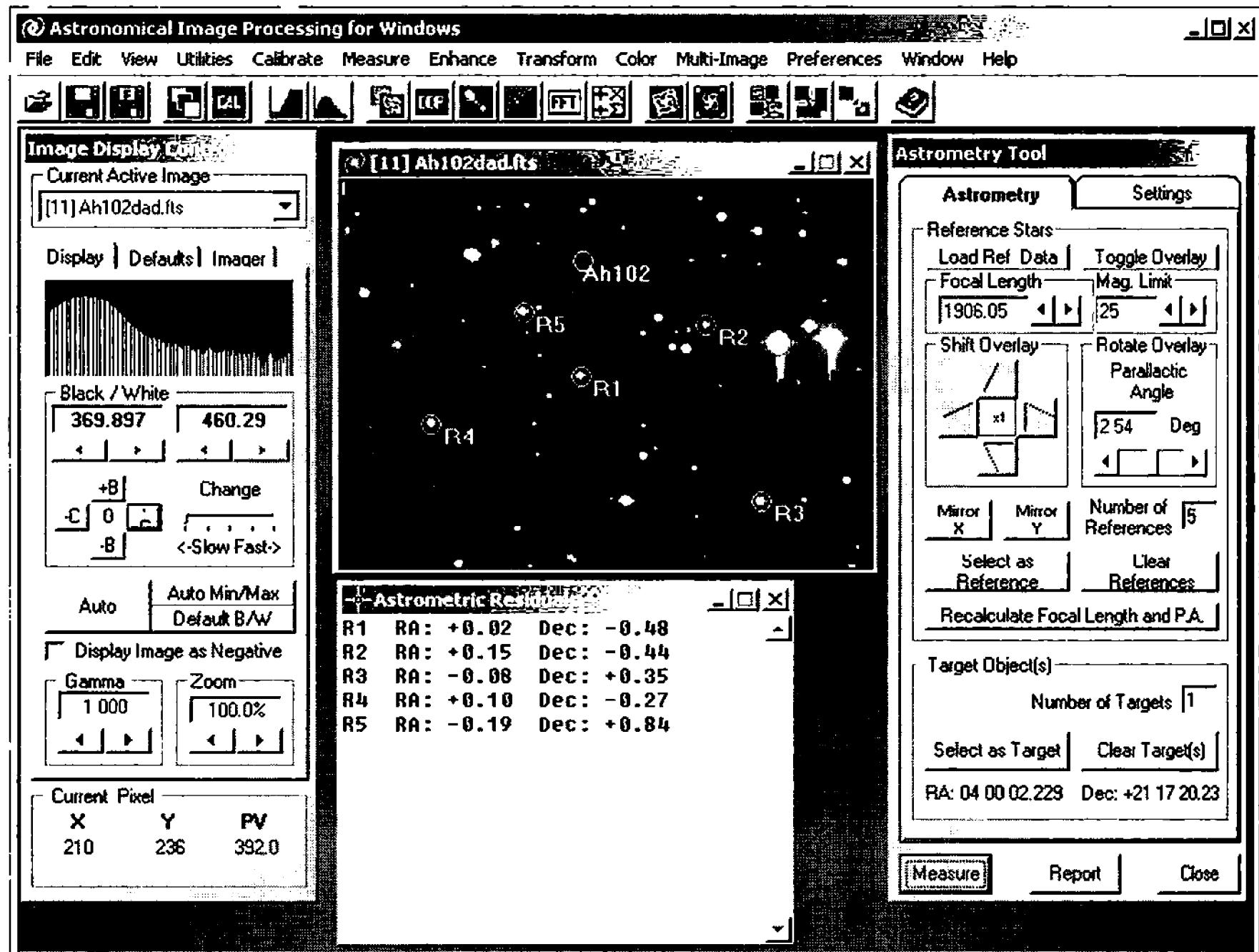


Figure C.4 Astrometry enables you to determine precise right ascension and declination coordinates for any object in a image. The first step is to link reference stars in your image to reference star coordinates. The next step is to designate a target, and the last step is to solve for the coordinates of the target object.

*Window|Close all Images* menu item to close them. You can also click the *Window|Close all open Tool Windows* menu item to close any tools you may have left open. These features are handy for quickly clearing your screen of old tools and images.

Load the image “Ah102dad.fts.” Click the *Measure|Astrometry...* menu item and the Astrometry Tool will appear. Drag it to a convenient location on the screen.

**Step 2: Create an Overlay.** For this tutorial, several reference star files have been prepared for this image, one using the Hubble GSC (via *MegaStar*) and one using the *USNO SA2.0*. The first one you will load will be the *GSC* version.

Before you load the reference data, you want to input the telescope focal length in mm in the **Focal Length** text box. This will scale the generated overlay so that it matches the scale of the image. For some images you may also need to use the **Mirror X** and/or **Mirror Y** buttons to reorient it so that north is to the top (roughly) and east is to the left, to match the overlay.

The telescope used to make this image had a focal length of approximately 1,909 mm. Input this value in the **Focal Length** text box. Later, the software will determine a precise value for the focal length.

It is not necessary to mirror this image about either axis.

Now you can create the overlay. Click on the **Load Ref Data** button and the Select Overlay window will appear. At the top you will see a set of five radio buttons corresponding to the different database types that **AIP4Win** can read. In addition to the four CD-ROM databases, there is a **Reference File** button. Click it and click **OK**, and a dialog box will appear. Navigate to the Astrometry subdirectory, select the file “Ah102.ref” and click **Open**; the reference data file will be loaded and an overlay will be generated.

To use an astrometric catalog stored on CD-ROMs, select the appropriate database; and input the Right Ascension and Declination of the field center along with the size of the field. **AIP4Win** will prompt you for the correct CD-ROM and create a list of all the stars in the selected region brighter than the magnitude limit set in the **Magnitude Limit** text box. You are then prompted to save the generated list—that is how the reference data files for this tutorial were generated. Chapter 9 provides information on obtaining these databases.

Click the **Toggle Overlay** button until the overlay appears. It will appear as a collection of red circles, with a blue cross marking the center of the field.

**Step 3: Align the Overlay.** For this image, the overlay needs to be shifted to line up with the stars. Look for a conspicuous trapezoid of stars just above and to the right of center. Just above and to the left of the trapezoid, you will see a set of four circles in a matching trapezoidal pattern.

In the Shift Overlay frame you will see a set of four arrowed buttons grouped around a center button marked **X1**. Click the **X1** button, and it will show **X10**; this is the rate at which the arrow buttons will shift the overlay. Click the left arrow button twice and the down arrow button twice; the overlay will shift so that the trapezoid of circles nearly overlays the trapezoid of stars. Click the **X10** button so that it shows **X1**, and refine the position with the arrow keys until the overlay matches the stars as closely as possible.

At this point you will notice that the trapezoid matches the overlay, but many of the other stars do not. This is because the image is slightly rotated with respect to the overlay. Correct this by typing 2.5 in the **Parallactic Angle** textbox, or use the associated slider. A few clicks of the **Shift Overlay** arrow buttons and the overlay and the image will come into perfect alignment. Once you identify the reference stars in the field, the overlay can be brought into perfect alignment automatically.

**Step 4: Select a set of Reference Stars.** Select the circled star in the center of the image with the mouse and click the **Select as Reference** button. A green circle will surround the star with the label **R1** below and to the right of it. This marks the star as a reference for measurement. When selecting a reference star, try to pick a clean star image that will give a good centroid. Avoid selecting stars in the middle of nebulosity or ones that show blooming trails. Clean star images will yield more accurate position and brightness measurements.

Select two more reference stars. Use the top right one in the trapezoid and the bottom right one in the image. Remember only to select stars that appear on the overlay. If you attempt to select a star that doesn’t appear on it, an error mes-

## Appendix C: Tutorials

sage will appear, prompting you to try again.

Once you have selected three reference stars, the **Recalculate Focal Length and PA** button will be enabled. This will refine the overlay alignment to sub-pixel accuracy. Click it now. The telescope focal length will be updated to a precise value near 1907 mm, and the parallactic angle is updated to a precise value near 2.46 degrees. (This is a handy way to determine the focal length of your telescope very accurately, by the way.)

Select a few more reference stars. You will need at least four in order for **AIP4Win** to calculate a set of residuals associated with the calculated plate constants for the CCD image. You will see that as soon as a set of plate constants is calculated, the RA and Dec of the cursor position are updated at the bottom of the Astrometry Tool window whenever the cursor is moved over the image. As you add more reference stars the accuracy of the solution increases.

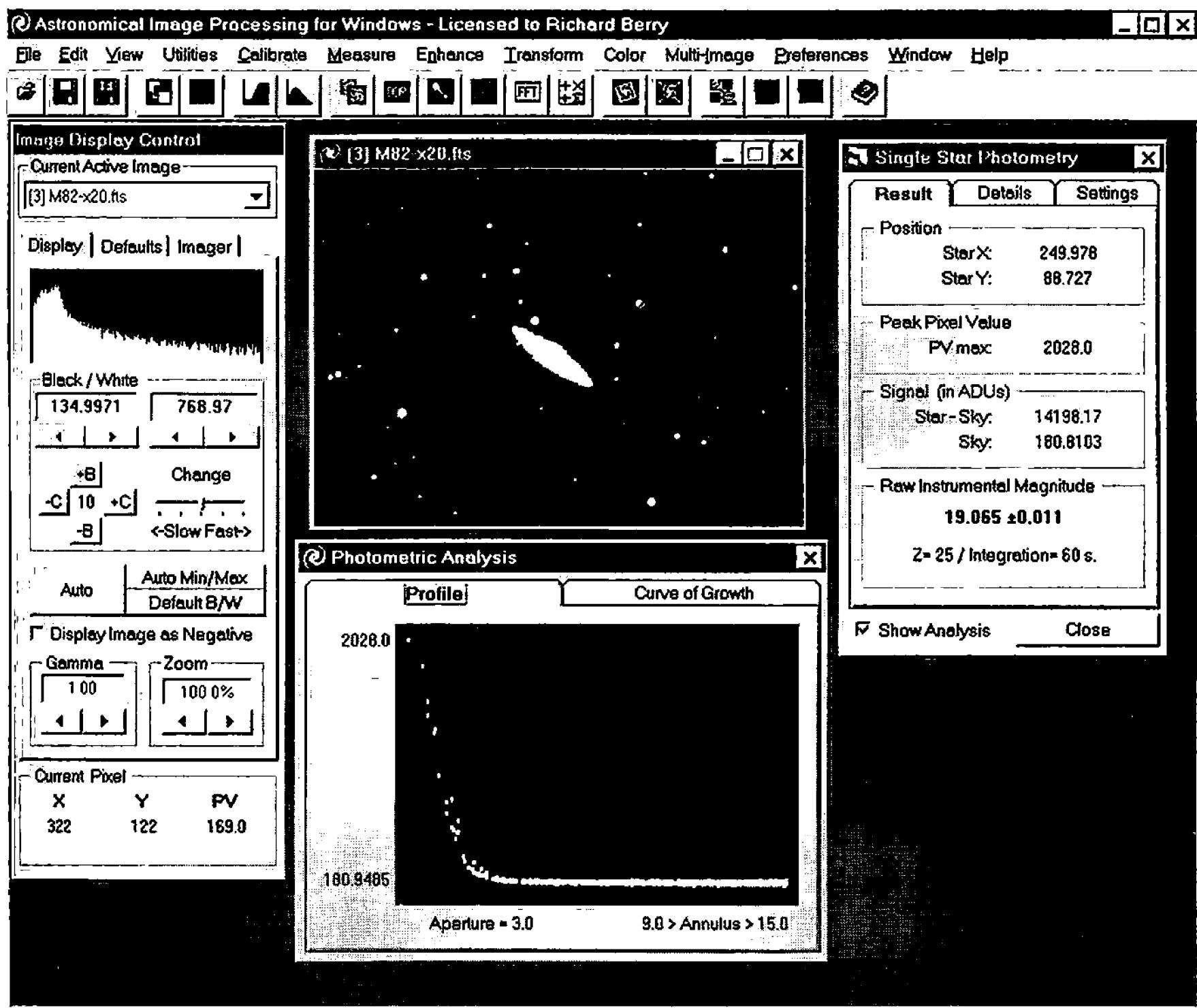
**Step 5: Select the Object to be Measured.** In this image there are three asteroids. One of them, ah102, is at X = 115, Y = 45. Click on this faint object with the left mouse button. Click the **Select as Target** button, and the Enter Target Name window will appear. Enter the name “Ah102” for the target and click **OK**; a yellow circle will appear around the target with the label “Ah102” below and to the right. If the circle centers on the brighter star below this object, click the **Clear Target(s)** button and try again. It might help either to reduce the search radius using the **Search Radius** slider on the **Settings** tab, or position the circle so that it includes the asteroid but not the star. Your cursor position doesn’t have to be accurately centered; **AIP4Win** will find the centroid of the object inside the search radius and align to it automatically. If two objects are inside the search radius, **AIP4Win** may attempt to center on their common “center of mass.”

**Step 6: Generate a Report.** Now click the **Measure** button. The Astrometric Residuals window will appear, showing the residuals from the calculated plate constants. If it is not already visible, open the Data Log window from the taskbar. In it you will see the calculated RA and Dec of the object, the error for both RA and Dec, and the object’s magnitude.

If you wish to generate a report, click the **Report** button and the Enter Astrometry Report Data window will appear. You are presented with two options here: a verbose (multi-line) report, or a single-line report in the form required by the Minor Planet Center (MPC). Select the verbose format, then type in an object name and an arbitrary time and date. Click **OK**, and you will be prompted for a location to save this report. Choose a convenient place (not on the CD-ROM!), and click **Save** to save it.

You can examine the report if you click the *File|Open Log (text) File...* menu item. This will open a file select dialog. Select the file you saved and it will be displayed in a small text window. You can also examine or edit this file with any ASCII text editor.

You can clear the reference stars and targets and select a new overlay if you want to experiment further. Try using the file “ah102.gsc.” It was generated from the *USNO A2.0* database. This is a phenomenally deep database with more than



**Figure C.5** The photometry tools enable you to measure the brightness of stars. The Single Star Photometry tool is a basic tool that determines the “raw instrumental magnitude” of stars—one star at a time. The image profile and curve of growth help you assess the suitability of a star for measurement.

half a billion stars in it. You will see that the overlay includes nearly every object in the image, with the exception of the three asteroids.

This concludes the astrometry tutorial.

## C.5 Photometry

In this tutorial, you will learn how to use **AIP4Win** to measure the magnitude of stars in a CCD image.

We will explore three photometry tools:

- Single-Star Photometry determines the brightness of a star by summing its light and subtracting the sky background light.
- Single-Image Photometry determines the magnitudes of two or more stars and gives the magnitude *difference* between them.
- Multiple-Image Photometry is an automated tool that measures either raw instrumental magnitudes or differential magnitudes from a series of images.

## Appendix C: Tutorials

Each of these tools builds on the others to provide everything you need to perform some pretty ambitious projects.

### C.5.1 Single-Star Photometry

From the Photometry subdirectory under the Tutorials directory, load and display the image “M82-x20.fts.” If you view the FITS header for this image, you will see that it is a stack of twenty 60-second exposures of the galaxy M 82.

**Step 1: Open the Single-Star Photometry Tool.** Open the Single-Star Photometry Tool by clicking the *Measure|Photometry|Single Star...* menu item. On the **Settings** tab, you will see three sets of text boxes and up/down controls for setting the star aperture and the inner and outer radii of the sky annulus. Set the aperture radius large enough to capture all the light from the star you wish to measure.

**Step 2: Test the Suitability of a Star for Photometry.** You can test a star to determine its suitability by selecting one and clicking on it. Try it by clicking on the star at X = 250, Y = 89 in the “M82-x20.fts” image and then checking the **Show Analysis** checkbox.

This will cause a tabbed window to appear. (Like all the other tool windows, you can save their start-up positions by clicking the *Window|Save Current Tool Window Positions* menu item.) Each tab displays a graph.

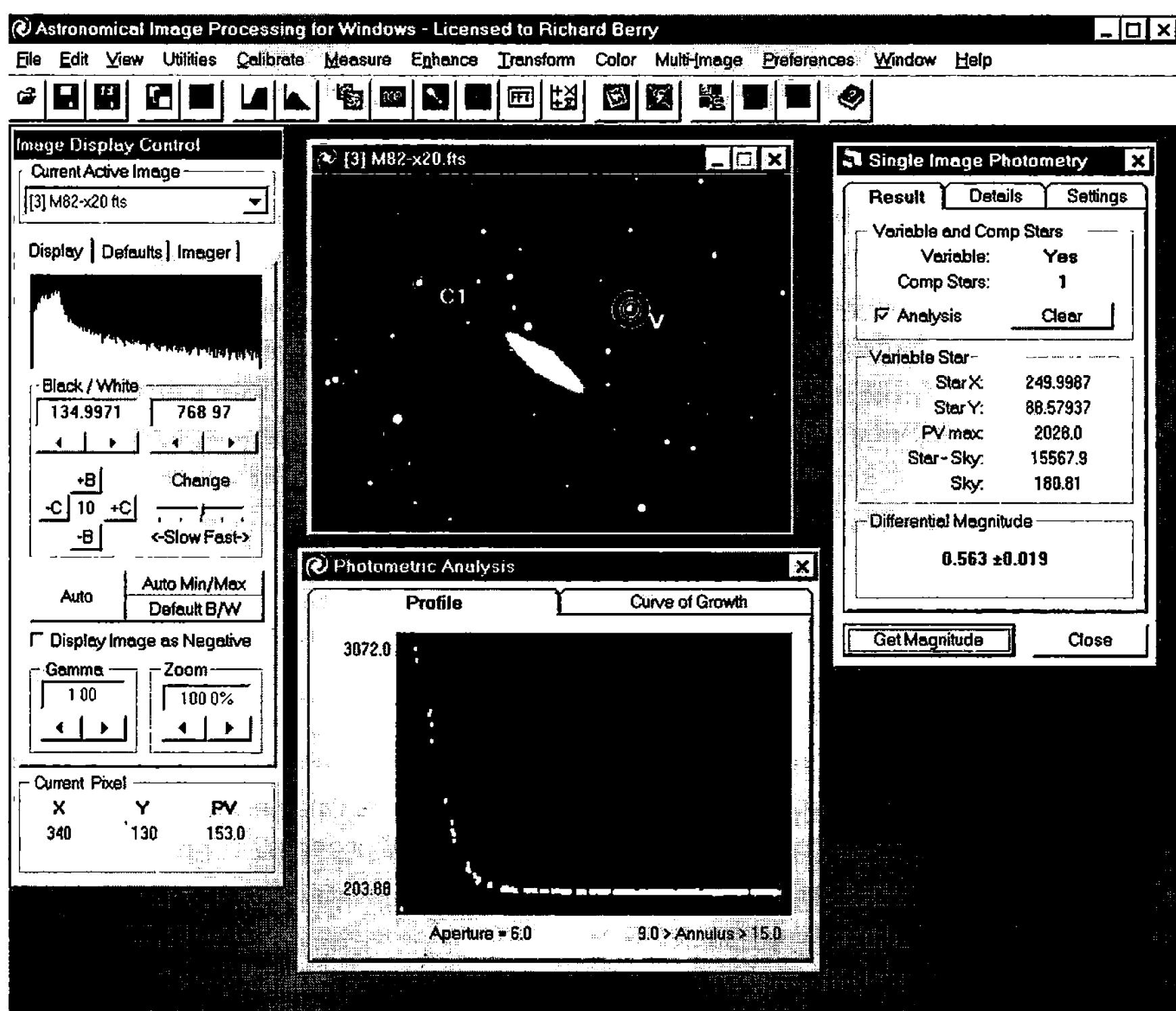
The Profile tab shows a brightness cross-section of the selected star, with the values of individual pixels plotted against their distance from the center of the star image. Ideally, what you want to see here is a smooth curve falling off from a peak at the center of the star image. The star we selected displays just this type of profile. You can estimate the half-width half-maximum of the star image by noting the radius at which the profile reaches half its peak value.

The Curve of Growth tab plots magnitude versus the star aperture radius. This is useful for determining a good setting for the star aperture. For this image, the default radius of 6 pixels works well. A level region on the graph indicates that the aperture captures all the light from the star. The inner and outer annulus radii should be set to about twice the value of the inner, unless it is necessary to make it smaller to avoid including other stars. The default values of 9 and 15 pixels work well with this image.

**Step 3: Input the Image Data.** Below the radii controls on the **Settings** tab is a frame containing text boxes for entering the Integration Time, the Zero-Point Magnitude, Readout Noise, Gain, and Dark Current. Although the total integration time for this image was 1200 seconds (20 exposures of 60 seconds each), the images were averaged (i.e., divided by 20) to improve the signal-to-noise ratio—so the effective integration time is 60 seconds. Enter this value in the **Integration Time** text box.

The zero point depends on your telescope, filters, and CCD camera. It represents the magnitude of the sky background, and usually turns out to be a value between 15 and 25. For this image, the default value of 25 will suffice.

The Readout Noise, Gain and Dark Current are dependent on the CCD cam-



**Figure C.6** The Single Image Photometry tool compares the brightness of a variable star against the brightness of one or more non-variable comparison stars. Because it determines only the difference between stars, results from differential photometry are relatively unaffected by atmosphere and thin clouds.

era used to make the image. For the purposes of this tutorial, we will leave them set to their default values.

**Step 4: Select and Measure a Star.** Select the **Result** tab. Now click on the star at X = 250, Y = 89. You will see the results of measuring the star. The Measurement Log is updated as well, allowing you to export these values to a spreadsheet later. The raw instrumental magnitude is the sum of pixel values in the star image corrected for the sky background converted to magnitudes—not the “real” magnitude of the star.

Chapter 10 explains how astronomers convert raw instrumental magnitudes into accurate magnitude measurements. The **Details** tab displays information used to determine the standard deviation of the magnitude determination.

Try selecting and testing some other stars in this image. For an example of a star not well suited for photometry, try selecting one in the halo of the galaxy, or one that has another star within a few pixels of it. You will see that its profile will have a lot of extra light around it, adding uncertainty to the measurement of the star’s brightness.

## Appendix C: Tutorials

Finally, check the Data Log; you will see that it has faithfully recorded the position and magnitude of each of the stars you have clicked on.

### C.5.2 Single-Image Photometry

The Single-Star Photometry Tool provides a means of measuring the light from a single star and removing the sky background from it. You could use this tool to take measurements of a variable star and a star of constant brightness. By comparing their values you can then detect tiny changes in the brightness of the variable. This is *differential* photometry—the difference in brightness between two stars. Differential photometry is a powerful technique because thin clouds, haze, and atmospheric extinction may dim the stars, but their difference is barely affected.

**Step 1: Open the Single-Image Photometry Tool.** Close the two graphs, if they are still open, and close the Single-Star Photometry Tool. Click the *Measure | Photometry | Single Image...* menu item and the Single-Image Photometry Tool window will appear. Like the previous one, this tool provides a **Settings** tab that allows you to set the photometer radii.

**Step 2: Select the Stars.** Select the **Result** tab. Open the M82 image if it is not already open, and click on any star in the field to select the variable star. You will see that in the **Variable and Comp Stars** box, the **Variable** setting now reads “Yes,” and the Raw Instrumental Magnitude box gives a magnitude reading for the star. Click the **Clear** button, and the setting will read “No.”

Click on the star at X = 250, Y = 89. Next choose the first comparison star. Click the star at X = 83, Y = 71. You will now see that the **Comp Stars** setting reads “1.” Each time you will see a display of the raw instrumental magnitude.

Click the **Get Magnitude** button, and **AIP4Win** will calculate the differential magnitude. The **Differential Magnitude** box shows that the variable is 0.563 magnitudes fainter than the comparison star.

To improve the quality of their data, astronomers select additional comparison star. As you add more comparison stars—clicking the **Get Magnitude** button as you select each one—note that the standard deviation decreases, meaning you are getting a more precise magnitude reading.

The information you obtain from these measurements is logged in the Data Log, but only when you click the **Get Magnitude** button. In normal practice you would select the variable star, select a series of comparison stars, and then click the **Get Magnitude** button to record one differential photometry reading.

If you wanted to monitor a supernova as its light faded, you could take a new image of it every clear night. You could then use the Single Image Photometry Tool to measure the difference between the supernova and a comparison star (V – C1) in each image. Over a period of a few months, you would obtain the supernova’s light curve.

When you are finished, dismiss all the tools by clicking on the *Window | Close all open Tool Windows* menu item, and close the M82 image.

### C.5.3 Multiple-Image Photometry

Though the Single-Image Photometry Tool provides a fast and accurate means of measuring stellar magnitudes, the real power of a computer comes into play when you have a series of images taken over time of a variable star and want to automatically extract the brightness data to create a brightness curve. **AIP4Win** provides this capability via the Multiple Image Photometry Tool.

For this tutorial, unlike the others, the images can be found in the Sullivan subdirectory of the Images directory on the CD-ROM. This collection of images was obtained on April 3, 2000, by Phil Sullivan. It consists of a series of images taken two minutes apart using a Cookbook 245 CCD camera. During this time, the eclipsing binary system Z Draconis reached minimum light. This directory also contains a composite master dark frame and a composite master flat frame.

**Step 1: Set Up Image Calibration.** The images in the Sullivan directory are raw CB245 image frames that need to be calibrated before any useful photometric information can be extracted from them. Click on the *Calibrate|Setup...* menu item and select the standard calibration protocol. Use the file “DKAVE.FIT” for the dark frame and “MSTRFLT6.FIT” for the flat frame.

**Step 2: Invoke the Multiple Image Photometry Tool.** Invoke the tool by clicking the *Measure|Photometry|Multiple Image...* menu item.

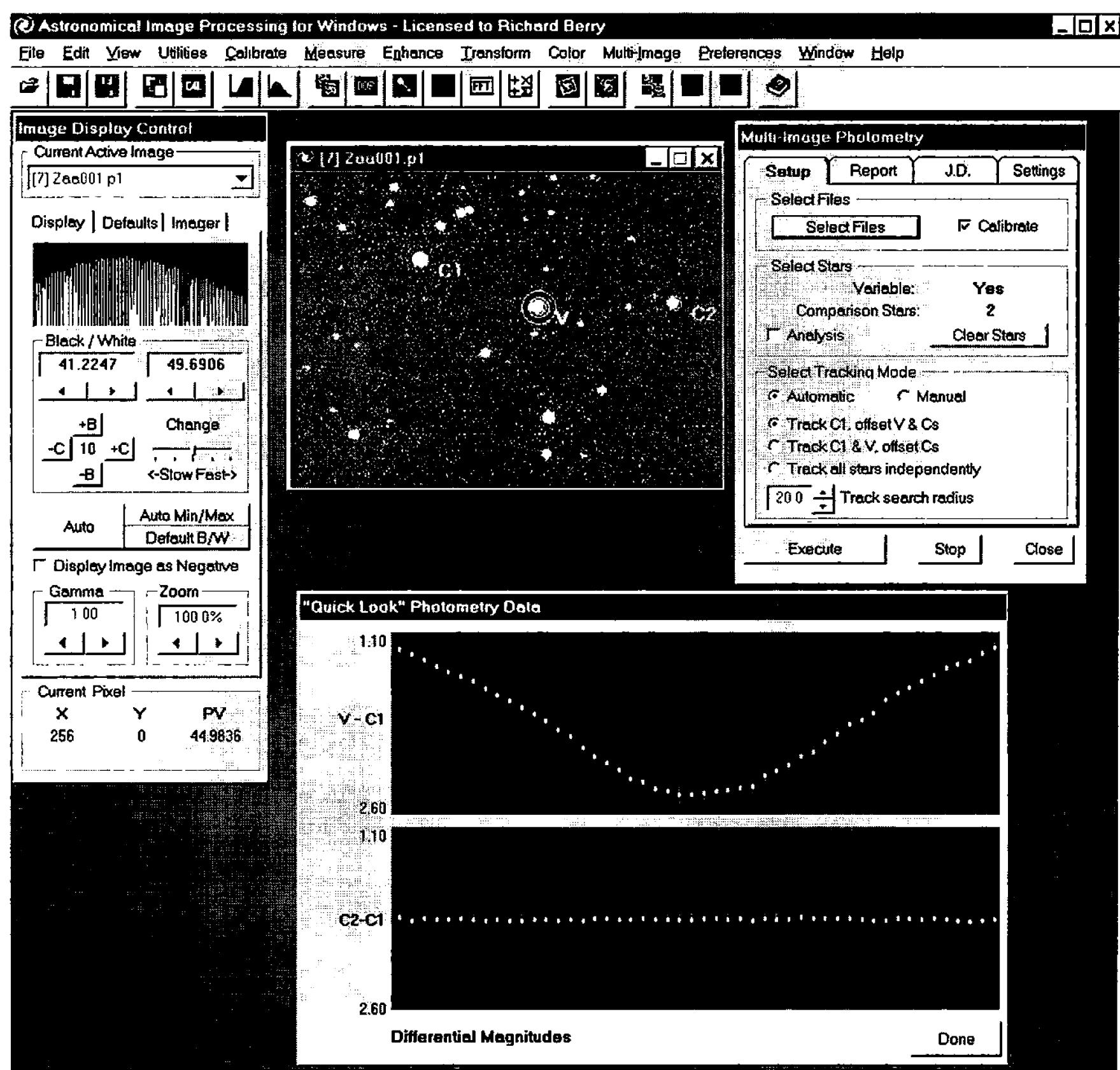
**Step 3: Select the Image Set to be Measured.** Check the **Calibrate Images** checkbox. Click the **Select Images** button, and then select all of the files in the Sullivan directory that have a “.p1” extension. It is easier if, when you have invoked the Select Files dialog, you use the **Files of Type** select box and select “Any CB245 File (\*.pa, \*.p1)” from the list. Then click on the first file in the directory, shift-click on the last file in it, and then click the **Open** button. This will select the “.p1” files but ignore the “.fts” files in the directory. The images are named such that they are in alphabetical order, in order of time. **AIP4Win** sorts the images in alpha-numeric order; it is a good idea to name your datasets this way so as to preserve the order of your data.

**Step 4: Select the Photometer Radii.** Once you have opened the image set, the first image will be displayed in an image window. Click anywhere in it and a “bullseye” representing the photometer aperture will be drawn. You can use the **Radii** controls on the **Settings** tab to adjust the aperture, just as in the previous tools. For these images, the default values of 6, 9 and 15 work well.

If your were doing photometry on stars that were trailed or out of focus, you would adjust the aperture to capture all the starlight, and set the annulus radii to exclude any of the star’s light. It is a good idea to scan through your images using the *File|Sort Images...* tool before you run the Multi-Image Photometry Tool, to make sure you spot any bad images ahead of time. These problems do not render an image unsuitable for photometry, but it is necessary to adjust the radii.

**Step 5: Select the Search Radius.** The Multiple Image Photometry Tool performs individual measurements on a series of images of the same star field. Once a V (variable) star and set of C (comparison) stars has been identified, the

## Appendix C: Tutorials



**Figure C.7** The Multi-Image Photometry tool can perform differential photometry on hundreds of images, one right after another, automatically. With this tool, amateurs have detected planets orbiting other stars by measuring the slight dip in light when the exoplanet passes in front of its star.

tool will attempt to track these stars from image to image. As there is the possibility that the field will shift between exposures, it is necessary to identify a radius, centered around the **C1** star, that will be searched to find the star centroid on each subsequent image in the set. This is done using the **Track Search Radius** control on the **Tracking** tab. For this image set, a value of 20 pixels accommodates the rather large drift between exposures without locking onto a neighboring star by mistake.

**Step 6: Select the C and V Stars.** Now you can select your V and C stars. For this tutorial you need to select the star at X = 215, Y = 102 (Z Draconis) for the V star, X = 112, Y = 65 for the first comparison star, **C1**, and the one at X = 335, Y = 98 for a second comparison star, **C2**. When selecting stars, the first one you select is always the variable, and the rest are numbered comparison stars. When you select each one, a bullseye is drawn showing the inner, middle and out-

er radii of the photometer aperture and a label, either **V** or **C $n$**  (where  $n$  is the number of the comparison star).

You can delete a comparison star by clicking on it a second time. The star will be removed from the list, and the other comparison stars will be renumbered. However, if you click the variable star a second time, the entire list will be cleared. You can also clear the list by clicking the **Clear All Stars** button on the **Setup** tab.

**AIP4Win** allows you to select up to 32 comparison stars, and will track each one. If a comparison star drifts out of the image, an error message is displayed at the conclusion of the measurement run.

**Step 7: Set the Tracking Mode.** The **Tracking Mode** radio buttons on the **Tracking** tab allow you to select the tracking mode. If the images are well tracked, select **Automatic** tracking. As the stars drift from one image to the next, **AIP4Win** finds them by searching one target search radius in all directions from the star's location in the previous image.

If the tracking error between images is greater than the tracking radius, select the **Manual** tracking mode. Manual tracking is cumbersome—it really pays to have a telescope with a good clock drive—but it allows you to do good photometry even though your telescope has poor tracking.

Although the star images wander quite a bit, automatic tracking will follow the stars in this data set, so click the **Automatic** radio button.

**Step 8: Set the Target Tracking Mode.** The **Target Tracking Mode** radio buttons control whether the V and C stars are tracked individually, or whether the C1 star only is tracked, using a fixed offset to identify the V and remaining C stars. The **Track C1** mode is for fixed targets such as variable stars. The **Track C1 & V** mode is for photometry of moving objects such as comets and asteroids. The **Independent** mode tracks every star individually, should that ever be necessary.

For this tutorial we are following the light curve of an eclipsing binary star, so the **C Track** mode should be selected.

**Step 9: Start the Photometry Operation.** Start the Multiple Image Photometry operation by clicking the **OK** button. You will be prompted for a location to save the photometry data that will be generated. Choose a directory and file name on your hard drive (not on the CD-ROM) for this purpose.

Each successive image in the set will now appear on the screen, the V and C $n$  stars will be found and marked, and their relative magnitudes logged. When the operation has finished, a graph is displayed showing data plots for the differential magnitude (V – C1) and for the difference between the first two comparison stars (C2 – C1), to provide a quick check of your data.

In this exercise, you have measured the *difference* between the variable and comparison stars to find the moment when the variable was in maximum eclipse.

**Step 10: Import the Data into a Spreadsheet.** When the tool has finished, you will be left with a report file in the Data Log that, like the output of other tools, can be imported into most of the popular spreadsheet programs.

Chapter 10 explains how astronomers use standardized color filters and care-

## Appendix C: Tutorials

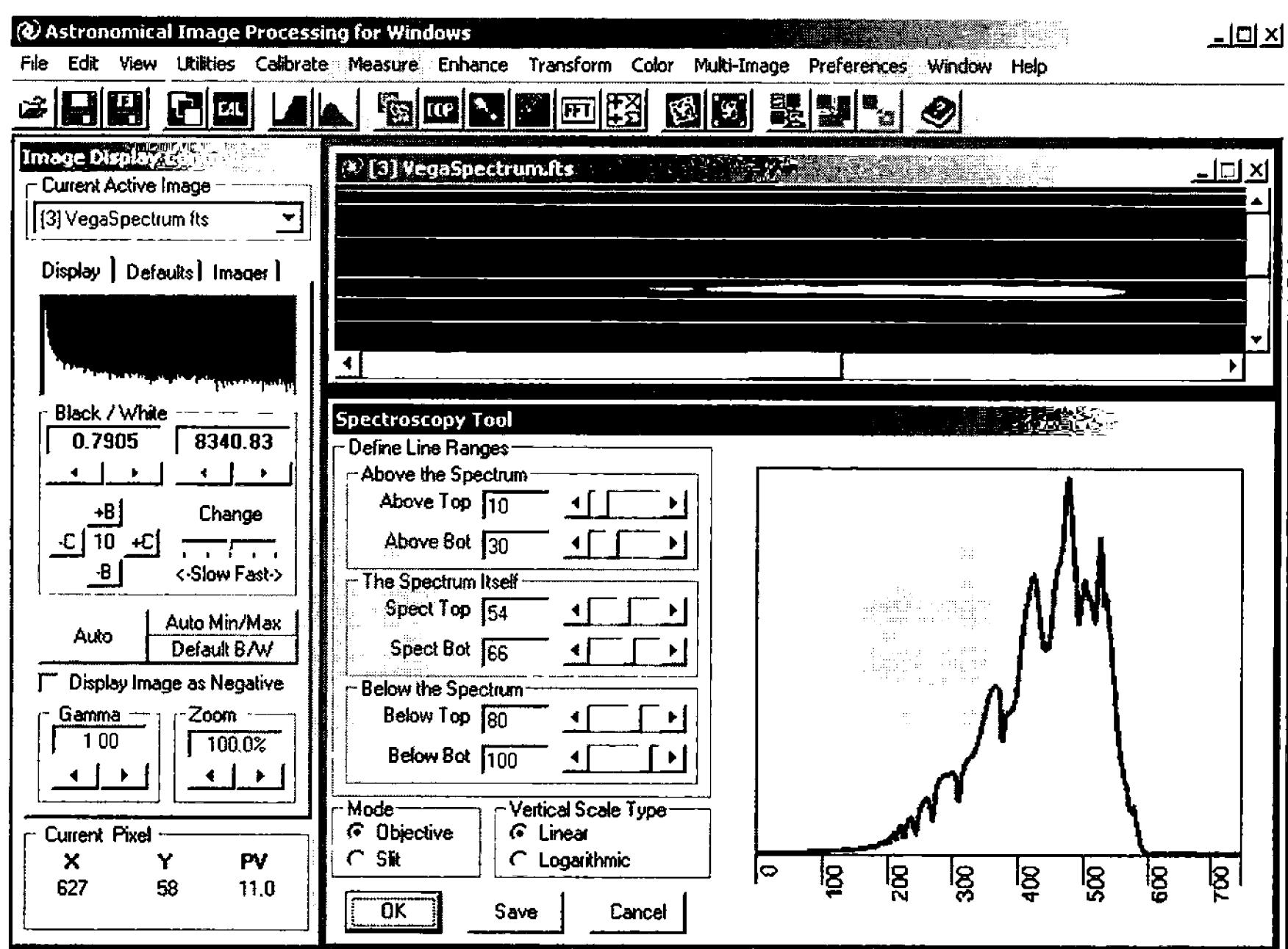


Figure C.8 By analyzing the starlight disbursed into wavelengths by a prism or diffraction grating, astronomers can tell the temperature and composition of distant stars. Seen here, the Spectroscopy tool is converting an enigmatic smear of light into a graph displaying the characteristic pattern of hydrogen atoms.

fully calibrated standard stars to measure stellar magnitudes.

## C.6 Spectroscopy

This tutorial demonstrates how to use **AIP4Win** to extract the spectral curve from a star image obtained using an objective prism. The same techniques can also be used with a slit or fiber-fed spectrograph.

The file we will use is called “VegaSpectrum.fts” and can be found in the Spectroscopy subdirectory of the Tutorial directory on the CD-ROM. This file is a spectrum of the zero-magnitude type A0V star Vega in the constellation Lyra. This image was acquired using a 6-inch objective prism mounted on a 6-inch  $f/5$  Newtonian telescope using a CookBook 245 CCD camera in 756-wide mode. The file has already been calibrated via dark frame subtraction and flat-fielding; it has been cropped to save space.

**Step 1: Load the Spectrum Image.** Open the “VegaSpectrum.fts” image and adjust the Display Control to best view the spectrum. The image will appear as a streak stretching across the middle of the display, with the wavelength increasing from left to right.

**Step 2: Invoke the Spectroscopy Tool.** Click on the *Measure | Spectroscopy*

*py... menu item to invoke the Spectroscopy Tool. Position the spectrum image and the Spectroscopy Tool so that both are visible on the screen.*

**Step 3: Select the region of the image containing the spectrum.** The Spectroscopy Tool has a set of text boxes and sliders used to define three regions on the image. These regions are:

- the background sky above the spectrum,
- the region of the spectrum itself, and
- the background sky below the spectrum.

The Spectroscopy Tool performs a median over the areas identified as sky background and subtracts them from the spectrum in order to remove the spectral contribution of those areas from the stellar spectrum.

With an objective spectrum, the median is taken over the entire sky background, since it may contain light from numerous other sources, especially in the case of a crowded star field.

In the case of a slit spectrograph, the median is taken on a column-by-column basis, and subtracted from the spectrum only in the corresponding column.

In the **Spectrum Itself Box**, move the **Spect Top** slider until the **Spect Top** text box shows a value of 54. You will see a line slide down the image until it just sits on top of the spectrum region. Move the **Spect Bot** slider until the **Spect Bot** text box shows a value of 66. These two lines now define the region of the image containing the spectrum.

Move the remaining sliders to the following positions:

**Above Top:** 10

**Above Bot:** 30

**Below Top:** 80

**Below Bot:** 100.

These two regions contain sky background with no visible stellar contribution. Note that it is not always necessary to set up the background sliders, especially if all you want to do is to quickly view a spectral curve. They are provided principally as a means of calibrating the spectrum to remove the sky background.

**Step 3: Select the Spectrometer Mode.** As mentioned above, the Spectroscopy Tool handles the sky background removal for objective and slit spectrographs in different ways. For this image you should select the **Objective** mode.

**Step 4: Select the Vertical Scale Type.** You have the option of displaying the amplitude on either a **Linear** or a **Logarithmic** scale. For now, select the **Linear** one. You can experiment with a log display later.

**Step 5: Generate the Spectral Curve.** Click the **OK** button and the spectrum will be displayed.

**Step 6: Save the Spectrum to a File.** Click the **Save** button, and you will be prompted to save the spectrum data to a file. This file is “spreadsheet friendly,” and can be easily parsed into any standard program such as Microsoft Excel or Lotus 123.

## Appendix C: Tutorials

Once the spectral data have been loaded into a spreadsheet, you can use the same steps listed above to load the spectrum of a calibration source taken with the same spectrograph. To calibrate the wavelengths of the unknown spectrum, you must calibrate the wavelength curve of the spectrograph against a set of known spectral lines. Although grating spectra are nearly linear, they are not perfectly so; and prism spectra are strongly nonlinear.

### C.7 Image Enhancement

This tutorial demonstrates a few of the image enhancement tools in **AIP4Win**. In it and those that follow, you will learn how to use these tools to extract detail from otherwise bland images. More than just producing “pretty pictures,” these techniques can be used to enhance details and show structures that, because of their low contrast, would otherwise be invisible. Later tutorials show you which tools and processes to use on deep-sky and planetary images.

You may have noticed that the arrangement of the menus and the toolbar, the book itself, and to some extent these tutorials, is such that the flow moves through the following steps:

1. Bring an image into the program,
2. view information contained within the image file header,
3. calibrate the image,
4. make measurements on the image, and
5. enhance the image, including geometric transformations.

This is not accidental. This flow of operation is necessary to preserve scientifically useful information. Up to this point in the tutorials we have not made any changes to the information contained in the image other than to remove noise and uneven illumination effects caused by the optical system or sensitivity variation across the CCD detector.

The types of operations you will perform in this tutorial will, for the most part, modify the data in a non-linear manner, rendering them useless for photometric purposes. You will also be able to modify the spatial distribution of the data, reducing their astrometric validity. It is for this reason that we covered the measurement capabilities first, in keeping with this flow.

The types of operations you will exercise in this tutorial are some of the most fun capabilities of **AIP4Win**. With these enhancement tools you can make pretty pictures as well as display the hidden structure of astronomical objects you have imaged with your CCD camera. These operations work well on scanned photographs as well, and provide capabilities that just aren’t possible in the darkroom.

A final note before starting: preserve your original files. You cannot “unenhance” a processed image and recover the original data. Now, let’s get started.

#### C.7.1 Brightness Scaling

This is the most basic of image enhancement functions, and is the most frequently

## Section C.7: Image Enhancement

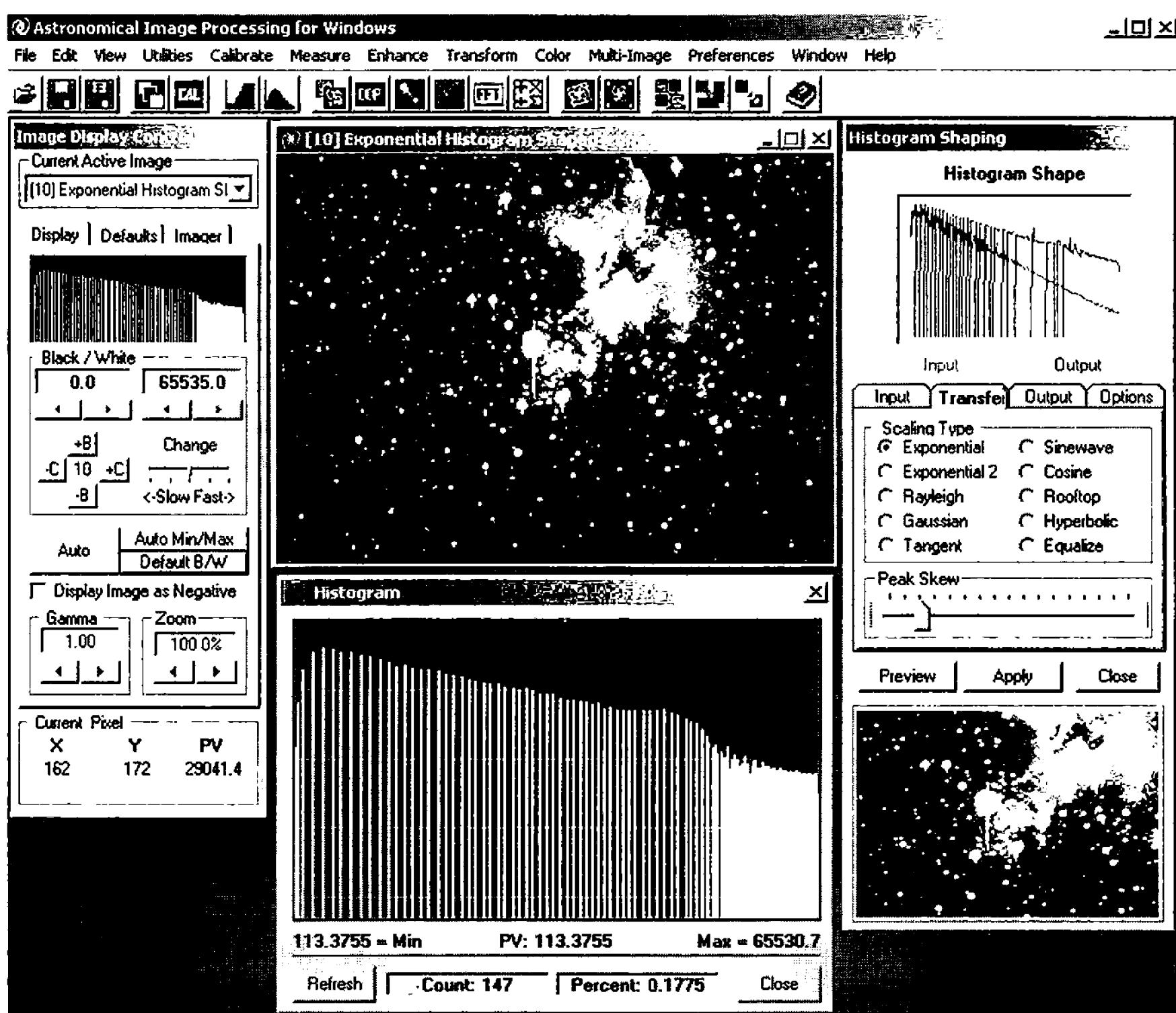


Figure C.9 Changing the value of pixels is the first and foremost technique used in astronomical image processing. Shown above, the Histogram Shaping tool modifies pixel values in the Eagle Nebula image to reveal both the dim glowing cloud of gas and the bright star cluster at its core.

used. **AIP4Win** uses brightness scaling operations when displaying data on your computer screen. Having the Auto Low/High Stretch capability selected causes the program to examine the range of pixel values in the image and assign the lowest value found to black and the highest one to white. All the other values are spread across the 254 available gray levels in between.

**Step 1: Load an Image.** To make the results of this tutorial more visible, we are going to set the default image stretch mode to Black/White Stretch. You do this by selecting the **Black/White Stretch** button on the **Defaults** tab of the **Image Display Control**. (You will need to load an image first, in order for this window to come up.) Make sure that the **Black/White Stretch** values are set to 0 and 65535. First load the image file “m16.fts” from the Image Enhancement subdirectory of the Tutorial directory on the CD-ROM. When the image loads, click the **Default B/W** button on the **Image Display Control**.

You will see that the image is completely black, or only faintly displays the brighter stars and blooming trails, and gives no hint at all at what faint objects hide in it. In this view, the pixel value of zero is assigned to the gray level of zero, and

## Appendix C: Tutorials

the value of 65535 (the maximum value for a 16-bit CCD camera) is assigned the gray level of 255. If you right-click on the image, the Image Status window will open, displaying (among other things) the minimum and maximum pixel values present in it.

For now, leave the **Black/WhiteStretch** as the Default Display Mode, as it will make the effects of the various enhancement tools much more apparent.

**Step 2: Invoke the Brightness Scaling Tool.** Click on the *Enhance|Brightness Scaling...* menu item to bring up the Scale Image window. Leave the Info window on the screen; you will be using it later.

The Brightness Scaling tool window has the following controls and displays:

- a **Transfer Function** display, which shows the histogram of the image, overlaid with the transfer function curve being applied.
- a set of tabs labelled **Inputs**, **Transfer**, **Outputs**, and **Options**,
- an **Inputs** tab, which contains the **Low** and **High Pixel Values** controls, used to set the black and white pixel values in the image. The **Low** and **High Pixel Values** controls are automatically updated by clicking the **Auto** button after adjusting the **Black Point** and **White Point** controls. They can also be set manually,
- a **Transfer** tab, which contains buttons to set the **Scaling Type** and its associated parameters. This controls the shape of the transfer function,
- an **Outputs** tab, which contains the **Low Output PV** and **High Output PV** controls, used to set the output range of the pixel values in the image,
- an **Options** tab, which contains controls that allow you to adjust what information is displayed in this tool,
- a **Preview** display, allowing you to see the results of a scaling operation before it is performed on the whole image. This is updated by clicking the **Preview** button, or by changing any of the parameters on this tool.

**Step 3: Set the Scaling Range.** Initially, the Min PV and Max PV values are set to the values determined by the default settings of the **Black Point** and **White Point** controls. Leaving the Scaling type set at Linear for now, set the **Black Point** to 0.01 and the **White Point** to 0.98, and then click the **Auto** button. You will see that **Low/High Pixel Values** change to near 51 and 172. If the preview display hasn't updated yet, click the **Preview** button to see what the image will look like with these values. Suddenly, the familiar form of the Eagle Nebula appears in the preview display. The preview capability operates on a reduced-size version of the CCD image. (It may not display the entire image.) It is much faster than trying out different combinations of **Low/High Pixel Values**, and **Scaling Types** and generat-

ing a new image for each one. This is especially important if you have limited memory, a slower CPU, or a large image loaded.

**Step 4: Create a Scaled Image.** Click the **Apply** button and a new image will be created from the original, scaled using the parameters you entered. Click on the Info window to bring it to the foreground and click its **Update** button; you will see that the Min PV is now zero, the Max PV is 65535. Examine the two images side by side and notice how much more detail is visible.

**Step 5: Gamma Scaling.** Click on the original “m16.fts” image to make it current. Set the **Black Point** and **White Point** values to 0.01 and 0.98 and select **Gamma** scaling on the **Transfer** tab. Click on the buttons to the right of the **Gamma** control in the **Scaling Parameters** box to obtain a value of 1.8. Notice that as the value increases from its default of zero, the low-level detail is now more prominent. Click the **Apply** button to create a new image and compare it to the one created using linear scaling. Notice the difference in the small histogram displayed on the Image Display Control as you click on the titlebar of the linear scaled image and then the titlebar of the gamma scaled image. See how the lower values in the histogram have been spread out.

**Step 6: Gammalog Scaling.** Perform the previous step again, but this time select **Gammalog** scaling. Remember to re-select the “m16.fts” image to make it the current image. Use the default GammaLog value of 0.350. Compare the image you created to the Gamma scaled image. Notice how the low-level detail is enhanced, but the background contrast has increased.

**Step 7: Sawtooth Scaling.** Select **Sawtooth** scaling. Use the default value for **Sawteeth** of 5. Remember to select the original image and to reset the **Low Point** and **High Point** values to 0.01 and 0.98 and to click the **Auto** button. The **Preview** display will show you a grayscale that repeats itself over and over to show up a great deal of detail in the structure of this nebula. This type of scaling is not always very pretty, but it is hard to beat when you want to see the subtle details.

**Step 8: Experiment.** Try out the different scaling techniques and play with the parameters to get a feel for how the tool affects different images.

Close all the open images and tools when you are finished. You can do this in one step by clicking on the *Window|Close all Images* menu item.

## C.7.2 Histogram Shaping

The previous tool, Brightness Scaling, operates on an image by examining the range of pixels in it and assigning new pixel values based on where each lies in the range between the values selected for black and white. Histogram Shaping, while looking superficially similar, works in an entirely different way. While both tools reassign each pixel value in the original image to a new value, Histogram Shaping does it by trying to force the histogram of the image, as a whole, to a new shape. While we used the Statistics Tool to examine the effects of the Brightness Scaling Tool, we will use the Histogram Display to show the effects of Histogram Shaping.

## Appendix C: Tutorials

**Step 1: Load the Image.** For this tutorial, we will again use the image of the Eagle Nebula, “m16.fts” from the Image Enhancement tutorial directory. Load it, and click the **Default B/W Stretch** button for this display mode.

**Step 2: Examine the Image Histogram.** Create a histogram of the image by clicking the *Measure|Histogram...* menu item. Examine it and notice how the spectrum peaks at 47 ADUs and falls off in an exponential fashion until it hits 10 pixels at a pixel value of around 1000 ADUs. It has a narrow peak at about 1500 and then a broader one at 3000. Most of the image structure is buried in that initial peak, and it’s only about 600 ADUs wide. What we will do is to change the shape of this histogram in various ways to bring out the structure of the nebula.

**Step 3: Invoke the Histogram Shaping Tool.** Click on the *Enhance|Histogram Shaping...* menu item to invoke the Histogram Shaping Tool. This tool window contains several tabs and a Preview display, just as with the Brightness Scaling Tool. Preview allows you to experiment with different histogram shapes until you find one that suits the image and features you are trying to bring out.

The **Transfer** tab contains a **Scaling Type** box with buttons showing ten types of histogram shaping available. Some of the different histogram shapes have a parameter which is set with the slider underneath the **Scaling Type** box, while others take no parameters.

**Step 4: Preview an Exponential Histogram Shape.** The default histogram shape is the Exponential Histogram. This option has proven to be effective for many faint deep-sky objects. Examine the **Preview** display and see how it works on the image of the Eagle Nebula. Slide the **Peak Skew** slider around and notice how the pre-calculated histogram shape changes. This is intended to help you visualize how you are shaping the image histogram. The red curve shows the histogram of the original image, while the blue curve shows the histogram that will result from the application of this tool, and the **Preview** display shows how the image will look.

**Step 5: Create a New Image with an Exponential Histogram Shape.** Set the **Sigma Spread** to a value of 42 and click **Apply**; a new image will be created with the Exponential histogram shape.

**Step 6: Compare the Histograms of the Two Images.** Now examine the histogram of the new image. Click on the still-open Histogram window and click the **Refresh** button. This histogram is radically different from that of the original image. First, it now stretches from down around zero to up around 65535, the maximum value represented in 16 bits. Second, it is evenly spread across the whole range. Because of the exponential shape, the range of values containing the nebula now stretches from about 10,000 to roughly 50,000, a full two-thirds of the histogram. As a result the nebula is very prominent in the image.

**Step 7: Try Other Histogram Shapes.** Close all the images, which will close all the tools, and reopen the “m16.fts” image. Try the different histogram shapes on it. Also try other images in the collection on the CD-ROM. See how planetary images benefit from different histogram shapes than those used for deep-sky objects.

When you are finished, close all the open images.

### C.7.3 Convolution Filtering

Convolution filters are rather coarse devices used to sharpen or soften an image. You can use them on lunar and planetary images to help remove the softness caused by poor focus and bad seeing. You can also use them to build your own unsharp masks, although **AIP4Win** provides several ways to do this automatically. Essentially, these filters are useful building blocks of more complex operations.

For this tutorial, be sure that **Auto Lo/HiStretch** is selected as the default display mode in the Image Display Control window.

**Step 1: Load an Image.** From the Image Enhancement Tutorial subdirectory, load the image “moon.pa.” This is an image taken with a Cookbook 245 CCD camera through a telescope with a tiny 6-millimeter aperture, so the image appears rather soft. We will use this image later with some more powerful techniques.

**Step 2: Apply the Noise Filter.** The Noise Filter is not, in a strict sense, a convolution filter, but rather a local median filter. It is useful in this case because a dark frame was not acquired with the image, and there are a few hot pixels that affect the processing. If you look around X = 80, Y = 180, you will see one of them. The Noise Filter can be useful in cleaning up images like these; but take care, it does remove fine detail.

Click the *Enhance|Noise Filter...* menu item and the Noise Filter window will appear. Leave the **Deviation** control set at 1 and click **Apply**. A new image will appear in which the hot pixels are gone.

**Step 3: Apply the Crispen Filter.** Click the titlebar of the noise filtered image to make sure it is active. Then click the *Enhance|Convolution Filters|Crispen...* menu item. A new image will appear in which the detail has been greatly enhanced. See how the contrast has increased and the detail in the craters has become more prominent.

Now try the crispen filter on the original “moon.pa” image by clicking its titlebar and invoking the crispen filter again. You will see a new image appear in which the lunar details are more prominent, as before; but also notice that the hot pixels in the image are enhanced along with the others. The filter enhances everything; it has no way of knowing which pixels are “pure” and which have been corrupted by noise. High-pass filters enhance noise.

**Step 4: Apply the Sharpen Filter.** Try the same steps you used in Step 2—but using the Sharpen Filter. Notice how the resulting image is much harsher. This is a fairly strong filter.

**Step 5: Apply the Low-Pass Smooth and Blur Filters.** Try the same steps using the Smooth and Blur filters and notice how the fine details in the image are suppressed.

When you are finished experimenting, close all the open images.

## Appendix C: Tutorials

### C.7.4 Unsharp Masking

Unsharp masking is a powerful technique for exposing detail in an image where bright regions tend to wash it out. It is especially useful with planetary images, bright nebulae and overexposed galaxy cores. It is the real workhorse of the planetary imager.

For this tutorial, be sure that **Auto Lo/HiStretch** is selected as the default display mode in the Image Display Control window.

**Step 1: Load an Image.** From the Image Enhancement Tutorial subdirectory, load “03119204.ccd.” This is a nice clean lunar image that has been affected by the turbulent atmosphere during the exposure. We will compare the unsharp masking technique to the crisper filter we used in the last tutorial section.

**Step 2: Invoke the Unsharp Masking Tool.** Click on the *Enhance|Unsharp Mask...* menu item to open the Unsharp Mask Tool. Leave the **Mask Type** set to Gaussian with a **Radius** of 2 and set the **Contrast** to 5 and click **Apply**. See how much more detail is visible in the crater floor.

Try it again with the Parabolic mask. Be sure to click on the original image first. See how the Parabolic mask has a slightly more contrasty result. Try the other mask types and compare the results.

Try adjusting the **Radius** and **Contrast** controls as well to become familiar with their effects.

As the screen fills up with images, you may get confused as to what image has had what kind of processing applied to it. To see the history of any image on the screen, right-click on it, or click on the *Measure|Statistics|Image...* menu item. The Image Status window will pop up. Select the **History** tab, and you will be presented with a list showing the original image name and all the processing steps in the order in which they were applied to the image. This history is saved in the FITS header when you save an image as a FITS file.

**Step 3: Open the Next Image.** Open the image “J0703075.ccd.” This one, by expert planetary imager Don Parker, shows Jupiter taken in white light. As you can see, there is a hint of detail there, but the contrast is very low. If you adjust the **Black** and **White** controls on the Image Display Control window, you can get a feel for how much detail is there that is not readily visible because of the wide dynamic range of the image. Unsharp Masking compresses the dynamic range to make it more visible to the eye.

**Step 4: Apply an Unsharp Mask to the Image.** Invoke the Unsharp Mask Tool and apply a Gaussian mask with a radius of 3 and a contrast of 4. Click the **Auto Min/Max** button on the Image Display Control window to see the entire range of brightness in the image. See how much hidden detail was brought out using the Unsharp Mask Tool.

When you are finished, close any open images.

### C.7.5 Deconvolution

Deconvolution as a technique for restoring astronomical images became really

well known when it was discovered that the Hubble Space Telescope mirror was flawed. The near miraculous ability of this computationally intensive technique saved the day for NASA, and, judiciously applied, can be used to reduce the effects of atmospheric seeing and slightly less-than-perfect focus in your own images. In this tutorial we will use the two types of deconvolution provided by **AIP4Win** to enhance the detail in both planetary and deep-sky CCD images.

For this tutorial, be sure that **Auto Lo/HiStretch** is selected as the default display mode in the Image Display Control window.

**Step 1: Load a Planetary Image.** Load the image “03119204.ccd” that we used in the previous session. You can compare the results of using deconvolution with the unsharp masking technique we covered earlier.

**Step 2: Set Up the Deconvolution Parameters.** Deconvolution is a much more complex process than any we have used up to this point and, depending on how the parameters are set up, can take a very long time to run. The execution time is dependent on the size of the image, the size of the mask used, the deconvolution algorithm applied, and the number of iterations. We have purposely selected a small image in order to allow you to experiment with the various processing parameters to get a feel for how they affect the results.

Invoke the Deconvolution Tool by clicking the *Enhance|Deconvolution...* menu item. The Deconvolution Tool will open, and you will see a series of three tabs containing the following displays and controls.

On the **Point Spread** tab:

- **Point Spread Function.** Three options for controlling the point spread function used in the deconvolution are provided.
- **Gaussian PSF** is the default, and it provides a text box in which you enter the sigma of the PSF (the sigma is half of the FWHM of the star image). This is the option we will use for this tutorial. Set the PSF to a value of 2.0.
- **Gaussian PSF based on selected stars** allows you to determine the sigma of the PSF by selecting a series of stars to average together.
- **PSF constructed from selected stars** allows you to create a custom PSF by selecting a series of stars, whose profiles are averaged together to create a kernel used in the deconvolution.
- **PSF Display.** Here you are presented with a view of the actual kernel used in the deconvolution process.

On the **Deconvolution** tab:

- **Select Process.** There are three options here: van Cittert, Lucy-Richardson (Fast) and Lucy-Richardson (Slow). Van Cittert is the best for planetary details, while L-R works best on deep-sky objects. It is worth experimenting on your images, but these are good starting points. For this image, select the van Cittert algorithm.

## Appendix C: Tutorials

- **Number of Iterations.** Each of the deconvolution routines is iterative, meaning it gets applied over and over. Five to ten iterations usually work the best, but larger numbers can be very effective (if somewhat time-consuming) for some images. For this first run, set to a value of 16.
- **Relaxation Parameter.** This controls the damping of noise. For this first run leave it set at the default value of 0.1.

On the **Settings** tab:

- **Display Iterations.** Checking this box will allow you to monitor the progress of the deconvolution. During this process, a status box will be displayed which contains a **Stop** button. Clicking on this button will stop the deconvolution at the current number of iterations. This can be handy when you are not sure how many iterations you want it to execute. For this tutorial, make sure the **Display Iterations** box is checked.
- **Process high-frequency components only.** In general, deconvolution has its greatest effect on the high-frequency components of an image. Low-frequency parts can be removed from the image, the deconvolution applied to the high-frequency components that remain, and then the original low-frequency part can be re-combined with the result. For now leave this unchecked.

**Step 3: Run the Deconvolution.** Now click the **Execute Deconvolution** button to begin the process. A status bar will appear to let you know the progress. Five iterations on a small image should take only a few seconds.

When it finishes, you will be left with a new image that looks a bit sharper than the original. With only five iterations, the results are slightly less sharp than the unsharp masked version we created earlier. Run an unsharp mask on the original image to compare the two.

Try running the deconvolution on the original image with more iterations but the same other parameters. For a start, try 32 iterations. The image initially becomes sharper with more iterations. However, at some point—typically several hundred iterations—increasing the number will produce no further improvement.

Close all the open images when you are done.

**Step 4: Try a Deep-Sky Object.** Load the image “NGC6888.fts” from the Image Enhancement tutorial subdirectory. This is an image of the Crescent Nebula imaged through a Hydrogen-alpha filter by one of the authors. We will use L-R deconvolution to sharpen up the star images. Invoke the Deconvolution Tool (it helps to move it off to the left side of the screen) and set it up as follows:

**Point Spread Function:** Gaussian PSF based on selected stars

**Deconvolution Type:** Lucy-Richardson (Slow)

**Relaxation Parameter:** 0.1 (default)

**Number of Iterations:** 32 (default)

**Display Iterations:** checked (default)

**Process high-frequency components only:** checked (default).

Now select a few stars with the mouse and click on each one. Try to choose stars that are not too bright or saturated—what we are looking for is “average” stars. As you click on each, you will see the PSF value changing as its PSF is averaged into the group. The **PSF Display** will show you the PSF that will be used in the deconvolution. The value you get for the PSF is dependent on the profiles of the selected stars, and varies depending on the optics, the camera pixel size, and the seeing conditions present when the image was acquired.

Then click **Execute Deconvolution** to start the deconvolution. Because this image is larger, iterations take longer than they did with the small planetary image. When it is done, you will need to change the display parameters to compare the two. Select each image and click on the **Auto** button. You will also find it helps to set the **Gamma** to a value of 2 to bring out the faint nebulosity. Notice how much sharper the star images are, and how the bright star at X=288, Y=280 is much less bloated. Detail in the nebula is also enhanced.

The value for sigma used in the PSF needs to be carefully chosen in order to avoid dark ring artifacts around stars. The rings are characteristic of Lucy-Richardson deconvolution with the PSF value set too large. By specifying this value through the selection of typical stars in the image, these artifacts can be all but eliminated.

### C.7.6 Wavelet Spatial Filtering

One of the most exciting image processing technologies to come along in recent years is wavelet processing. **AIP4Win**’s Wavelet Spatial Filter provides unique capabilities for image enhancement not matched by other processing techniques. To get the most out of this tutorial it is recommended that you read Chapter 18, so you will have a better idea of what is happening, as well as a better understanding of the terminology.

For this tutorial, be sure that **Auto Lo/HiStretch** is selected as the default display mode in the Image Display Control window.

**Step 1: Load an image.** Load the image “IC5070HaStack.fts” from the Image Enhancement tutorial subdirectory. This image is a Hydrogen-alpha filtered image of the Pelican Nebula. (The pelican namesake is recognizable if you use the *Transform|Translate-Rotate-Scale...* tool to perform a quick 90 degree counter-clockwise rotation of the image. Try it and see.)

**Step 2: Invoke the Wavelet Spatial Filter tool.** Click on the *Enhance|Wavelet Spatial Filter...* menu item, and the Wavelet Spatial Filter tool will open. This tool is divided into four panels with the following controls:

- **Maximum Wavelet Scale.** Sets the limit of the largest pixel scale that will be processed.
- **Wavelet Profile** sliders. Set the amount of enhancement for each pixel scale.

## Appendix C: Tutorials

- **Add Constant** checkbox. Allows you to add a bias to the image, to help keep the processed values from going negative.
- **No Dark Rings around Stars** checkbox. Does just what it says; checking this box prevents the dark rings that can occur during aggressive filtering operations.
- **Select/Un-Select All**. These buttons can be used to control whether the entire range of pixel scales is processed or not.
- **Settings**. Allows you to save and recall your favorite filter settings.
- **Apply Filter** button. Starts the wavelet filtering process.

To make this tutorial easier, we have saved a filter using the **Save** button and will recall it to quickly set up the tool.

**Step 3: Load a predefined filter.** Click the **Recall** button to load the filter that has already been set up. This tool allows you to configure a filter and save the settings in a file so that you can recall it later. In this way you can create a library of your favorite filters and share them with other users. The filter is called “Sharpen.wlt” and is located in the Enhancement tutorial subdirectory where you found the current image file.

**Step 4: Run the Wavelet Spatial Filter.** Now click the **Apply Filter** button. A status bar will appear to let you know the progress. Wavelet filtering demands an enormous amount of computation, so filtering a *really* large image can take several minutes. The status bar provides a **Cancel** button if you need to stop the process before completion.

You will see that when stretched to the same range as the original image, the resulting image has much greater contrast. This tool tends to enhance brighter features and abrupt edges and is wonderful for enhancing nebulae. It works particularly well on images taken through a Hydrogen-alpha filter.

Try an experiment, and compare the result with the Unsharp Mask and Deconvolution tools.

### C.7.7 Morphological Processing

Morphological processes are not the usual stock-in-trade of astronomical image processing, but some of them do have their uses from time to time. **AIP4Win** provides a whole suite of these functions, the most useful of which are the feature-detection operators. In this tutorial, we will explore several of those functions. However, the Contour Map is probably the most fun. Our test image shows the Whirlpool Galaxy, M51.

**Step 1: Load the Image.** Load the image file “m51.fts” from the Image Enhancement subdirectory of the Tutorial directory on the CD-ROM. Check the **Auto** button on the Image Display Control to make sure the image is fully stretched.

**Step 2: Invoke and Setup the Contour Map Tool.** Click the *Enhance | Morphological Operations | Contour Mapping...* menu item to bring up the Con-

tour Map Tool. There are two mapping options: **Outline Regions of Equal Area**, or **One Contour Line at a Given Pixel Value**. Select the first one, which will create a display like a topographic map. There is a slider/text box labelled **How Many Contour Lines?**. The more lines you have, the longer the map will take to generate. For a slow PC, this can be quite a while. For this example, pick 5 (unless you have a fast machine; then, choose 10).

**Step 3: Create the Contour Map.** Click the **Apply** button and a window will pop up that shows the status of the mapping operation. When it is done, a new image will appear containing the contour map of the original image.

**Step 4: Adjust the Display.** You can use the Image Display Control to remove the gray sky background by increasing the **Black** value until the background darkens. What you are doing here is to turn off the contour lines around the background image noise.

**Step 5: Blur the Original Image.** Click on the original image, and then invoke the Blur Tool by clicking on the *Enhance|Blur...* menu item. We will use the Blur Tool to apply a Gaussian blur to the image to smooth it. This will reduce the jaggedness of the contour lines.

Select a **Radius** of 3 pixels, click **Apply**, and a fuzzy version of M51 will be created.

**Step 6: Rerun the Contour Map.** Invoke the Contour Map Tool and set it up as in Step 3, but this time using the blurred version of the image, and click **Apply**. You will notice that the contour lines are much smoother in the resulting map.

Experiment with the various settings and try resampling the image to 200% (resample using *Transform|Resample*). This will leave more room for the contour lines in the map. Close all open images when you are done.

## C.8 Fast Fourier Transform

**AIP4Win** implements one of the most powerful of all image processing tools: spatial filtering. The Fast Fourier Transform (FFT), the basic process behind most spatial processing, is covered in Chapter 17. Before running this tutorial, read this chapter to gain an understanding of spatial filtering.

The Fast Fourier Transform is a computationally accelerated technique for determining the frequency spectrum of a signal. In two dimensions, it can be applied to an image. The reason for transforming an image from the spatial domain to the frequency domain is that some operations, most notably filtering operations, become much simpler. In filtering an image, we are usually trying to enhance or suppress detail that occurs when pixels make a transition in intensity. Transitions that happen rapidly contain high frequencies, and transitions that happen more gradually contain low ones.

When we applied the Crispener Filter to the image of the Moon in Section C.7.3, we were enhancing the high frequencies to bring out detail. The Crispener Filter is a convolution filter, which is limited in what it can do, and tends to be

## Appendix C: Tutorials

rather coarse. The FFT filters we will demonstrate in this tutorial are extremely flexible, and allow you to exercise much more control over the filter characteristics.

The basic steps in filtering an image using FFT techniques are:

1. Convert the image to suitable dimensions.
2. Perform a forward FFT to transform it to the frequency domain.
3. Create a filter mask suitable to the filtering desired.
4. Multiply the image by the filter mask.
5. Perform an inverse FFT on the result.

This tutorial will demonstrate these steps using the manual FFT tools, and then demonstrate the use of the Auto FFT feature of **AIP4Win**. The images to be used can be found in the Fast Fourier Transform subdirectory of the Tutorials directory on the **AIP4Win** CD-ROM.

**Step 1: Load an Image and Convert the Dimensions.** The FFT technique has an inherent limitation. Any image you wish to apply it to needs to be square (i.e., have an equal number of pixels in each dimension), and the dimensions need to be integer powers of 2 (e.g., 2 x 2, 4 x 4, 8 x 8, 16 x 16, and so on) These are requirements of the FFT algorithm. It is also desirable that the pixels in the image be squared up prior to the processing.

Load the image “triangle.fts.” It is a simple 128 x 128 image consisting of a white triangle on a black background. This image is of suitable dimensions already. Later on we will apply the technique to a more “real world” example.

**Step 2: Perform a Forward FFT.** Perform a forward FFT on this image by clicking the *Enhance|Manual FFT|Forward FFT* menu item. An image appears which is a bunch of diagonal lines. This image container holds the frequency-domain version of the triangle image. The picture it shows is a plot of the frequencies in the image versus their amplitude. The frequency plot has zero in the center with positive frequencies to the right and negative ones to the left. This is called the real axis. Running vertically through the center of the plot is the imaginary axis. Amplitude is plotted as intensity of the pixels in the picture. Low frequencies are clustered around the center of the plot and increase in frequency as you move outward.

**Step 3: Create a Butterworth Filter Mask.** Click the *Enhance|Manual FFT|Butterworth Filter Mask...* menu item and the Butterworth Filter Generator window will pop up. This window is used to generate a filter mask to be applied to the image. The filter mask is multiplied by the FFT image to change the distribution of frequencies in the image. If the filter mask has very small values around its center, the masked FFT will have severely reduced low-frequency components. If the mask is dark around the edges, the masked FFT will have reduced high frequency components. One big advantage this approach gives us is that we can also generate “band pass” and “band stop” filters, which leave the low and high frequencies and remove those in between, or vice-versa.

The Butterworth Filter Generator window has a display showing a plot of the frequency characteristics of the filter it will generate, along with controls to

change the shape of it. There is also a control to affect the size of the filter mask. This mask must be the same size as the image you are filtering.

Click the **Low Pass**, **High Pass**, **Band Pass** and **Band Stop** radio buttons to see what the various filter shapes look like. Move the **Filter Order** slider back and forth, and you will see the filter get steeper and shallower. When the low pass filter is selected, a **Low Pass Radius** control is visible which changes the cutoff point for the filter. Each of the other options enables a different control. The band pass and band stop filters have an additional control to adjust the width of the pass or stop band.

Now let's generate a mask. Set the filter mask for a **Low Pass** filter with a **Filter Size** of 128 x 128. Set the **Filter Order** to 1 and a **Low Pass Radius** of 16. Click **Generate Filter**, and a filter mask will be displayed as an image. As you can see, it is bright in the center, gradually fading to the edges. This tells us that the low frequencies will be relatively unaffected, but that high ones will be attenuated more and more as frequency increases.

**Step 4: Mask the FFT.** Now let's apply the mask to the FFT. Click the *Enhance|Manual FFT|Mask Fourier Transform...* menu item. The Mask FFT window will appear.

This window contains four controls:

- **Fourier Image.** Selects the Fourier Transform image that you want to mask.
- **Masked With.** Selects which image to use as a mask. Choose the Low-Pass Filter image.
- **Base Value.** When the base value is 0, frequencies blocked by the filter remain blocked; when the base value is 1, all frequencies present in the original image are passed through the filter. For now, leave this set to 0.
- **Contrast.** Multiplies the transmission of the filter by the contrast factor. This means that the filter does not merely attenuate unwanted frequencies, but can also amplify desired ones, increasing the effective contrast of the image. For now, leave this set to 1.

Now click **Apply** and a copy of the FFT image appears, but it is bright in the center and darker toward the edges, showing the effects of having been multiplied by the filter mask. Remember, this display is just a representation of the FFT of the image. The real FFT is a pair of complex, floating point arrays containing the real and imaginary frequency components of the triangle image. You cannot directly edit the FFT.

**Step 5: Perform an Inverse FFT.** Now click on the Butterworth Filter masked version of the FFT that was just produced to make sure it is active and then click the *Enhance|Manual FFT|Inverse FFT* menu item. A “fuzzy” version of the original triangle will appear. This “fuzziness” is caused by the high frequencies associated with the abrupt transition from pure white to pure black at the edges of the triangle.

## Appendix C: Tutorials

**Step 6: Experiment.** You can use the FFT Mask Generator to create different filters that have a number of effects on the triangle image. Just repeat steps 3, 4 and 5 above, using different parameters for the Butterworth Filter.

Close all the images except for the original “triangle.fts” image when you are done.

**Step 7: Auto FFT.** Performing FFT filtering manually is a complicated process. **AIP4Win** provides an automated tool to do this for you. Click the *Enhance | FFT Filter...* menu item. The Butterworth FFT Filter window will appear.

At first glance this looks just like the FFT Mask Generator window, but note that the **Filter Size** radio buttons are missing, and the controls from the Mask FFT window have been added. This tool will automatically take the selected image and float it on a larger one, if necessary, to bring it to the required square, power-of-2 dimensions needed by the FFT algorithm. It will then automatically run each of the steps listed above, and you will be able to see the individual windows appear as the steps are performed.

Select the “triangle.fts” image, and then set up the filter as follows:

**Filter Shape:** Low Pass

**Filter Order:** 4

**Low-Pass Radius:** 4

**Base Value:** 0

**Contrast:** 1

Click **Apply** and watch what happens. The result should be a series of images flashing by, leaving a very fuzzy version of the original image.

**Step 8: Try a Real Image.** Open the “030119204.ccd” image from the Image Enhancement tutorial directory. Invoke the Auto FFT Tool and set it up as follows:

**Filter Shape:** High Pass

**Filter Order:** 1

**High-Pass Radius:** 16

**Base Value:** 0

**Contrast:** 1

Click **Apply**. A new image will be generated which is a greatly sharpened version of the original. You may want to raise the **Min. Pixel Value** control on the Display Control window to improve the contrast. You can use the Brightness Scaling Tool to adjust the contrast as well.

Try out different settings on the Auto FFT Tool to see how they affect the appearance of the final image. Like all the other tools, it pays to experiment to get comfortable with how it will perform.

Close all open images when you are finished.

## Section C.9: Multiple Image Processing

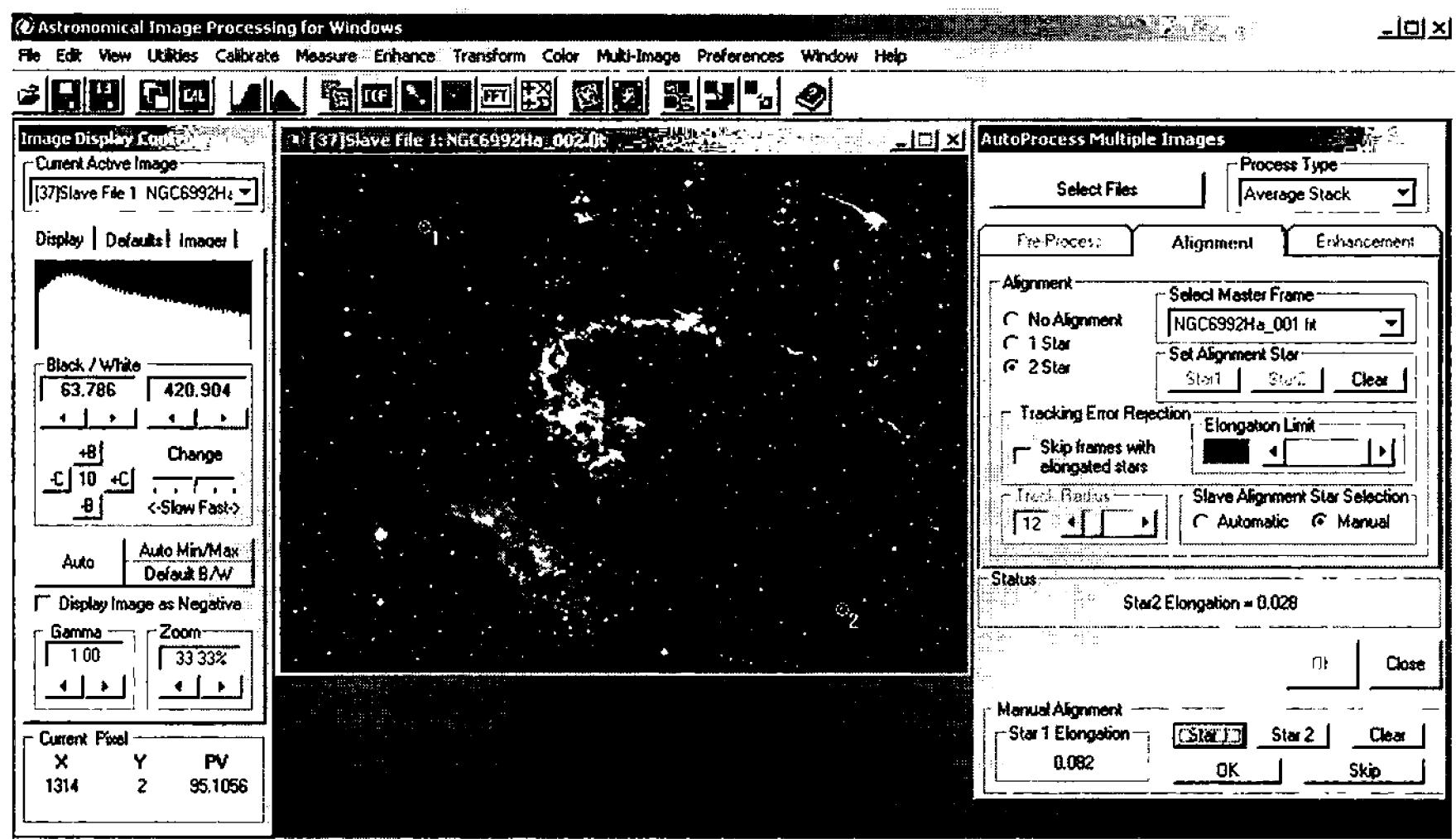


Figure C.10 Your telescope mounting may not track perfectly for an hour, but by summing short exposures in **AIP4Win** software, you can have perfect tracking with an imperfect telescope. In this tutorial, you will stack eight deep H $\alpha$  images to make one super-deep image of a section of the Veil Nebula.

## C.9 Multiple Image Processing

**AIP4Win** provides a set of multiple-image processing features that allow you to perform repetitive operations on a specified set of images. These processes provide the ability to:

- Calibrate an entire observing session's worth of images at one time automatically.
- Align and enhance a set of images in preparation for creating a movie.
- Align and combine a group of images to create a single, “deeper” image.
- Enhance a set of planetary images.

This tutorial demonstrates creation of images with Track & Stack, one of the most useful of the multiple-image operations.

### C.9.1 Track and Stack

Track and stack is a technique by which multiple shorter-exposure images are aligned with each other and added together to achieve the effect of a single, deeper image. This method is useful when short exposures are desired to prevent bright stars in the field from blooming. It is also frequently used with CCD cameras attached to telescopes that don't track by themselves long enough to avoid trailing on longer exposures. Some people even use this method to avoid guiding images altogether.

## Appendix C: Tutorials

Nothing comes for free, however. While the image on a CCD adds directly, making the final image value equal to a single exposure with the same total integration time, the quantization and readout noise adds with the square root of the number of images combined. So a stack of images is noisier than the equivalent single exposure.

In this tutorial we will walk through the steps used to combine a stack of eight exposures into a single image. The relevant files are found on the CD-ROM in the *Tutorials\Multiple Images* directory and comprise a series of images of the Veil Nebula along with their corresponding calibration frames.

**Step 1: Set up the Image Calibration.** You need to have the calibration set up before you begin the AutoProcessing. Click on the *Calibrate|Setup...* menu item to open the Calibration setup window. Select Advanced for the **Calibration Protocol**.

Select the **Bias** tab and click the **Use Bias Frame** radio button. Click the **Select Bias Frame(s)** button and when the file selection dialog pops up, navigate to the “*Tutorials\Multiple Images*” directory on the CDROM and select the file named “*1x1SXVMasterBias.fts*” and click **Open**. Now that the bias frame has been loaded, click the **Process Bias Frame(s)** button; you will see that the **Subtract Bias** checkbox is now checked.

Select the **Dark** tab and click the **Select Dark Frame(s)** button. Use the file “*1x1SXVMasterDark480.fts*” for the dark frame. Click the **Automatic Dark Matching** radio button and then click the **Process Dark Frame(s)** button; a status bar will pop up as the dark frame is processed. When it finishes, you will see that the **Subtract Dark Frame** box is now checked.

Select the **Flat** tab and click the **Select Flat Frame(s)** button. Use the file “*1x1SXVMasterFlat.fts*” for the flat frame. Click the **Process Flat Frame(s)** button; you will see that the **Apply Flatfield Correction** box is now checked.

Select the **Defect** tab and click the **Select Defect Map** button. Use the file “*1x1SXVDefectMap040714.fts*” for the defect map; you will see that the **Correct Defects** box is now checked.

Now dismiss the Calibration setup window by clicking **Close**.

**Step 2: Invoke the Deep Sky AutoProcess Tool.** Click on the *Multi-Image|Auto-Process|Deep Sky...* menu item, and the AutoProcess Multiple Images window will appear. This is definitely one of the most complex windows in the entire program, but for this tutorial we will only be using a few of its features.

The Deep Sky AutoProcess window is arranged in a series of tabs, in the order in which each is used. Rather than describe each one here, we urge you to read the help file associated with this window. You can access the context-sensitive help screen for this window by clicking its titlebar and then pressing the F1 key. You don’t need to read it now, but be aware that this tool is completely described in the on-line help, as are all the others.

Until a set of files is selected, the tabs are disabled, in order to prevent unintended user actions.

**Step 3: Select the Files to be Stacked.** Click the **Select Files** button and select the files “NGC6992Ha\_001.fit” through “NGC6992Ha\_008.fit” by clicking the first of the files, holding down the shift key, and clicking the last of the files and click **Open**.

**Step 4: Setup the AutoProcess Parameters.** For this tutorial you will need to select the following options:

- **Process Type.** Select **Average Stack** if it is not already selected. It will cause the images to be combined in a stack and averaged together. Other possibilities are **Sum Stack**, **Median Stack**, **k-sigma Stack** and **Individual Files**.
- **Calibrate Image.** Check this box on the **Pre-Process** tab to use the calibration we just set up.
- **2X Resample.** Check this to preserve as much of the image resolution as possible when the image is shifted. Leave it unchecked for this tutorial.
- **Prescale.** Check this box to cause pixel values in each image to be multiplied by the Scale Factor before the image is added to the stack. Leave it unchecked for this tutorial.
- **Scale Factor.** Leave this set to the default of 10.
- **Noise Filter.** Check this box and set the **Deviation** to its maximum value of 3. This will remove any remaining hot pixels or cosmic ray hits in the images as they are stacked.

**Step 5: Select the Master Frame.** On the **Alignment** tab, use this drop-down list to select one of the images for use as a master when doing an alignment. Generally, you will have fewer difficulties tracking if you select the first image in the series; in this case you would select “NGC6992Ha\_001.fit.” The image will now be calibrated, noise filtered and displayed. You can see that it is a bit noisy. By lowering the **White** value on the Image Display Control, you will be able to more easily visualize the noise. This is a large image. To make room on your screen, you can set the **Zoom** on the Image Display Control to some convenient value less than 100%. Now when each subsequent image is displayed, it will be zoomed to this size.

**Step 6: Select the Alignment Mode.** For this image stack you will be using the **2 Star** alignment mode. This gives results superior to 1-star alignment, but it involves a little bit of additional work, since two stars need to be selected. It has the advantage of correcting field rotation between images.

Click on the **2 Star** alignment radio button, and you will see a set of alignment star selection buttons appear. There is also a **Track Radius** control, which is used to adjust the size of the region in which the program will attempt to find a stellar centroid. In dense areas it helps to make this small enough so that only a single star fits inside. In sparser starfields it can be set larger to make it easier to capture a centroid. For this tutorial you should set it to a value of 12.

**Step 7: Choose the Alignment Stars.** The alignment stars in the image

## Appendix C: Tutorials

consist of a pair of stars as widely separated as possible, which exhibit nice round images. The only serious requirement for selecting each star is that it be present in all of the images you are trying to stack. To this end you will want to keep well away from the edges of the image. Generally, selecting stars in opposite corners works the best.

There is an excellent candidate at X = 286, Y = 138. If the image covers the Display Control window, you can move it over a bit until the window is visible, or reduce the image size by zooming to a value less than 100%. Click on this star, and a circle will appear around it. Now click the **Star 1** button, and a yellow circle with the number 1 below it will appear around the star.

There is another good candidate at X = 1136, Y = 916. Click this star, and then click the **Star 2** button. A yellow circle will now appear around this star with the number 2 below it.

If for some reason you decide to try a different set of stars, the **Clear** button allows you to clear your selection and start over. Go ahead and click the **Clear** button and select the same stars again.

Don't worry about selecting the exact center of the star; **AIP4Win** has an excellent centroiding routine that mathematically determines the exact center of the star. It can, however, be fooled if the track radius is set so large that multiple stars fall within inside its boundaries. In this case, it will find the "center of mass" of the group of stars and center on that.

Notice that as you select each star, a message displaying the star's elongation is displayed in the Status box. Knowing this value can be used to help you select a value for the **Elongation Limit** when the **Skip frames with elongated stars** box is checked. This feature of **AIP4Win** will automatically reject frames with tracking errors. We will not be using this feature in this tutorial, as the set of images we are using is quite well tracked.

Notice also that the **Track Radius** control is disabled once you select the first alignment star. If you need to adjust this value, you will need to click the **Clear** button in order make this control active again.

If you wish to manually inspect your star images as you select them, use the Magnifying Glass tool to zoom in on a selected region of the image without changing how it is displayed.

**Step 8: Start the Track & Stack Operation.** Now we are ready to start the Track & Stack operation. For this tutorial we will use the Manual Slave Alignment Star Selection option. Make sure the **Manual** button is clicked. In manual mode, the user must select the alignment stars in each subsequent image. A guide is provided to show you where **AIP4Win** thinks the guide star is. If the images are well-tracked, and the Track Radius is set large enough, you can use Automatic mode and **AIP4Win** will find the stars for you.

Now click the **OK** button, and you will be presented with each of the remaining images in turn. A **Manual Alignment** panel will also appear as an extension at the bottom of the AutoProcess Multiple Images tool window. The stars you selected on the master image are displayed with numbered yellow circles.

If the yellow circles contain your alignment star selections, all you need to do is to click **OK**. If they don't line up, just click the correct stars and the **Star 1** or **Star 2** button, and then click **OK**. You can correct the position of either star.

If an image shows signs of obvious tracking errors, satellite trails, or has other defects, you can click **Skip** to skip it. Skipped images are not added to the stack.

This image set was nicely guided, and **AIP4Win** will follow the alignment stars as best it can as they drift across the frame from image to image, so you will most likely be able to click **OK** for each image without needing to correct the alignment star position.

When it is finished you will see a nice, deep image of the Waterfall portion of the Veil Nebula that has much less noise than any of the individual images in the set. Right-click on the image to bring up the Image Status window and select the **History** tab. A description of the "Track and Stack" operation is stored in the FITS header when you save the file. This record keeping is performed for each tool in **AIP4Win** and allows you to see how each image was processed.

Click the **Close** button to dismiss the Auto-Process window. Restore the Auto-Process Log and scroll down through it. You will see the coordinates of the alignment stars in each image and see how the images were translated, rotated and scaled to bring them into alignment.

### C.9.2 Multi-Image Processing

With the Multi-Image feature, **AIP4Win** provides you with the ability to process a whole night's worth of images automatically. Multi-Image processing is performed similarly to the Track & Stack processing. Instead of selecting the **Stack** process type, if you select **Individual Files**, the images will be calibrated and saved to separate files instead of being summed together. The Auto-Process Tool provides a set of the same basic image enhancement functions we used in the Image Enhancement tutorial, and they can be applied automatically to each image.

When you choose individual image processing, you are presented with the ability to choose a format to save your images, and to select where they are saved.

### C.9.3 Multi-Image Alignment

You can also choose to align a set of images without stacking them. This is provided so that you can combine the images using the Movie Tool into an AVI movie that can be played back on any PC running Windows. Several movies have been provided on the CD-ROM that have been created using this technique.

Multi-Image Alignment is performed when you select **Individual Files** for the Process Type and select **1 Star** or **2 Star** Alignment and select a Master Frame. Other than that, it is identical to Multi-Image Processing as described above.

## C.10 Image Registration and Blinking

In this tutorial you will learn how to align two images into accurate sub-pixel registration; and then blink them to search for asteroids.

## Appendix C: Tutorials

The image files can be found in the Image Registration subdirectory of the Tutorial directory on the CD-ROM. We used one of these files earlier in the Astrometry tutorial (Section C.4). They consist of a pair of images, taken just 26 minutes apart, of a trio of asteroids. We will use the Register Images Tool to align one image with the other; then we will use the Blink Images Tool to quickly alternate the view between the two. The result will be that the asteroids will appear to hop back and forth while everything else stays in place.

For this tutorial, be sure that **Auto Lo/HiStretch** is selected as the default display mode in the Image Display Control window.

**Step 1: Load the Images.** Load the images “ah102dad.fts” and “ah102dcd.fts” from the Image Registration tutorial directory.

**Step 2: Invoke the Register Images Tool.** Click on the *Multi-Image|Register Images...* menu item and the Image Registration Tool window will appear.

**Step 3: Select the Reference and Subject Images.** The Register Images Tool window has a drop-down listbox labelled **Master Image**. In this drop-down listbox select the “ah102dad.fts” image. Click the **Select Slave Images** button, and a dialog pops up with a pair of lists. The left list shows all the currently loaded images. The right list is empty. Select “ah102dcd.fts” from the left list and click the **Add>>** button. The Register Images tool can register a number of images to a common master. For this tutorial we will only be registering one. Click **OK** when this is done.

**Step 4: Register the images.** Click the **Register Images** button and the master image (Ah102dad.fts) will be activated (its title bar will be highlighted). Find a clean star image in this image, click on it, and then click the **Star 1** button. Next select a second star and click the **Star 2** button. Then click the **OK** button. A pair of green circles will identify the stars you selected. The first slave image will now be activated, and a pair of yellow circles will identify the positions of the stars you selected from the master image. If the selected stars are inside the circles, click **OK**; otherwise, identify and mark the correct stars and click **OK**. The slave image will now be aligned in register with the master. If multiple slave images were specified, you would be presented with each image in turn until they were all aligned.

To get the best image registration, select alignment stars that are as far apart as possible. When the subject image is aligned, it is translated, rotated and scaled to match the reference image. Selecting stars that are too close together will reduce the accuracy of the calculations used to determine the rotation to be applied.

**Step 5: Invoke the Blink Images Tool.** Click on the *Multi-Image|Blink Images...* menu item and the Blink Images Tool window will appear.

**Step 6: Select the Images to Blink.** The Blink Images Tool window contains two drop-down list boxes, one for each image. In the top list, select the original “ah102dad” image. In the lower list, select the registered image.

**Step 7: Blink the Images.** Click **Manual** several times and you will see the asteroids in the image hop back and forth. If you look carefully, you will see three asteroids hopping and a cosmic ray hit in one of the images blinking on and off.

**Step 8: Auto-Blink.** Click the **Auto** button and the images will blink at a rate set by the **Blink Interval** slider. Clicking the **Stop** button will return the mode to manual blinking.

Close all open images when you are finished.

## C.11 Deep-Sky Images

Up to this point, the tutorials have been focussed on specific features of **AIP4Win**. This and the following tutorial demonstrate using a group of **AIP4Win** functions to solve typical image processing challenges. This tutorial focuses on enhancing deep-sky images.

The task to be performed in this tutorial is the calibration and enhancement of a typical track-and-stacked deep-sky image. Images of this type share a number of common characteristics.

- Since they are images of faint objects, they are usually fairly long exposures (as CCD exposures go).
- They typically have a wide dynamic range (especially nebulous objects).
- The pixels in the image are dominated by the sky background (i.e., most of the pixels are black).

The common characteristics of deep-sky images lend themselves to a standard processing flow which consists of:

1. Calibration
2. Evaluation
3. Enhancement (assuming that the images are not being acquired for photometric or astrometric purposes).

The type of enhancement that will be performed depends on:

- the subject of the image (nebulosity vs. stars),
- the amount of visible noise, and
- the dynamic range of the image and how it is distributed across the range of available pixel values.

Due to the length of the exposure, and the desire to remove system-related artifacts from the image, the diligent CCD user will have taken the requisite calibration frames, including:

- dark frames of duration at least equal to the image frames,
- flat frames,
- flat-dark frames,
- and, optionally, a set of bias frames.

Trying to enhance a raw CCD image can be a futile endeavor without calibration frames. For most CCD cameras, the dark current pattern tends to overwhelm the

## Appendix C: Tutorials

low-level detail which we are trying to bring out, and you end up with a noisy-looking image.

The image set which we are going to work on here consists of four 4-minute exposures of the Helix Nebula taken with a Cookbook 245 CCD camera w/LDC, attached to an 11-inch  $f/6.3$  telescope. These files are found in the Deep Sky Images subdirectory of the Tutorial directory on the CD-ROM. This image set is the white-light portion of a much larger set of color-component frames which are found in the Images\McMickle\ColorHelix directory on the CD-ROM. Included with the four exposures are darks, flats and flat-darks. These CB245 files are saved with a fixed bias value of 100, so bias frames are not needed.

**Step 1: Examine the Raw Images.** Click the **Load an Image File** button on the toolbar, open the files “Helw1.fts” through “Helw4.fts,” and briefly examine each frame. Make sure that the **Auto Low/High Stretch** is the default display mode and that the image is stretched sufficiently to make its structure visible. What you will see is a faint image of the Helix Nebula sprinkled liberally with hot pixels. This is a pretty typical raw image frame. Notice that the stars are well formed and show no signs of trailing over the duration of their four-minute exposures. This is an indication of good guiding. Notice the absence of any pattern noise, such as horizontal banding, which would indicate electrical interference. Notice the darkening in the lower corners due to vignetting in the optical system.

It is good to give the raw frames at least a cursory glance before they are combined into an image, to avoid using frames with obvious flaws. It is also worthwhile to check them from time to time just to see how well the camera is functioning. Close the files when you are finished.

**Step 2: Examine the Calibration Frames.** In the same manner as described above, open the files “1240drk.fts” through “5240drk.fts.” Compare them to each other. They should all appear consistent. They should appear as a field of white dots on a dark gray or black background, with some slight electroluminescence from the output amplifier visible on the left edge. When you are finished examining these files, close them.

Open the files “Flat001.fts” through “Flat010.fts.” You will see a series of typical flat frames, with the same signs of vignetting that were visible in the image frames. These flats are only 11 seconds long, and the camera was well-chilled and stable, so they show very little dark current noise. When you are finished examining these files, close them.

Now open “Drflt001.fts” through “Drflt010.fts,” and you will see a “salt and pepper” noise pattern typical of a short dark frame for the CB245 camera with the Low Dark Current feature turned off. If you run the cursor around the image, you will see that the variation from pixel to pixel is only a few ADUs. If you fire up the Pixel Tool and sample groups of pixels randomly, you will see a standard deviation of less than one. This shows that the frame is pretty consistent. When you are finished examining these files, close any open images.

Examining your calibration frames is a good practice to make sure that no pathological problems exist. Any problems that appear in your calibration frames

will appear in any images you process with them. Weed out the defective frames now, before they cause you any trouble.

Fortunately, this is a high-quality image set, and you will not need to discard any frames.

**Step 3: Set Up Advanced Calibration.** The advanced calibration protocol allows us to use the automatic dark matching feature of **AIP4Win**, and it is the preferred method of calibrating images. It only requires that the bias value of the camera be known, or that a bias frame be available. It allows a set of dark frames to be used that may be of a different (preferably longer) exposure time or taken at a slightly different temperature.

Click the **Setup Calibration** button on the toolbar, or use the menu item to open the Calibration Setup window. When the window opens, select the **Advanced** calibration protocol and activate the **Bias** tab, and select the **Use Constant** bias button and leave the value at its default of 100. Activate the **Dark** tab and click the **Select Dark frame(s)** button; then select the files “1240drk.fts” through “5240drk.fts.” After you click **Open** in the file dialog box, the green light next to the **Select Dark frames(s)** button should go on. Click the **Median Combine** and **Automatic Dark Matching** radio buttons; then click the **Process Dark Frame(s)** button. The dark frames you selected will now be median-combined, and a table of all the hot pixels will be created for use in the dark matching function. Click the **Process Dark Frame(s)** button, and you will see that the **Subtract Dark Frame** box is checked.

Next set up the flat-frame correction. Select the **Flat** tab and then click the **Select Flat Frame(s)** button to invoke the Select Flat frames dialog. Select the files “Flat001.fts” through “Flat010.fts” and click **Open**; the dialog box will close and the green light next to the **Select Flat Frame(s)** button will turn on. Click the **Median Combine** radio button in the Flatframe Correction area to set this mode for combining the flat frames. Check the **Subtract Flat-Dark** box, and the **Select Flat-Darks** button will become enabled. Click it and select the files “Drflt001.fts” through “Drflt010.fts” for use as flat-darks and click **Open**. Select the **Median Combine** radio button in the Flat-Darks area as well. Finally, click the **Process Flat Frame(s)** button, and the master flat frame will be created, finishing your calibration setup.

Since we are not performing defect correction for this exercise, this completes the setup for calibration. Dismiss the Calibration Setup window by clicking the **Close** button.

**Step 4: Set Up and Run the Deep Sky Auto-Process Tool.** To bring up the Deep Sky Auto-Process Tool, click on *Multi-Image|Auto-Process|Deep Sky....* Select the files “Helw1.fts” through “Helw4.fts,” and set the **Process Type** to **Average Stack**. Initialize the following controls to their indicated settings:

**Calibrate Image** should be checked

**2X Resample** should be checked

**Prescale** should be left unchecked

**Scale Factor** is not used

## Appendix C: Tutorials

**Noise Filter** should be left unchecked.

Now activate the **Alignment** tab and select the file “Helw1.fts” on the **Select Master Frame** drop-down list. Select **2 Star** for the alignment mode and set the **Track Radius** to 12.

The Master Image window should now be visible; click on the star at X = 58, Y = 67, and click the **Star 1** button to choose it as your first alignment star. Pick the star at X = 680, Y = 407 as **Star 2**. If it is not already selected, select **Automatic** as the **Slave Alignment Star Selection** mode. These images were guided well, so we will let **AIP4Win** take care of stacking these images for us. Click **OK**.

Watch as **AIP4Win** flashes each image on the screen, calibrates it, and produces a stacked image. The process is finished when the image, now two times larger, is displayed as an image labelled “Track & Stack: Average of 4 images.” You can now close the AutoProcess Multiple Images window; it has completed the job you asked it to do.

**Step 5: Square Up the Image Pixels.** Since this image has been resampled by 2x during processing, it no longer conforms to any known CCD camera chip size. If this image is ever read back into this program, or any other image processor, it will be displayed with an incorrect aspect ratio, because the program will not know the pixel size. To prevent this, we need to resample the image to square up the pixels.

Click on the *Transform|Resample...* menu item to bring up the Resample Tool. Make sure the **By Percentage** tab is active and set the **Resample to:** value to 100%, if it not already set. Check the **Make Pixels Square** box and click **Resample Image**. A new image will be created in which the pixels are square. Because **AIP4Win** automatically displays images with the correct aspect ratio, where possible, the image will not look any different; but if you bring up the Image Status tool by clicking the *Measure|Statistics|Image...* menu item and look on the **Specs** tab, you will see that the original image was 756 x 484, and the resampled image is 756 x 562.

**Step 6: Save the Stacked Image.** Save the image you just created in your C:\Windows\Temp directory, or another of your choosing. Click the **Save Image in FITS Format** button on the toolbar. (This is the picture of the floppy disk with a red “F” on it. A Tool Tip will appear if you float the cursor over the button.) The Save Image as FITS File window will appear, giving you a choice of FITS formats, as well as the option to edit the various FITS header fields and save the file.

If you click on the **Examine/Edit FITS Header** button, you will see the contents of the FITS header for the file you are saving. Included in the FITS header you will see a complete processing history of the file. This is exceptionally useful information to have when you are trying to reproduce a processing sequence later and need to know how you got a certain result. You can click on the *Edit|FITS Header...* menu item to bring up the FITS header with the processing history of any image on the screen, whether or not it has been saved as a FITS file.

**Step 7: Evaluate the Image.** Now that the image has been calibrated and

stacked, you can evaluate it. Upon visual examination you see that it is a bit noisy, but not objectionably so. Some of this is due to the stacking process, in which the noise adds with the square root of the number of images stacked. This is unavoidable unless you just want to make a single, longer exposure. You will notice a blooming trail in the upper right corner of the image. This trail imposes limits on any single exposure, and will get objectionably longer and brighter as the exposure time increases. So this exposure is about as long as you want for an image that has that bright star in the frame.

Another thing you will notice about the image is the artifacts along the top and left edges due to the stacking process. The scope drifted about 10 pixels to the right and about 10 pixels down between the first and last exposure. This left an area along these two edges that does not contain contributions from all of the images. The pixel values encountered in these regions throw off the statistics for the image and can often cause the Auto-Stretch function to display it incorrectly. It will also affect image enhancement tools that rely on image statistics to do their jobs. You will usually want to crop these edges.

**Step 8: Crop Off the Edges.** When a series of images is stacked, there is usually a band of pixels along one or two adjacent edges that has values inconsistent with those in the rest of the image. This results from areas where the source images did not completely overlap due to drifting between the exposures. It helps to crop these edges off so that they do not affect measurements made on the image. It also results in a more cosmetically pleasing image.

Invoke the Crop Tool by clicking the *Transform|Crop...* menu item. You may want to move this tool to the left edge of the screen to give yourself some room. Make sure that the **Select With Mouse** radio button is active, press the left mouse button and scroll the mouse across a region of the image, excluding the edge pixels. You can repeatedly select the region until you are satisfied with it. Fine adjustments of its location can be made using the arrow keys on your keyboard. When you have selected the area, just click the **Apply** button and a new, cropped image will be created.

**Step 9: Measure the Image.** Let's make some measurements on this image. Invoke the Statistics Tool by right-clicking the image button. This tool will show you some things about the image that you cannot easily visualize. First notice the Min and Max Pixel Values. They show a reasonably wide range, over 2,400 ADUs wide, depending on the region that you cropped. This range will give us a good deal of data to work with. You can better visualize this if you invoke the Histogram Tool to display the image histogram.

**Step 10: Histogram Shaping.** You can use the Histogram Shaping Tool to bring out more detail in the image by redistributing the histogram to create more contrast between the available brightness levels. Invoke this tool and try some of the available histogram shapes to see which ones provide the most pleasing results.

**Step 11: Brightness Scaling.** This is another useful technique for bringing out image detail. Invoke this tool and experiment with the different scaling types

## Appendix C: Tutorials

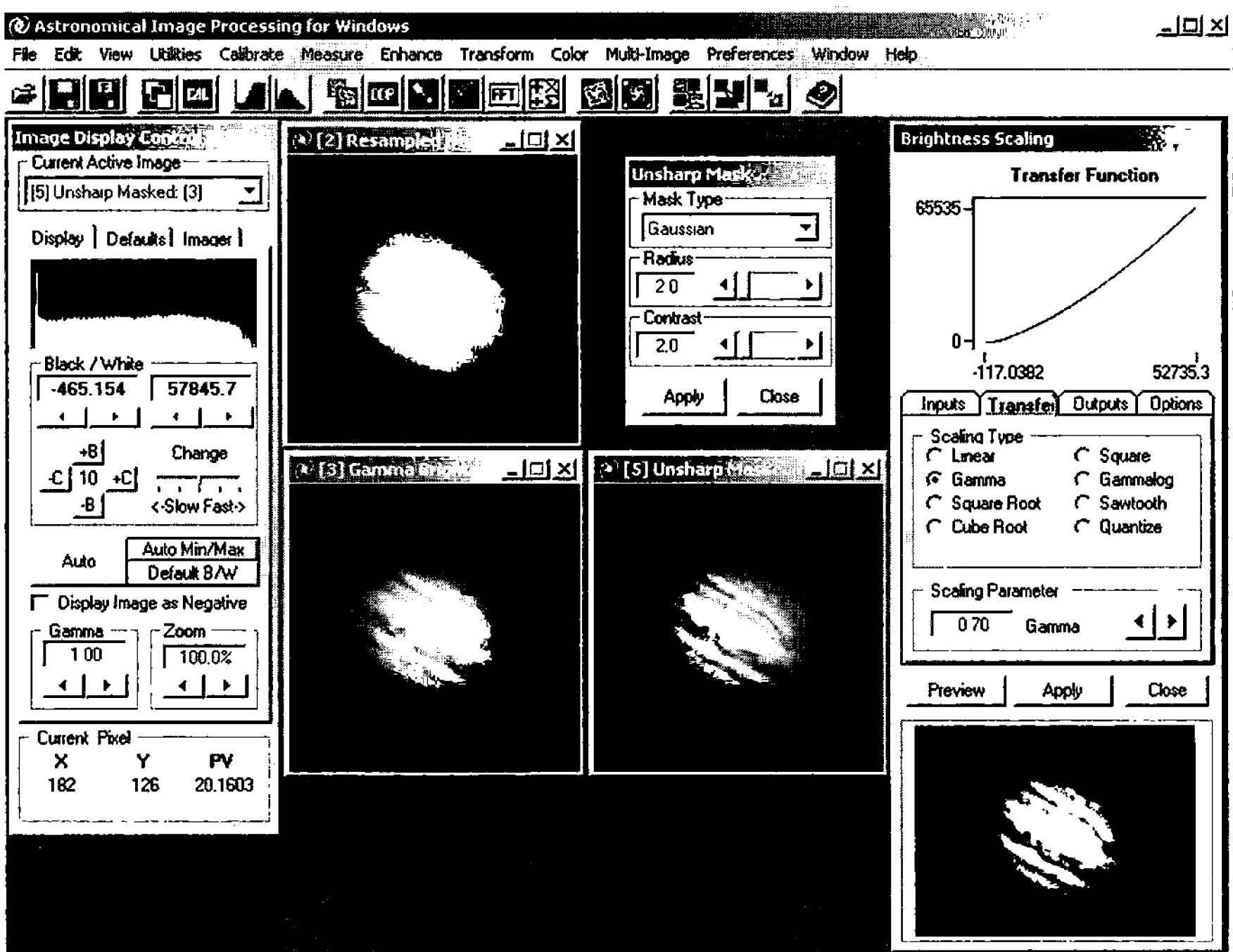


Figure C.11 Before you apply the powerful image processing techniques to them, images of the planets almost always look soft and fuzzy. In this tutorial, you will see how nonlinear scaling, unsharp masking, and deconvolution reveal the true aesthetic and information content of planetary images.

while adjusting the scaling range. The preview feature of this tool and the Histogram Shaping Tool make it easy to evaluate your settings quickly.

## C.12 Planetary Images

This tutorial shows you how to process planetary images using **AIP4Win**. These share characteristics that make them quite different from deep-sky images:

- The exposures are short (although they still require dark-frame subtraction to remove bias).
- They lack point sources (like stars) which makes them hard to register to each other.
- The planets rotate rapidly enough that features in different exposures may not line up.
- They often contain high-contrast features blurred by atmospheric seeing.
- You usually end up shooting a lot of them in order to sort out the ones that show the best seeing.

These features are common to most whole-disk images. Lunar images present

their own special challenges.

For this tutorial we will start with processing an image of Jupiter. This image can be found in the Planetary Images subdirectory of the Tutorial directory on the CD-ROM.

**Step 1: Load the Image.** Load the image “M0611029.fts.” If you have Auto-Scale turned on, the planet should appear very bright and washed out. You can enhance the contrast by moving the **White** control up a bit, and the **Gamma** control down a bit, until the belts are more visible. We will use the Brightness Scaling Tool later to make this change permanent.

**Step 2: Center the Planet.** The planet was not centered well on this image. This is pretty typical when you are shooting large numbers of planetary images, searching for that fleeting moment of perfect seeing. Click on the *Transform|Center Planet...* menu item and a small window will pop up asking you to set the planetary disk threshold. This sets the threshold percentage of pixels values above which the edge of the disk is recognized. For this image, the default of 20% will work just fine. Click the **OK** button, and the planet’s disk will be centered on the image.

**Step 3: Square Up the Pixels.** This image has highly non-square pixels, which is not apparent, because **AIP4Win** automatically displays images corrected for their pixel aspect ratio. Invoke *Transform|Resample...* and square up the pixels like we did in step 5 of the Deep Sky Images tutorial.

**Step 4: Brightness Scale the Image.** This image has literally thousands of gray levels in it. Unfortunately, neither the computer monitor nor our eyes can distinguish more than 256 levels. Therefore, to see image features, it is necessary to enhance the image; that is, to bring the features we want to see within the range of 256 levels that we *can* see.

We can improve the contrast and bring out the belts by applying a gamma brightness scaling. For the typical deep-sky image, we generally use values of gamma greater than one to bring out the faint details. Our planetary detail in this image, however, is darker than the background. For this, values of gamma less than one are more effective.

Invoke the Brightness Scaling Tool using the toolbar button or the menu. Click the **Auto** button on the **Inputs** tab of the Brightness Scaling window to set the initial **Low** and **High Pixel Values**.

If you click on the sky background in the image somewhere close to X = 15, Y = 30, you will see that its level is about 0.5 ADUs (as a result of the interpolation that occurred when the planet was centered). Type the value 0.5 into the **Low** text box in the Brightness Scaling Window. Bring the high end up a bit by increasing the **High** pixel value to around 3200.

Select the **Transfer** tab; click on the **Gamma** scaling button, and the **Gamma** control will appear. Adjust the Gamma value down to around 0.7 and observe the **Preview** display. You will see a wealth of detail suddenly pop into view. Now click **Apply** and a new image will be created.

## Appendix C: Tutorials

**Step 5: Apply an Unsharp Mask.** The planetary features are still a bit washed out and fuzzy in this image. Applying an unsharp mask will increase the contrast. Invoke the Unsharp Mask Tool, use the default **Radius** and **Contrast** values and click **Apply**. If the resulting image is not contrasty enough, try it again with a larger radius or with the contrast level increased.

**Step 6: Try Deconvolution.** Deconvolution can also help bring out detail and remove the effects of poor seeing. Click on the title bar of the Gamma scaled image to make it the active image again and invoke the Deconvolution Tool. Set the tool up as follows:

**Point Spread Function:** Gaussian PSF

**Sigma:** 2

**Deconvolution Type:** van Cittert

**Number of Iterations:** 16

**Relaxation Parameter:** 0.1

**Display Iterations:** checked

**Process high-frequency components only:** checked

Click **Execute Deconvolution**. Shortly, a new image will appear in which the belts practically “jump right off the screen.” This is a very powerful technique for planetary processing.

Compare the original image with the results you obtained using these techniques. An image that you might have discarded as bland ended up revealing a tremendous amount of detail.

## C.13 Color Images

This tutorial demonstrates the basic color operations of **AIP4Win**.

The files used in it may be found in the Color subdirectory of the Tutorial directory on the CD-ROM. The files comprise a calibrated RGB image set of the Eagle Nebula, M16, taken by Neil McMickle of Stanhope, New Jersey, using a MX716 CCD camera and a set of RGB dichroic filters and an IR-blocking filter through a 4-inch fluorite apochromatic refractor. In this tutorial we will register four individual filtered exposures and combine them into a single, color image.

**Step 1: Load the Component Images.** Load the four images found in the Color Images tutorial directory. They consist of the following:

- M16B.fts: the red-filtered exposure,
- M16G.fts: the green-filtered exposure,
- M16R.fts: the blue-filtered exposure, and
- M16L.fts: a white-light exposure with an IR-blocking filter.

The white-light exposure is not required, and can be synthesized from the red, green and blue images; but using a separate, deeper white-light image will result in a more detailed final color image.

**Step 2: Register the Images.** Click on the *Multi-Image|Register Images...*

menu item and the Image Registration Tool window will appear. Register the red, green and blue images using the white-light image (M16L.fts) as the master, just as we did in the Image Registration and Blinking tutorial.

**Step 3: Invoke the Join Colors Tool.** Click on the *Color|Join Colors Tool...* menu item and the Join Colors Tool window will open. (You will probably want to move this tool to the right edge of the screen so it won't obstruct your view of the image.) When the tool opens you will see a set of four drop-down list boxes labelled **Red**, **Green**, **Blue** and **Luminance**. Select, for the respective images:

M16R.fts for the **Red**,

M16G.fts for the **Green**,

M16B.fts for the **Blue**, and

M16L.fts for the **Luminance**.

**Step 4: Create a Color Image.** **AIP4Win** provides the capability of automatically balancing your color images. Create a color image using the color channels you selected by selecting the **Automatic Color** radio button and then clicking the **Make Color Image** button. A color image of M16 is created.

**Step 5: Adjusting the Brightness.** This image looks pretty dark. Select the **Automatic** tab and set the **Image Brightness** control to  $\sim 0.993$  and the **Histogram HiPoint** control to  $\sim 0.998$ , and click the **Refresh Current Color Image** button. The resulting image should look much brighter. Select the Adjuster tab and slide the Gamma slider to get a value of  $\sim 1.7$  and click the **Refresh Current Color Image** button again; you will see that the faint detail is more prominent. Note that while we were doing this, the color balance remained unaffected. One of the powerful features of **AIP4Win**'s color tools is the ability to adjust the chrominance components (hue and saturation) independently from the brightness component (luminance).

**Step 6: Cleaning Up Color Artifacts.** This image looks nice and bright, and there is a lot of detail in the background; but some of the stars have blue rings around their centers. Two things contribute to this: the chromatic aberration present even in an apochromatic refractor, and the fact that these stars are saturated. **AIP4Win** provides a useful tool for controlling these artifacts, Saturation Rolloffs and Cutoffs. Check the **Apply Rolloffs and Cutoffs** box and click the **Refresh Current Color Image** button. You will see that the false color has been removed from the brighter stars (as well as from the brightest parts of the nebula). This can be adjusted by increasing the **High Rolloff** value to about 245 and clicking the **Refresh Current Color Image** button again.

**Step 7: Crop the Color Image.** Quite often, there is an area around the image border in which the color channels did not overlap when they were registered, due to the shifting of the telescope between the color-filtered exposures. You can crop off the edges of the image using the **Crop Tool**. Just click on the color image and then click the *Transform|Crop...* menu item; the **Crop Tool** window will appear. Use the mouse to select the region you wish to keep and click the **Apply** button. A new image will be created from the selected region. You can now close the Join Colors Tool.

## Appendix C: Tutorials

**Step 8: Save the Color Image.** When you are satisfied with your results, the color image you just created can be saved in a huge variety of formats. Just click the *File|Export...* menu item, and a dialog box will appear to prompt you for the file name and location. You can save your file in JPEG format for posting to the web, or in 8 or 16-bit TIFF format for import into PhotoShop. (When you plan on using 16-bit TIFF, be sure to check the **0 to 65535 (48-bit Color)** option in the **Output Range** selection on the Join Colors Tool before creating your color image.)

If you want to preserve the greatest flexibility for future editing of your image, it is recommended that you save it as a 32-bit floating point FITS file. This file format preserves the full dynamic range of your data along with all the color detail. The image can be reloaded any time later, and edited using the Color Image Tool.

**Step 9: Later Adjustment of the Color Image.** As you look at the color image you just created, you notice that the color in the nebula looks a bit too saturated, giving it an unnatural appearance. You can fix this with the Color Image Tool. Click on the *Color|Color Image Tool...* menu item to invoke the tool. You will see that it has a similar layout and most of the same controls as the Join Colors Tool, but it is intended to work on images where the color files have already been joined. Notice that the icon in the upper left corner of your color image is itself in color. You will also notice that grayscale images have a grayscale icon in the corner. This feature helps you keep track of your image types—as not all tools operate on color images—and they will be grayed out on the menu and toolbar.

On the Color Image Tool, select the **Adjusters** tab. Slide the **Color Saturation** control down to a value of 0.91, and click the **Make New Color Image** button. A new image is created with a more realistic-looking level of saturation. If you slide the **Color Saturation** control all the way down to 0 and click the **Refresh Current Color Image** button, you will see that your image looks like a grayscale image, as you have removed all the color from it. You can just slide the slider back up to 0.91 and click the button again; your color image will be restored.

## C.14 Conclusion

These tutorials barely scratch the surface of **AIP4Win**'s capabilities. Experiment with the tools to get familiar with them and what they can do for your images. You will soon learn that no one tool does everything, and that the *real* power in image processing lies in learning how to use tools together.

Remember, too, that **AIP4Win**'s help file is just a keystroke away. It provides information on each of the tools in this program. The help tools are context-sensitive; all you need to do is press the **F1** key while tool is open, and the help tools automatically bring up the information for that tool.

The CD-ROM included with this book contains hundreds of test images which run the gamut—from those taken by amateurs with modest equipment through images from professional observatories. Be sure to give them a try with **AIP4Win**.

# Index

## Numerics

24-bit color 558

## A

à trous algorithm 478

aberration 250

aberrations 5

ADU 23, 40

afocal 9

air mass 288

Airy diffraction figure 499

Airy disk 7

angular diameter of 7

all-sky photometry 279, 290, 296

analog-to-digital unit 40, 44

Ångström, Anders 304

angular 7

angular field of view 26

angular field of view for CCDs 79

angular pixel size for CCDs 79

annulus 274

Anscombe transform 488

anti-blooming gates 21

apparent positions 250

archiving digital information 151

asteroid astrometry 265

astrometric catalogs 249

astrometric position 250

astrometry 221, 249

atmospheric extinction 286

atmospheric turbulence 499

AVI 57

## B

back-illuminated CCDs 18

Barlow lens projection 145

Barnard's Star 250

barycenter 216

bas-relief operators 385

Bayer array 21, 558

bias frames 124, 165

binary data array 66

BITPIX (FITS keyword) 61

black body radiation 516

blinking 447

blocking filters 549

blooming trail 21

blue bloat 85

BMP 57

boxcar unsharp mask 390

Bunsen, Robert 303

burst noise 242

Butterworth filter 472

byte order 58

## C

calibration

advanced protocol 123, 158

basic protocol 123, 157

errors 199

bias drift 199

changing CCD temperature 201

changing optical configuration 202

noise 196

standard protocol 123, 158

camera, digital SLR 128

Canon 10D 557

Canon Digital Rebel 557

Cape Photographic Catalog 251

CCD 14

CCD camera's performance

charge skimming 228

conversion factor 228

dark current 228

interference 228

linearity 228

noise pickup 228

readout noise 228

uniformity 228

CCDs

back-illuminated 18

frame-transfer 17

## Index

- front illuminated 18
- reflections 104
- sizes 83
- centroid determination 216
- Cepheid variables 296
- charge skimming 243
- chrominance 514
- chrominance component 513
- clock drive
  - testing 95
  - tuning 95
- CMOS 14
- CMY filters 554
- color
  - wavelengths 515
- color channel 573
- color plane 573
- color space 539
- comet astrometry 265
- COMMENT (FITS keyword) 61
- computer-controlled mountings 93
- constant sigma scaling 489
- conversion factor 44, 227
- convolution 365, 464
  - commutivity 376
  - linearity 375
  - separability 377
- Cooley, P. M. 469
- coordinate reference frame 251
- Cousins, Andrew 282
- cropping 332
- CRW 57
- cylinder focusing 103
- D**
  - dark current 43, 162, 227
    - determining 247
  - dark frame 46, 170
    - matching 176
    - scalable 174
    - standard 174
    - taking 124
  - dark frames 43
  - Davis, Don 304
  - deconvolution 499
  - deconvolve 368
  - deep-sky imaging
    - guided one-shot imaging 132
    - track-and-stack imaging 132
  - deep-sky objects 130
  - deep-sky targets
- clusters of galaxies 139
- comets 140
- dark nebulae 137
- galaxies 138
- globular clusters 136
- HII regions 136
- open clusters 135
- planetary nebulae 136
- quasars 140
- reflection nebulae 137
- defect mapping and correction 203
- detector 9
- detector limited 49
- determining a centroid 216
- differential photometry 278, 290, 292, 293
- diffraction grating 305
- diffraction-limited images 6
- digital
  - development 422
  - setting circles 93
- digital number 44
- digital single-lens-reflex (dSLR)
  - camera 557
- digital SLR camera 128
- digital snapshots 130
- discrete Fourier series 466
- dispersion 312
- DN 23
- dome flats 184
- do-what-you-can photometry 292, 298
- drift subtract mode 168
- dust donuts 102, 107
- dyadic 480
- E**
  - eclipsing binary stars 294
  - edge-detection kernels 385
  - electroluminescence 178
  - electronic transitions 517
  - electrons per ADU 44
  - epoch 250
  - erratic errors 94
  - expectation maximization method 507
  - exponential histogram shaping 363
  - extinction values 532
  - extrafocal star test 107
  - eyepiece projection 144
- F**
  - fast Fourier transform 469
  - feature masking 474

- fiber-fed spectrograph 310  
 fiber-fed spectrograph images 318  
 field flooding 102, 104  
 field of view 77  
 filter matrix 537  
 finder scopes 92  
 finders  
     big 92  
     flip-mirror systems 92  
     unity-power 92  
 FITS 57, 59  
     Basic FITS 59  
     binary image data 60  
     header 59, 61  
     keyword 61  
     tailer 60  
 FITS variants 68  
     nonconforming data arrays 68  
     nonconforming headers 68  
     un-corrdinated header and data arrays 68  
 FK3 251  
 FK4 251  
 FK5 251  
 flat-field frame 125, 180  
     dome flats 184  
     light-box 184  
     sky flats 184  
     twilight flats 184  
 flats 186  
     flipping 331  
     floating 332  
     flopping 331  
 focus  
     methods  
         diffraction-spike 120  
         holes-in-a-mask 121  
         longest blooming trail 120  
         number of faint stars 120  
         peak pixel value 119  
         smallest star image 119  
 Fourier integral 458  
 Fourier series 456  
 Fourier transform  
     inverse scaling 463  
     linearity 462  
     symmetry 462  
 Fourier, Jean Baptiste Joseph 455  
 Fraunhofer, Joseph 303  
 free spectral range 312  
 front-illuminated CCDs 18  
 full width halfmaximum 273  
     FWHM 273
- G**
- G2V stars 531  
 gain 44, 227  
 gamma transfer function 349  
 gammalog transfer function 351  
 Gaussian  
     histogram 361  
     histogram shaping 362  
     unsharp mask 392  
 Gaussian distribution 39, 42  
 Gavin, Maurice 304  
 geometric transforms  
     cropping 332  
     flipping 331  
     floating 332  
     flopping 331  
     resampling 334  
     rotation 325  
     scaling 328  
     translation 321  
 gradient kernels 385  
 gratings  
     reflection 305  
     transmission 305  
*Guide Star Catalog (GSC)* 251  
 guiding methods  
     off-axis 116
- H**
- H II regions 521  
 high-pass filters 381  
 high-point pixel values 210  
 high-resolution images 147  
 Hipparchos Catalog 252  
 histogram equalization 362  
 HISTORY (FITS keyword) 61  
 hot pixels 46, 163  
 hot spots 102, 103  
 how to  
     shoot dome flats 184  
     shoot light-box flats 184  
     shoot sky flats 186  
     shoot twilight flats 186  
 how to install *AIP for Windows* 613  
 HSL color space 541  
 hue 541  
 Huggins, William 304  
 human color vision 514  
 hydrogen hypersensitizing 13

# Index

## I

image  
    aspect ratio 23  
    calibration  
        advanced 191  
        basic 187  
        standard 189  
    feature analysis 212  
    histogram 210  
    profiles 220  
    stacking 199  
    times-five rule of thumb 172  
image calibration  
    advanced 441  
    basic 440  
    dark subtraction 436  
    flat-fielding 438  
    standard 441  
image estimation by iteration 501  
image math  
    absolute difference 429  
    addition 425  
    average 432  
    division 429  
    merge 430  
    multiplication 428  
    subtraction 426  
image ranking  
    maximum 435  
    median 433  
    minimum 435  
image registration  
    rotation 444  
    scaling 444  
    translation 443  
impulse function 485  
instrumental zero point 275  
interference filters 549  
interline-transfer CCDs 17  
internal lens reflections 104  
inverse convolution problem 500  
inverse wavelet transform 480  
iterative wavelet noise filter 492

## J

Johnson, Harold 282  
JPEG 57, 59

## K

Kirchhoff, Gustav 303  
Kirsch operators 386  
k-sigma filter 494

## L

Lab color space 544, 576  
Laplacian operator 388  
level 477  
light box 126  
light, definition 1  
light-box flats 184  
linear transfer function 347  
linearity 227  
line-detection operators 387  
logarithmic transfer function 350  
low point pixel values 210  
low-level light source (L3S) 229  
    construction 234  
low-pass filters 378  
LRGB imaging 546  
Lucy, L.B. 507  
Lucy-Richardson method 507  
luminance 514, 541

## M

master bias frame 169  
maximum likelihood method 507  
maximum pixel value 208  
mean of the median half 216  
mean pixel value 209, 214  
median pixel value 209, 216  
Menzies, John 282  
Mexican Hat function 477  
milk plastic 126  
minimum pixel value 208, 213  
Morgan, W.W. 282  
morphological operators 412  
    dilation 419  
    erosion 419  
    Frei and Chen 415  
    opening, and closing 419  
    skeleton 418  
multi-luminance) imaging 552  
multiple convolutions 371  
multiple-image processing 663  
multiplicative error 157  
multiresolution support image 490

## N

narrowband color images 553  
native color space 574  
nebulae spectral lines 518  
NEF file format 57  
negative transfer function 352  
Nikon D70 557  
1/f noise 242

- noise filters 410  
 noise pattern 46  
 non-linear enhancement  
   extreme value 406  
   local adaptive sharpening 408  
 non-linear operators  
   median 399  
   minimum and maximum 398  
   multiplicative rank 403  
   rank-order 401  
 normalized master flat frame 567  
 Nyquist  
   criterion 143  
   sampling theorem 29
- O**  
 objective prism images 314  
 objective prism spectrograph 305  
 objective prism, ray paths 307  
 observational error, analysis 268  
 Okano, Kunihiko 422  
 oversampling 29
- P**  
 Parseval's Theorem 466  
 periodic error 94  
   correction (PEC) 97  
 photographic emulsions 11  
 photometric  
   image profile 223  
   systems 279  
 photometry 221, 271  
 photon  
   flux 159  
   statistics 194  
 photon limited 49  
 pinhole camera 2  
 pixel 27  
   angular size 29  
   aspect ratio 23  
 pixel measurements  
   pixel coordinates 206  
   pixel value 207  
 pixel statistics 212  
 pixel value 22  
 Planck's constant 2  
 planetary nebulae 521  
 plate coordinates 254  
 Pogson scale 272  
 Pogson, Norman 271  
 point operations 337  
 point-spread function 29
- Poisson distribution 38  
 Poisson noise 488  
 polar alignment 113  
 power-law unsharp mask 393  
 precession 250  
 Prewitt operators 386  
 progressive-scan CCDs 17  
 proper motion 250
- Q**  
 quantize transfer function 353
- R**  
 rate errors 94  
 RAW file format 57  
 raw image 43  
 raw instrumental magnitude 275, 276  
 readout noise 43, 227  
 resampling 334  
 residual image 481  
 resolution 81  
 retina  
   cone cells 10  
   rod cells 9  
 retina, human 9  
 RGB filter matrix 538  
 RGB triad 23, 572  
 Richardson, William H. 507  
 Richardson-Lucy iteration 507  
 root mean square 41  
 rotation 325  
 RR Lyrae stars 296  
 RS Canis Venaticorum stars 296
- S**  
 sample 35  
 saturated 21  
 saturation 541  
 sawtooth transfer function 353  
 scaling 328, 357  
 shot noise 43  
 Sigma, as radius of star image 273  
 signal bias 41  
 signal-to-noise ratio 37  
 significance image 490  
 significant signal 486  
 Skiff, Brian 292  
 sky atlas software 93  
*Sky Catalog 2000.0* 251  
 sky flats 186  
 sky-limited image 48  
 slit spectrograph 309

# Index

slit spectrograph images 316  
*Smithsonian Astrophysical Observatory Catalog (SAO)* 251  
SNR 37  
Sobel operators 386  
solar-system targets  
    Jovian Satellites 152  
    Jupiter 152  
    Lunar Imaging 154  
    Mars 151  
    Mercury 151  
    Moon 154  
    Neptune 154  
    Saturn 154  
    Sun 155  
    Uranus 154  
    Venus 151  
spatial filters 377  
spatial frequency 453  
spatially encoded 572  
spectral resolution 312  
spectral sensitivity 83  
spectroscopy 225  
spectrum  
    analysis 319  
    calibration 319  
    image properties 311  
standard coordinates 253  
standard deviation 35, 209, 214  
stellar  
    parallax 268  
stretch scaling 356  
support frames 165  
synchrotron emission 518

## T

Tagged Image File Format (see TIFF)  
technique 113  
thermal frame 158, 173  
three-sigma detection 486  
TIFF 57, 59  
topographic operator  
    419  
Track 659  
track-and-stack imaging 447  
tracking sensitivity 93  
transfer curve 244  
transfer functions 343  
translation 321  
triangular unsharp mask 391  
trichromatic 514  
tri-color imaging 523

Tukey, J. W. 469  
tutorials  
    astrometry 615, 631  
    basic skills 615, 616  
    calibration 615, 621  
    deep sky images 615  
    fast Fourier transform 615, 655  
    image enhancement 615, 644  
        brightness scaling 644  
        convolution filtering 649  
        deconvolution 650  
        deep-sky images 665  
        histogram shaping 647  
        morphological processing 654  
        multiple image processing 659  
        planetary images 670  
        unsharp masking 650  
    image evaluation 615, 627  
    image registration and blinking 615, 663  
    multiple image processing 615  
    photometry 615, 635  
        multiple-image photometry 639  
        single-image photometry 638  
        single-star photometry 636  
    planetary images 615  
    spectroscopy 615, 642  
twilight flats 186  
Tycho catalog 252

## U

UBV system, primary standards 281  
UBV(RI) filters for CCD photometry 282  
undersampling 29  
unsharp mask 388  
USNO-A2.0 catalog 252  
USNO-SA2.0 catalog 252

## V

Van Cittert deconvolution 506  
Van Cittert, P.H. 506  
variance 214  
Vierte Fundamental Katalog 251  
vignetting 102, 106, 162

## W

wavelet coefficient 480  
wavelet function 479  
wavelet noise filter 485  
wavelet scale 477  
wavelet spatial filter 497  
wavelet spatial filtering 485  
wavelet transform 480

## Index

webcam 31  
white balance 576  
window reflections 104  
Wollaston, William 303  
Wratten filters 549

**Z**  
Z 275  
Z (zero point) 275

**Astronomical Image Processing for Windows 2.0 (AIP4Win 2.0)**  
**SOFTWARE LICENSE AGREEMENT AND WARRANTY STATEMENT**  
**(End-User Perpetual License - Redistribution Prohibited)**

The authors (Richard Berry and James Burnell) and the publisher (Willmann-Bell, Inc.) will license this Software to you only if you accept all of the terms contained in this agreement. Please read the terms carefully before continuing with this installation. Installing the Software indicates your acceptance of the terms; if you do not agree with the terms, do not install the Software.

1. The Software consists of the accompanying computer programs, data compilations, and documentation.
2. The license granted to you shall be perpetual unless terminated by written notice by you or terminated by either party for material breach. Upon termination of this license for any reason, you will immediately return to all copies of the Software to Willmann-Bell, Inc.
- 3a. A single-user license grants you non-exclusive rights to install and use the Software on two personal computers belonging to you and used by you. You agree that you are the primary user of the Software installed on your personal computers. You may copy the Software for archival purposes, provided that all copies must contain the original Software's proprietary notices in unaltered form.
- 3b. A multiple-user license grants you non-exclusive rights to install and use the Software on the number of computers specified in your purchase order and invoice. This license grants you the right to allow persons other than yourself to use the Software installed on the licensed computers. You may copy the Software for archival purposes, provided that all copies must contain the original Software's proprietary notices in unaltered form.
4. If you purchase a single-user license, you may purchase additional licenses to use the Software on additional personal computers, or you may purchase a multiple-user license.
5. The authors and its publisher own the Software and all intellectual property rights embodied therein, including copyrights and valuable trade secrets embodied in the Software's design and coding methodology. United States copyright laws and international treaty provisions protect the Software. This agreement provides you only those rights granted in this license, and no ownership of any intellectual property.
6. You agree: (i) not to modify or translate the Software; (ii) not to reverse engineer, decompile, or disassemble the Software, except to the extent this restriction is expressly prohibited by applicable law; (iii) not to create derivative works based on the Software; (iv) not to merge the Software with another product; (v) not to copy the Software, except as expressly provided above; or (vi) not to remove or obscure any proprietary rights notices or labels on the Software.
7. You agree not to transfer the Software or rights granted under this Agreement without the prior written consent of the Publisher, which consent shall not be unreasonably withheld. A condition to any transfer or assignment shall be that the recipient agrees to the terms of this Agreement. Any attempted transfer or assignment in violation of this provision shall be null and void.

# Software License Agreement and Warranty Statement

## LIMITED WARRANTY STATEMENT

**LIMITATION OF LIABILITY.** Authors warrant only to You that the Software shall perform substantially in accordance with accompanying documentation under normal use for a period of **NINETY (90) DAYS** from the purchase date. The entire and exclusive liability and remedy for breach of this Limited Warranty shall be, at Author's option, either (i) return of the list price of the Software, or (ii) replacement of defective Software and/or documentation provided the Software and/or documentation is returned to Willmann-Bell, Inc. with a copy of your purchase confirmation.

**THE AUTHORS AND THEIR PUBLISHER SPECIFICALLY DISCLAIM THE IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. THERE IS NO WARRANTY OR GUARANTEE THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED, ERROR-FREE, OR VIRUS-FREE, OR THAT THE SOFTWARE WILL MEET ANY PARTICULAR CRITERIA OF PERFORMANCE OR QUALITY EXCEPT AS EXPRESSLY PROVIDED IN THE LIMITED WARRANTY.**

No action for the above Limited Warranty may be commenced after one (1) year following the expiration date of the warranty. To the extent that this Warranty Statement is inconsistent with the jurisdiction where You use the Software, the Warranty Statement shall be deemed to be modified consistent with such local law. Under such local law, certain limitations may not apply, and you may have additional rights which vary from jurisdiction to jurisdiction. For example, some states in the United States and some jurisdictions outside the United States may: (i) preclude the disclaimers and limitations of this Warranty Statement from limiting the rights of a consumer; (ii) otherwise restrict the ability of a manufacturer to make such disclaimers or to impose such limitations; or (iii) grant the consumer additional legal rights, specify the duration of implied warranties which the manufacturer cannot disclaim, or prohibit limitations on how long an implied warranty lasts.

**INDEPENDENT OF THE FORGOING PROVISIONS, IN NO EVENT AND UNDER NO LEGAL THEORY, INCLUDING WITHOUT LIMITATION, TORT, CONTRACT, OR STRICT PRODUCTS LIABILITY, SHALL AUTHOR OR ANY OF ITS SUPPLIERS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND, INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER MALFUNCTION, OR ANY OTHER KIND OF COMMERCIAL DAMAGE, EVEN IF AUTHOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT PROHIBITED BY APPLICABLE LAW.**

**IN NO EVENT SHALL AUTHOR'S LIABILITY FOR ACTUAL DAMAGES FOR ANY CAUSE WHATSOEVER, AND REGARDLESS OF THE FORM OF ACTION, EXCEED THE AMOUNT OF THE PURCHASE PRICE PAID FOR THE SOFTWARE LICENSE.**

**EXPORT CONTROLS.** You agree to comply with all export laws and restrictions and regulations of the United States or foreign agencies or authorities, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. As applicable, each party shall obtain and bear all expenses relating to any necessary licenses and/or exemptions with respect to its own export of the Software from the U.S. Neither the Software nor the underlying information or technology may be electronically transmitted or otherwise exported or re-exported (i) into Cuba, Iran, Iraq, Libya, North Korea, Sudan, Syria or any other country subject to U.S. trade sanctions covering the Software, to individuals or entities controlled by such

## **Software License Agreement and Warranty Statement**

countries, or to nationals or residents of such countries other than nationals who are lawfully admitted permanent residents of countries not subject to such sanctions; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals and Blocked Persons or the U.S. Commerce Department's Table of Denial Orders. By downloading or using the Software, Licensee agrees to the foregoing and represents and warrants that it complies with these conditions.

**LICENSEE OUTSIDE THE U.S.** If You are located outside the U.S., then the following provisions shall apply: (i) Les parties aux presentes confirment leur volonte que cette convention de meme que tous les documents y compris tout avis qui siy rattache, soient rediges en langue anglaise (translation: "The parties confirm that this Agreement and all related documentation is and will be in the English language."); and (ii) You are responsible for complying with any local laws in your jurisdiction which might impact your right to import, export or use the Software, and You represent that You have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

**MISCELLANEOUS.** This Agreement constitutes the entire understanding of the parties with respect to the subject matter of this Agreement and merges all prior communications, representations, and agreements. This Agreement may be modified only by a written agreement signed by the parties. If any provision of this Agreement is held to be unenforceable for any reason, such provision shall be reformed only to the extent necessary to make it enforceable. This Agreement shall be construed under the laws of the State of Texas, USA, excluding rules regarding conflicts of law. The application of the United Nations Convention of Contracts for the International Sale of Goods is expressly excluded. This license is written in English and English is its controlling language.

**U.S. GOVERNMENT END USERS.** The Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer Software" and "commercial computer Software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire the Software with only those rights set forth herein.

...continued from back cover

produce superior image enhancements. Also new is a suite of editing functions to fix bloated star images, patch image flaws, and smooth bothersome sky gradients along with an image display control that helps you visualize all the information — and beauty — contained in your digital images. In short, *AIP4Win 2.0* is a full-featured image-processing program designed and optimized for loading, processing and displaying astronomical images. Here are a few of its capabilities:

- Load and process 16-bit, 32-bit, and 64-bit floating-point and integer images from all makes of astronomical CCD cameras.
- Load and process RAW, CRW, and NEF files from popular digital single-lens reflex cameras.
- Display, scroll, and magnify images from 16% to 1600%.
- Image file size is limited only by available of your computer's memory.
- Display pixel values, image statistics, histograms.
- Edit bloated stars, blooming trails, sky gradients, hot spots, and cosmic ray tracks.
- Full-featured image calibration functions include bias subtraction, automatic dark-frame matching and subtraction, flat-fielding, defect correction. Calibrate images individually, in groups, and during image stacking.
- Perform astronomical astrometry and photometry on images; determine celestial coordinates to sub-arc second accuracy and magnitudes with the millimag precision.

Plus much, much more...

### For optimum performance\* *AIP4Win 2.0* requires the following:

- Operating System: Windows® 98SE or later
- CPU: Intel® Pentium® IV, 2 GHz, or equivalent  
\*AIP4Win2.0 operates equally well but more slowly with slower CPUs.
- RAM: 256 Mb min., 512 Mb or better preferred
- Display: 1024 x 768 min., 1600 x 1200 or better preferred
- CDROM Reader: 16x or better (copy files to HDD)
- Hard Disk: 20 Mb free disk space for the program, plus storage for your image files  
(recommended minimum 20 Gb)

When you purchase the *Handbook of Astronomical Image Processing*, you receive one CDROM containing *AIP4Win 2.0* that is licensed for use on one personal and a portable computer if you own one. This software requires validation via the web (or telephone) within 30 days of installation. Once validated, you also receive the right to updates (free downloads via the web) and upgrades (for a fee) as they become available. You agree not to give or sell copies of *AIP4Win 2.0* to others because you understand that, with our low pricing and our generous use policy, your purchase allows us to continue expanding and updating your investment in *AIP4Win*, so it is clearly in your best interest to become a properly registered *AIP4Win* user. See page 682 for the Software License Agreement and Warranty Statement.

### To install *AIP4Win 2.0* on your computer...

To install *AIP4Win 2.0* on your computer, remove the CDROM from its pocket in the back of the book. Insert the CDROM in your CDROM reader and the installation package will run automatically. The first time you run *AIP4Win 2.0*, the program will ask for your name plus the serial number (S/N). You **MUST** register your copy of *AIP4Win 2.0* within 30 days. To register, go to the Willmann-Bell website (<http://www.willbell.com>) and browse to the *AIP4Win 2.0* webpage. After 30 days, *AIP4Win 2.0* will not run until you have registered it.

On the *AIP4Win 2.0* webpage, you will find information on available updates (free) and upgrades (for a fee) as they become available. You **MUST** register for access to the page.

### Bug Reports and other problems...

You can communicate bug reports and other problems directly to Richard Berry, James Burnell and Willmann-Bell, Inc. by e-mailing at [AIP4Win@willbell.com](mailto:AIP4Win@willbell.com). Please provide as much information as you can about the events that lead up to the problem.

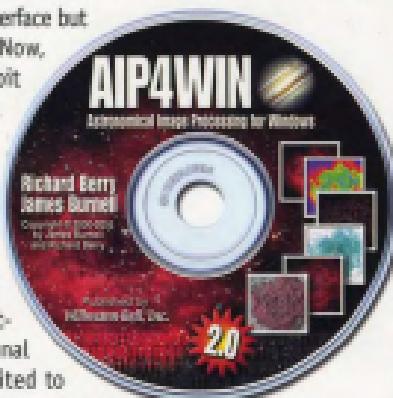
## About this book and software...

This second edition of the *Handbook of Astronomical Image Processing* (HAIP) and its integral *AIP for Windows 2.0* image processing software (*AIP4Win 2.0*) addresses many important changes that have taken place in astronomical imaging since the publication of the first edition. Today's affordable astro-imaging capable digital single-lens-reflex (DSLR) cameras, the growing power of personal computers, and the proliferation of telescopes and imaging accessories has brought imaging capabilities within the reach of practically every amateur astronomer — and this second edition of the *Handbook* plus *AIP4Win 2.0* is ready, willing and able to assist every observer in making great astronomical images.

In the *Handbook*, we amplified the original chapters on astronomical equipment and imaging techniques, revised our discussions of astrometry and photometry to reflect the steady growth in these scientific fields, and expanded tutorials in the back of the book to help you get up to speed quickly. On the accompanying CDROM (found on the inside back cover) you will find hundreds of megabytes of sample images you can use to learn techniques such as image registration and stacking that guarantee good results even from those living with suburban and urban skies. Also new are comprehensive chapters on color imaging with astronomical CCD cameras, processing color images from digital cameras and photon-counting fundamentals every serious astro-imager needs to know.

*AIP4Win 2.0* retains its highly acclaimed interface but with significantly increased capabilities. Now, *AIP4Win 2.0* processes all image data in 32-bit floating-point format to insure that you will not lose even one photon of precious light. Abop these powerful 32-bit floating-point core routines, we built an image display engine capable of showing you images in both color and black-and-white, from a minimum of 16% to a maximum of 1600% size. What you can load into *AIP4Win 2.0* and display is now limited only by the memory on board your personal computer — and this capability is not limited to black-and-white images — *AIP4Win 2.0* now has a suite of sophisticated software tools for loading and processing astronomical color images plus a new signal-to-noise ratio calculator to assist you in optimizing your imaging strategy.

*AIP4Win 2.0* now has the ability to perform wavelet image enhancement, wavelet image analysis and wavelet noise reduction — techniques so new they're not even mentioned in the classic works about image processing. The deconvolution procedures have been enhanced so not only do they run faster than ever before, but they also



Published by



P.O. Box 38025  
Richmond, VA 23235  
Tel. Free 1 (800) 825-7827  
(804) 320-7098 • Fax (804) 272-5980  
[www.willbell.com](http://www.willbell.com)

(continued on inside back cover flyleaf)

ISBN 0-943398-82-4



010101100011010010011100100111  
1100100101011010110011010110101101