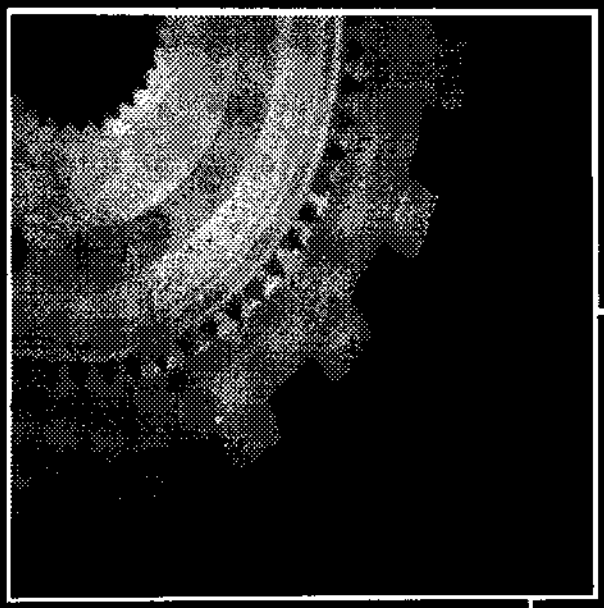
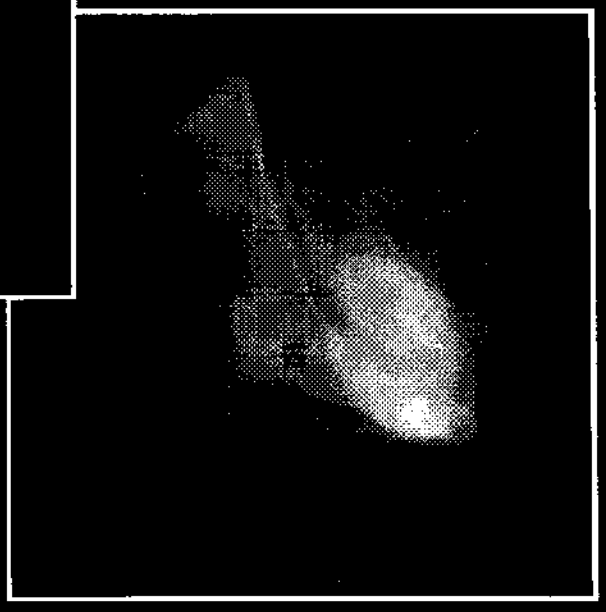
COMPUTER VISION:



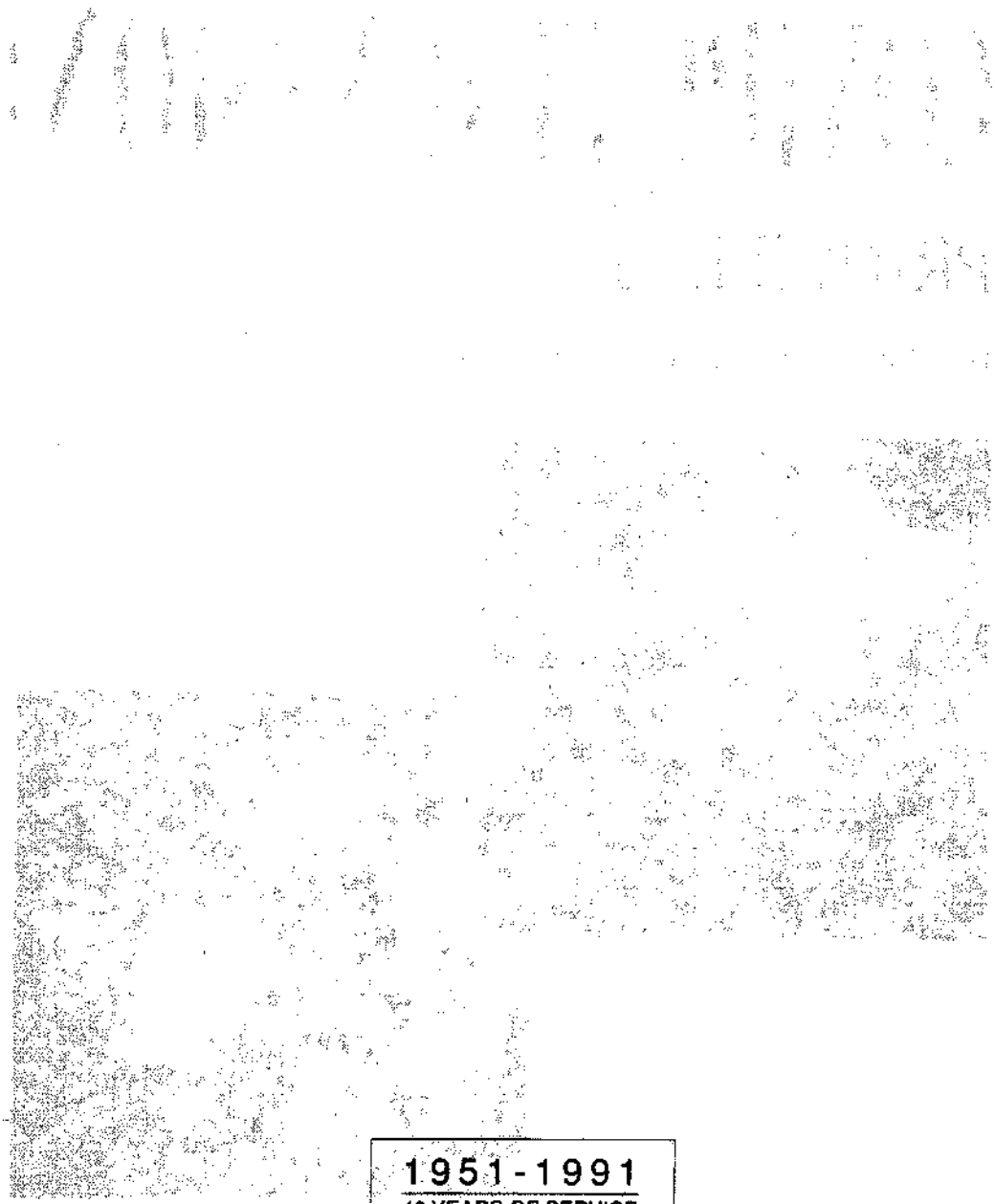


1951 1991

IEEE COMPUTER SOCIETY PRESS

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

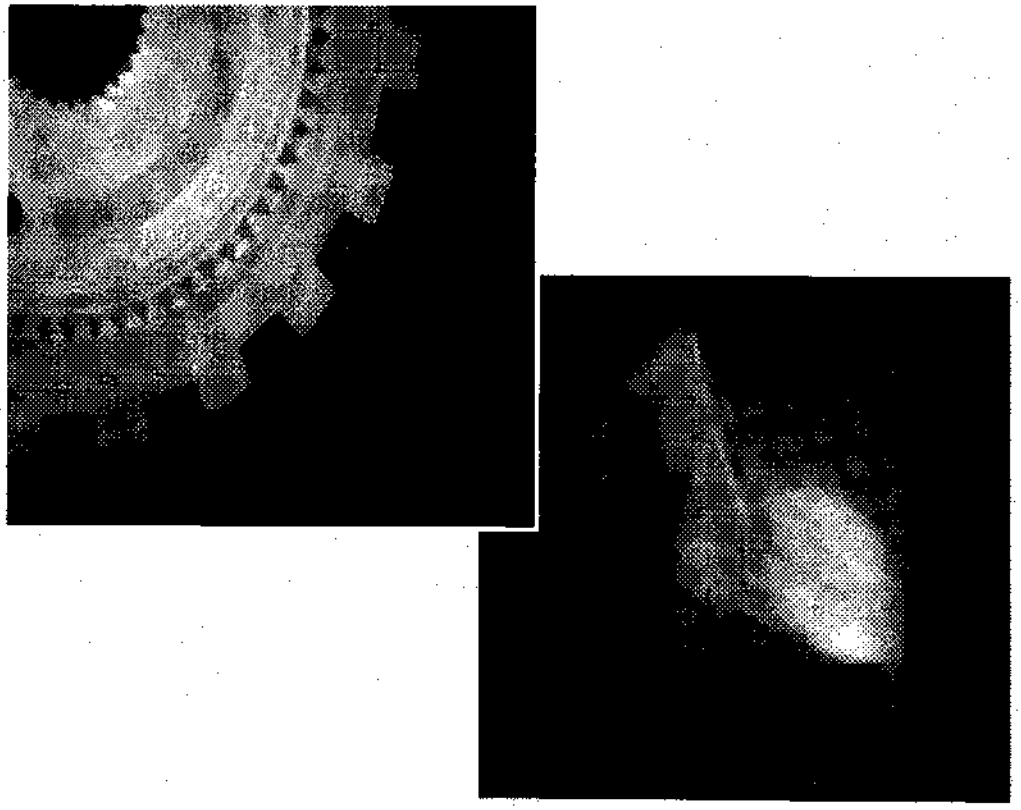
Computer Vision: Principles



COMPUTER VISION:

PRINCIPLES

BY Rangachar Kasturi and Ramesh C. Jain



1951–1991
IEEE COMPUTER SOCIETY PRESS
THE INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS, INC.

Computer Vision: Principles

Rangachar Kasturi
and
Ramesh C. Jain

1951-1991
40 YEARS OF SERVICE

IEEE COMPUTER SOCIETY
A member society of the
Institute of Electrical and Electronics Engineers, Inc.

IEEE Computer Society Press
Los Alamitos, California

Washington • Brussels • Tokyo

IEEE Computer Society Press Tutorial

Library of Congress Cataloging-in-Publication Data

Computer Vision / [edited by] Rangachar Kasturi and Ramesh C. Jain.
p. cm. — (IEEE Computer Society Press tutorial)
Includes bibliographical references.
Contents: v. 1. Principles — v.2. Advances and Applications.
ISBN 0-8186-9102-6
1. Computer vision.
I. Kasturi, Rangachar, 1949-. II. Jain, Ramesh. III. Series.
TA1632.C6582 1991
621.39'9 — dc20 91-13117
CIP

1951-1991 40 YEARS OF SERVICE IEEE COMPUTER SOCIETY PRESS

Published by the
IEEE Computer Society Press
10662 Los Vaqueros Circle
PO Box 3014
Los Alamitos, CA 90720-1264

IEEE Computer Society Press Order Number 2102
Library of Congress Number 91-13117
IEEE Catalog Number 91EH0339-2
ISBN 0-8186-6102-X (microfiche)
ISBN 0-8186-9102-6 (case)

Additional copies can be ordered from

IEEE Computer Society Press
Customer Service Center
10662 Los Vaqueros Circle
PO Box 3014
Los Alamitos, CA 90720-1264

IEEE Service Center
445 Hoes Lane
PO Box 1331
Piscataway, NJ 08855-1331

IEEE Computer Society
13, avenue de l'Aquilon
B-1200 Brussels
BELGIUM

IEEE Computer Society
Ooshima Building
2-19-1 Minami-Aoyama
Minato-ku, Tokyo 107
JAPAN

Technical Editor: Fred Petry
Production Editor: Lisa O'Conner
Copy Editor: Phyllis Walker
Cover image: Karen Brady
Cover design : Alex Torres
Printed in the United States of America by Braun-Brumfield, Inc.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

Preface

Designing intelligent computer-based systems with scene interpretation capabilities comparable to those of humans has attracted the attention of researchers for more than two decades. Initially, the memory and processing limitations of computers constrained computer vision research. But recent advances in computer hardware have opened the door for the design of systems that analyze and interpret complex three-dimensional scenes.

Like other evolving technologies, computer vision is based on certain fundamental principles and techniques. For example, models of image formation and techniques for pixel-level image processing are reasonably well developed. Several commercial machine vision systems, notably in the area of industrial inspection, have been successfully developed and installed. However, designing a general-purpose natural-scene interpretation system that operates in an unconstrained domain remains an elusive and challenging task.

To design successful practical computer vision systems requires a thorough understanding of all system aspects, from initial image formation to final scene interpretation. To convert raw data at the sensors to meaningful information about objects in the scene being interpreted, typical vision systems use the following processing-step sequence:

- Image capture and enhancement;
- Segmentation;
- Feature extraction;
- Matching of features to models;
- Exploitation of constraints and image cues to recover information lost during the imaging process; and
- Application of domain knowledge to recognize objects in the scene and their attributes.

In each of these steps, many factors influence the choice of algorithms and techniques. The system designer should be knowledgeable about the issues and design trade-offs inherent in the realization of a practical system. For example, careful choice of factors influencing image formation — such as imaging modality — could greatly simplify subsequent image analysis problems.

This book, together with its companion book *Computer Vision: Advances and Applications* (IEEE Computer Society Press), is (1) a tutorial, (2) a guide to practical applications, and (3) a reference source on recent advances in computer vision research. The tutorial component will benefit students and professionals who are relatively new to the computer vision field. The description of practical applications of machine vision technology will act as a guide to practicing engineers. And the collection of papers on recent research advances will be an excellent reference source for active researchers in the computer vision field.

The seven chapters in this book introduce the following fundamental topics in computer vision:

- Image formation;
- Segmentation;
- Feature extraction and matching;
- Constraint exploitation and shape recovery
- Three-dimensional object recognition;
- Dynamic vision; and
- Knowledge-based vision.

The order of topic presentation is generally the same as that followed in vision systems that generate scene interpretation from image data, with some overlapping between chapters. Each chapter begins with an introductory tutorial, followed by a collection of key papers covering the topics presented in the tutorial. We have chosen papers that are tutorial in nature or that describe a fundamental principle, concept, or commonly used algorithm. The Epilogue presents current research trends and future directions for computer vision. The Bibliography, at the end of the book, lists selected papers. The large

volume of published literature in the computer vision field precludes the Bibliography from being comprehensive; we have generally limited the selection to papers published since 1980. It is intended to function as a pointer to related literature.

The organization of the companion book, *Computer Vision: Advances and Applications*, is the same as that followed in this book, with chapter titles being the same in the two. The papers included therein describe recent research advances in the topics covered in each of the chapters in this book. Also included in the companion book is a representative set of papers describing the following five machine vision application areas:

- Aerial image analysis;
- Document image analysis;
- Medical image analysis;
- Industrial inspection and robotics; and
- Autonomous navigation.

We believe that the ideas and techniques that are described in the two companion books, *Computer Vision: Principles* and *Computer Vision: Advances and Applications*, will continue to influence vision system research and design for many years to come.

The contributions of a large number of researchers have enriched the computer vision field. Unfortunately, due to space constraints, we could not include in these two companion books all papers containing significant contributions. The 77 papers selected for inclusion in these two books include frequently cited reference papers and recent papers that we believe represent significant contributions to the literature.

To identify papers that are frequently cited in current literature, we compiled a list of papers cited in about 275 relevant papers published in the *Proceedings of the IEEE International Conference on Computer Vision, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, and IEEE Transactions on Pattern Analysis and Machine Intelligence*. The resulting list comprised 1,740 papers, excluding references such as books and reports. Out of this list of 1,740 papers, 280 were cited more than once and 40 of these were cited at least five times. We selected only papers that are relevant to the topics covered in these two books. (Papers were replaced with their most recent versions, whenever available.) To this list of relevant, frequently cited papers, we added papers from recent journals and conferences to obtain the final list of 77 papers. Most of the papers selected were published within the past five years. Earlier papers were included only if they are key papers describing a particular topic.

We appreciate the assistance provided by Thawach Sripradisvarakul, Yuan-liang Tang, Jayant Kattepur, Chan-pyng Lai, Erliang Yeh, and Chia-hong Chen in compiling the Bibliography. We would like to thank Professor Mohan Trivedi and anonymous reviewers for their helpful comments and suggestions; Professors Jon Butler and Fred Petry for their help and encouragement; Kathy Dewitt, Elaine Smiles, and Dolores Bolsenga for secretarial assistance; Karen Brady for cover art; Alex Torres for cover design, Phyllis Walker for copyediting, IEEE Computer Society Press Editorial Director Henry Ayling for his comments, and Lisa O'Conner and Catherine Harris of the IEEE Computer Society Press for their help in assembling all of the material in a short time to ensure timely publication of these books. We would also like to thank the many authors who provided reprints of their articles for reproduction and gave us permission to use their work. We hope that the readers find that these books meet their expectations and welcome any comments, corrections, and suggestions.

September 27, 1991: Rangachar Kasturi and Ramesh C. Jain

Table of Contents

Preface v

Chapter 1: Image Formation 1

Understanding Image Intensities 10
B.K.P. Horn *(Artificial Intelligence*, Vol. 17, 1977, pp. 201-231)
Active, Optical Range Imaging Sensors 36
P.J. Besl *(Machine Vision and Applications*, 1988, pp. 127-152)

Chapter 2: Segmentation 65

Theory of Edge Detection 77
D. Marr and E. Hildreth *(Proc. Royal Society of London*, Series B, 1980, pp. 187-217)
Scale-Space Filtering 108
A.P. Witkin *(Proc. 8th Int'l Joint Conf. Artificial Intelligence*, August 1983, pp. 1019-1022)
A Computational Approach to Edge Detection 112
J. Canny *(IEEE Trans. Pattern Analysis and Machine Intelligence*, November 1986, pp. 679-698)
Discontinuity Detection for Visual Surface Reconstruction 132
W.E.L. Grimson and T. Pavlidis *(Computer Vision, Graphics, and Image Processing*, June 1985, pp. 316-330)
Segmentation Through Variable-Order Surface Fitting 144
P.J. Besl and R.C. Jain *(IEEE Trans. Pattern Analysis and Machine Intelligence*, March 1988, pp. 167-192)

Chapter 3: Feature Extraction and Matching 171

Generalizing the Hough Transform to Detect Arbitrary Shapes 183
D.H. Ballard *(Pattern Recognition*, 1981, pp. 111-122)
The Topographic Primal Sketch 195
R.M. Haralick, L.T. Watson, and T.J. Laffey, *(Int'l J. Robotics Research*, Spring 1983, pp. 50-72)
Image Analysis Using Mathematical Morphology 218
R.M. Haralick, S.R. Sternberg, and X. Zhuang *(IEEE Trans. Pattern Analysis and Machine Intelligence*, July 1987, pp. 532-550)
A Structural Model of Shape 237
L.G. Shapiro *(IEEE Trans. Pattern Analysis and Machine Intelligence*, March 1980, pp. 111-126)
Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography 253
M.A. Fischler and R.C. Bolles *(Communications of the ACM*, June 1981, pp. 381-395)

Chapter 4: Constraint Exploitation and Shape Recovery 269

Cooperating Processes for Low-Level Vision: A Survey 282
L.S. Davis and A. Rosenfeld *(Artificial Intelligence,* 1981, pp. 245-263)

Photometric Method for Determining Surface Orientation From Multiple Images 301
R.J. Woodham *(Optical Engineering,* January/February 1980, pp. 139-144)

Region-Based Stereo Analysis for Robotic Applications 307
S.B. Marapane and M.M. Trivedi *(IEEE Trans. Systems, Man, and Cybernetics,* November/December 1989, pp. 1447-1464)

Shape from Texture: Integrating Texture-Element Extraction and Surface Estimation 325
D. Blostein and N. Ahuja *(IEEE Trans. Pattern Analysis and Machine Intelligence,* December 1989, pp. 1233-1251)

Chapter 5: Three-Dimensional Object RecognitionAbout the authors 345

Volumetric Description of Objects From Multiple Views 358
W.N. Martin and J.K. Aggarwal *(IEEE Trans. Pattern Analysis and Machine Intelligence,* March 1983, pp. 150-158)

Model-Based Three-Dimensional Interpretations of Two-Dimensional Images 367
R.A. Brooks *(IEEE Trans. Pattern Analysis and Machine Intelligence,* March 1983, pp. 140-150)

Characteristic Views as a Basis for 3-Dimensional Object Recognition 378
I. Chakravarty and H. Freeman *(Proc. SPIE Conference on Robot Vision,* Vol. 336, 1982, pp. 37-45)

Fleshing Out Projections 387
M.A. Wesley and S. Markowsky *(IBM J. Research and Development,* November 1981, pp. 934-954)

Recognizing 3-D Objects Using Surface Descriptions 408
T.-J. Fan, G. Medioni, and R. Nevatia *(IEEE Trans. Pattern Analysis and Machine Intelligence,* November 1989, pp. 1140-1157)

Model-Based Recognition in Robot Vision 426
R.T. Chin and C.R. Dyer *(ACM Computing Surveys,* March 1986, pp. 67-108)

Chapter 6: Dynamic Vision 469

Determining Optical Flow 481
B.K.P. Horn and B.G. Schunck *(Artificial Intelligence,* Vol. 17, 1981, pp. 185-203)

An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields From Image Sequences 498
H.-H. Nagel and W. Enkelmann *(IEEE Trans. Pattern Analysis and Machine Intelligence,* September 1986, pp. 565-593)

Disparity Analysis of Images 527
S.T. Barnard and W.B. Thompson *(IEEE Trans. Pattern Analysis and Machine Intelligence,* July 1980, pp. 333-340)

Finding Trajectories of Feature Points in a Monocular Image Sequence 535
I.K. Sethi and R.C. Jain *(IEEE Trans. Pattern Analysis and Machine Intelligence,* January 1987, pp. 56-73)
Generalizing Epipolar-Plane Image Analysis on the Spatiotemporal Surface ..553
H.H. Baker and R.C. Bolles *(Int'l J. Computer Vision,* May 1989, pp. 33-49)
Analysis of a Sequence of Stereo Scenes Containing Multiple Moving Objects Using Rigidity Constraints 570
Z. Zhang, O.D. Faugeras, and N. Ayache *(Proc. 2nd Int'l Conf. Computer Vision,* 1988, pp. 177-186)

Chapter 7: Knowledge-Based Vision 581

Survey of Model-Based Image Analysis Systems 585
T.O. Binford *(Int'l J. Robotics Research,* Spring 1982, pp. 18-64)
Low-Level Image Segmentation: An Expert System 632
A.M. Nazif and M.D. Levine *(IEEE Trans. Pattern Analysis and Machine Intelligence,* September 1984, pp. 555-577)

Chapter 8: Applications 655

Epilogue 657

Bibliography 659

Books, Special Issues, and Conferences 702

Index 707

About the authors 712

Chapter 1: Image Formation

An image is formed when a sensor records received radiation as a two-dimensional function. The brightness or intensity values in an image may represent different physical entities. For example, in a typical image obtained using a video camera, the intensity values represent the reflectance of light from various object surfaces in the scene; in a thermal image, they represent the temperature of corresponding regions in the scene; and in a range image, they represent the distance from the camera to various points in the scene. Multiple images of the same scene are often captured using different types of sensors to facilitate more robust and reliable interpretation of scene than the interpretation based on a single image. Selecting an appropriate image formation system plays a key role in the design of practical computer vision systems. Principles of image formation are described in this chapter.

Intensity Images

Intensity images of scenes formed using visible light are widely used in computer vision systems. The primary challenge in computer vision is the analysis of two-dimensional images of scenes to generate a three-dimensional interpretation. Construction of a model of image formation, which encapsulates knowledge of imaging geometry, the projection process, and the reflectance properties of objects, is essential for image analysis and interpretation.

Imaging geometry. A simple camera-centered imaging model is shown in Figure 1.1.[1] The coordinate system is chosen such that the X,Y-plane coincides with the image plane and the Z-axis passes through the lens center, which is at a distance of f, the focal length from the image plane. The image of a scene point (X,Y,Z) forms at a point (x,y) on the image plane where

$$x = \frac{fX}{(f - Z)}, \qquad y = \frac{fY}{(f - Z)}. \tag{1.1}$$

These are the perspective projection equations for an imaging system. (Note that an abstract frontal image plane that is located at a distance of f in front of the lens center is often used to model imaging. In such a model, the signs of coordinates of scene points are preserved in their corresponding image point coordinates.) When f is very large, the perspective projection equations can be approximated by the orthographic projection equations $x = X$ and $y = Y$. In a typical imaging situation, the camera may have several degrees of freedom, such as translation, pan, and tilt. Also, more than one camera may be imaging the same scene from different points, in which case it is convenient to adopt a world coordinate system in reference to which the scene coordinates and camera coordinates are defined. For example, the camera shown in Figure 1.2[1] is translated by (X_0,Y_0,Z_0), panned by an angle θ (the angle between the x- and X-axes), and tilted by an angle α (the angle between the z- and Z-axes). In addition, there is a displacement of the image plane with respect to the gimbal center by vector $r = (r_1,r_2,r_3)$. The image coordinates (x,y) of a point (X,Y,Z) in the world coordinate system are then obtained by applying appropriate transformations, yielding[1]

$$x = f\,\frac{(X - X_0)\cos\theta + (Y - Y_0)\sin\theta - r_1}{-(X - X_0)\sin\theta\sin\alpha + (Y - Y_0)\cos\theta\sin\alpha - (Z - Z_0)\cos\alpha + r_3 + f} \tag{1.2}$$

and

$$y = f\,\frac{-(X - X_0)\sin\theta\cos\alpha + (Y - Y_0)\cos\theta\cos\alpha + (Z - Z_0)\sin\alpha - r_2}{-(X - X_0)\sin\theta\sin\alpha + (Y - Y_0)\cos\theta\sin\alpha - (Z - Z_0)\cos\alpha + r_3 + f}. \tag{1.3}$$

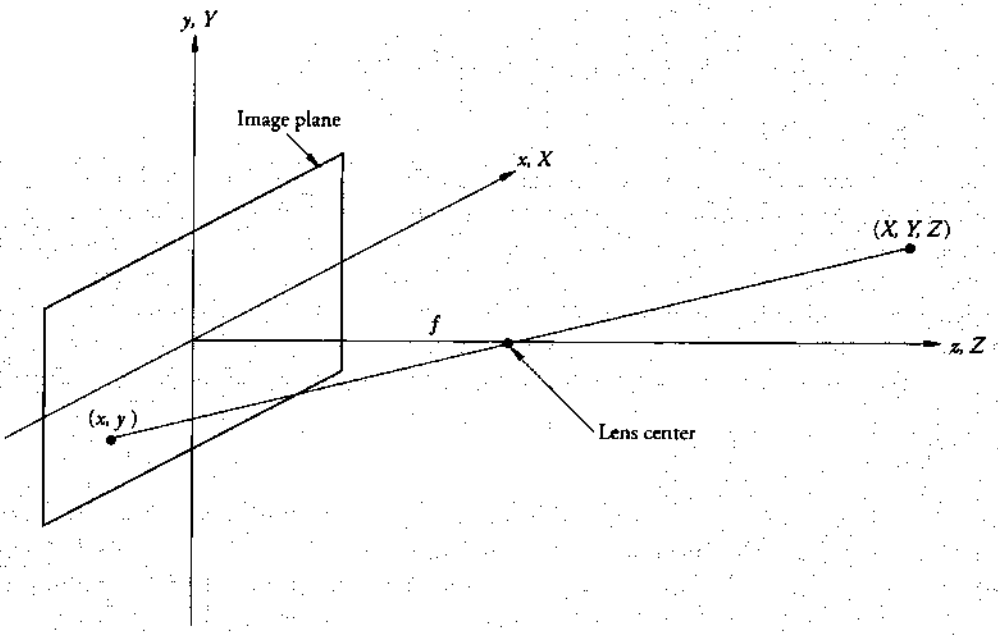


Figure 1.1: A camera-centered imaging model (from R.C. Gonzalez and P. Wintz, *Digital Image Processing*, © 1977 by Addison-Wesley Publishing Company. Reprinted with permission of the publisher).[1]

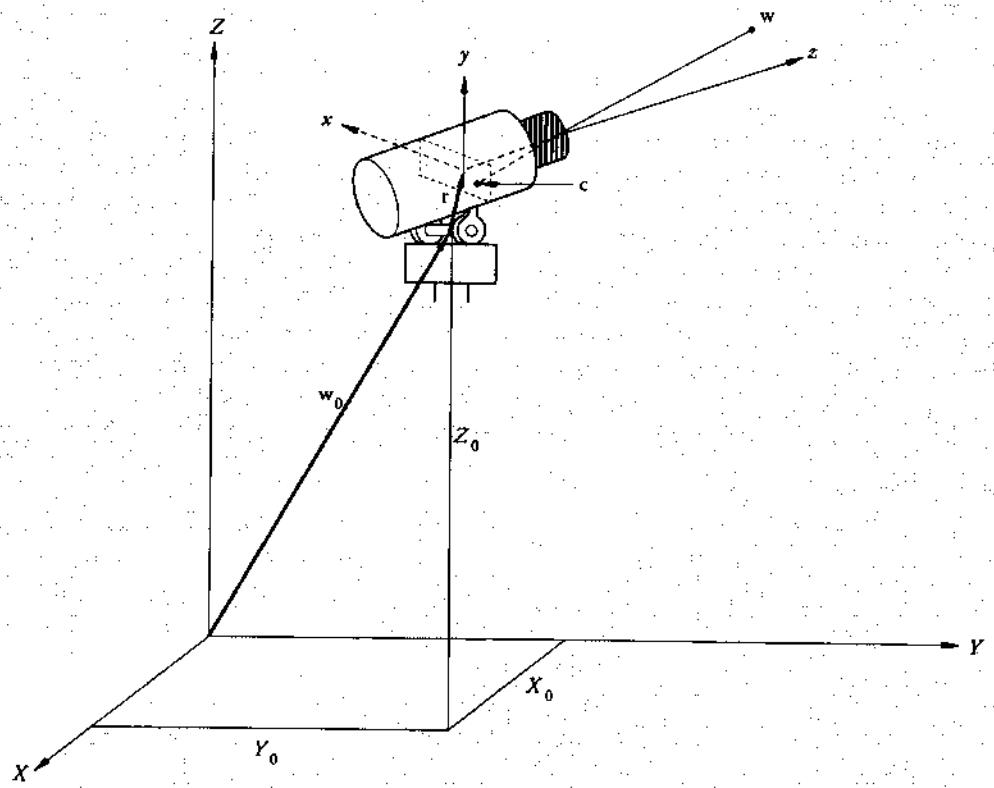


Figure 1.2: Imaging geometry to determine the relationship between scene points defined in world coordinates and image coordinates (from R.C. Gonzalez and P. Wintz, *Digital Image Processing*, © 1977 by Addison-Wesley Publishing Company. Reprinted with permission of the publisher).[1]

Often, the exact location, orientation, and focal length of the camera are not known. Then, the parameters of transformations must be computed, using known scene and image coordinates of a set of points. For example, the perspective transformation relationship between the world coordinates of a point (X,Y,Z) and its image coordinates (x,y) for a camera system with an arbitrary location and orientation is given by[1]

$$\begin{aligned} a_{11}X + a_{12}Y + a_{13}Z - a_{41}xX - a_{42}xY - a_{43}xZ - a_{44}x + a_{14} = 0, \\ a_{21}X + a_{22}Y + a_{23}Z - a_{41}yX - a_{42}yY - a_{43}yZ - a_{44}y + a_{24} = 0, \end{aligned} \tag{1.4}$$

The unknown coefficients in these two equations can be computed — using a process called "camera calibration" — when the scene and image coordinates of a set of at least six points are known. The image coordinates of any other scene point can then be easily computed using these coefficients.

Image intensity. While the imaging geometry uniquely determines the relationship between scene coordinates and image coordinates, the brightness or intensity at each point is determined not only by the imaging geometry but also by several other factors, including scene illumination, reflectance properties and surface orientations of objects in the scene, and radiometric properties of the imaging sensor.

The reflectance properties of a surface are characterized by the Bidirectional Reflectance Distribution Function (BRDF)[2]. BRDF is the ratio of the radiance in the direction of the observer to the irradiance due to a source from a given direction. It captures how bright a surface will appear when viewed from a given direction and illuminated by another direction. For example, for a Lambertian surface, BRDF is a constant; hence, the surface appears equally bright from all directions. For a specular (mirrorlike) surface, BRDF is an impulse function, as determined by the laws of reflection. The scene illumination, together with BRDF, determine the scene radiance (flux density emitted into a unit solid angle in a given direction) at a point. The relationship between the image irradiance E (flux density incident on the image plane at a given point), the scene radiance L, the diameter d and focal length f of the imaging lens, and the angle α between the camera axis and the line connecting the scene point to the lens center (see Figure 1.3[3]) is given by[2,3]

$$E = L\,\frac{\pi}{4}\left(\frac{d}{f}\right)^2 \cos^4\alpha. \tag{1.5}$$

Note that although the image irradiance is directly proportional to the scene radiance, the sensitivity of the system is not constant over the entire image plane; rather, it diminishes as the fourth power of the cosine of the off-axis angle α. However, this diminishment can be ignored when the angular field of view of the imaging system is small. Image intensity is related to the image irradiance by the relative luminous efficiency function of the sensor.[4] Sensors with spectral responses at different bands of the electromagnetic spectrum are often used to obtain multispectral image data.

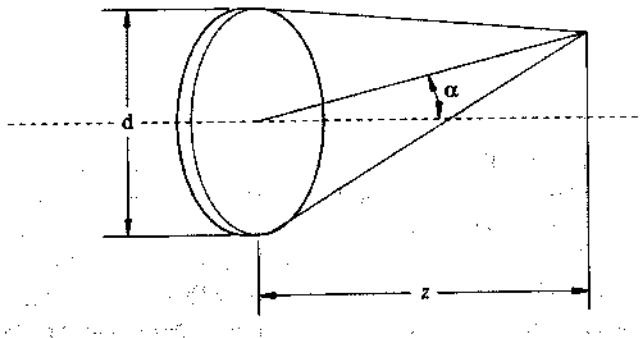


Figure 1.3: Geometry to find the relationship between image irradiance and scene radiance (from B.K.P. Horn, *Robot Vision*, © 1986 McGraw-Hill, Inc., reprinted with permission.).[3]

Reflectance map. The scene radiance at a point on the surface of an object depends on the reflectance properties of the surface, as well as on the intensity and direction of the illuminating sources. For example, for a Lambertian surface illuminated by a point source, the scene radiance is proportional to the cosine of the angle between the surface normal and the direction of illumination.

The relationship between surface orientation and brightness is captured in the reflectance map. In the reflectance map for a given surface and illumination, contours of constant brightness [$R(p,q) = constant$] are plotted as a function of surface orientation specified by the gradient space coordinates (p,q). A typical reflectance map for a Lambertian surface illuminated by a point source of light is shown in Figure 1.4.[3] In this figure, the brightest point [$R(p,q) = maximum$] corresponds to the surface orientation such that its normal points in the direction of the source. Since image brightness is proportional to scene radiance, a direct relationship exists between the image intensity at a point and the orientation of the surface at the corresponding scene point. Shape-from-shading algorithms exploit this relationship between image intensity and surface orientation to recover three-dimensional object shape. An alternative method, photometric stereo, exploits the same principles to recover object shape from multiple images obtained by illuminating the object from different directions.[3] These methods are discussed in detail in Chapter 4.

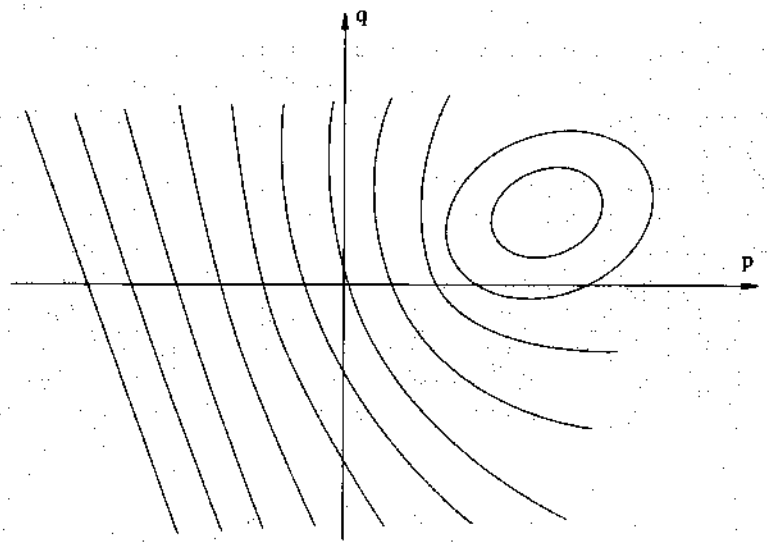


Figure 1.4: Typical reflectance map for a Lambertian surface illuminated by a point source (from B.K.P. Horn, *Robot Vision*, © 1986 McGraw-Hill, Inc., reprinted with permission.).[3]

Color vision

Perception of color by humans is attributed to differences in the spectral responses of photoreceptors — rods and cones — in the retina. Rods are responsible for monochromatic vision. The human visual system consists of three types of cones, each with a different spectral response. Assuming a simple multiplicative model, the total response of each of the cones is given by the integral

$$\int f(\lambda) r(\lambda) h_i(\lambda) d\lambda, \quad i = 1,2,3 \tag{1.6}$$

where $f(\lambda)$is the spectral composition of the illumination, $r(\lambda)$ is the spectral reflectance function of the reflecting surface, and $h_i(\lambda)$, where $i = 1,2,3$ is the spectral response of each of the three types of cones.[5] Note that two surfaces with reflectance functions $r_1(\lambda)$ and $r_2(\lambda)$ illuminated by $f_1(\lambda)$ and $f_2(\lambda)$ are perceived as having the same color if the total response for each sensor, as given by the above integral, is equal to that of the other. In fact, any given color can be matched by using a weighted combination of a suitably chosen set of three primary colors.[4] Knowledge of this fact leads to the familiar notion of representing color in images by three components: red, green, and blue. Note that the choice of the three primary colors determines the range of possible colors that can be realized as a weighted combination of the primary colors. The CIE (Commission Internationale de l'Eclairage) standard primary system uses the following spectral primaries: red [700 nanometers (nm)], green (546.1 nm), and blue (435.8 nm). All colors that are within the triangle drawn on a chromaticity diagram with vertices at these three points can be matched by a weighted combination of these colors. Matching colors that are outside this triangle would require choosing a different set of primary colors.

Although color plays an important role in scene interpretation by humans, the bulk of computer vision research has been limited to processing monochromatic images. Exceptions are applications — such as remote sensing — in which multispectral data are

used for classification. A threefold increase in processing and memory requirements, compared to those of monochromatic images, has been the principal inhibiting factor in color image processing. But with recent advances in computer hardware, activity in exploiting the additional dimensionality provided by color images has increased. An important first step toward designing color vision systems is obtaining an accurate model for color image formation.

Light reflection model for color images. Color in images is determined primarily by

- Light source chromaticity;
- Spectral reflection properties of object surfaces; and
- Sensor spectral responsivity of the imaging system.

Light reflected from surfaces is modeled as consisting of

- An interface reflection component and
- A body reflection component.[6]

The spectral composition of the interface reflection component is assumed to be the same as that of the illuminating source, whereas the spectral composition of the body reflection component is determined by the pigments in the object. Using this assumption, the Bidirectional Spectral Reflectance Distribution Function (BSRDF) — which includes both spectral and directional dependence of the reflectivity function — can be written as

$$f_r(\theta_i, \phi_i; \theta_r, \phi_r; \lambda) = a(\lambda)g(\theta_i, \phi_i; \theta_r, \phi_r) + sh(\theta_i, \phi_i; \theta_r, \phi_r), \tag{1.7}$$

where (θ_i, ϕ_i) and (θ_r, ϕ_r) are the direction of the source and the direction of the observer, respectively. The first term represents the body reflection component (which is wavelength selective) and the second term represents the interface reflection component.[7] Lee, et al., have shown that this neutral-interface-reflection model is appropriate for polychromatic collimated light sources that illuminate surfaces whose BSRDFs have separable spectral and geometric factors. Also, they have presented experimental results evaluating the accuracy of this model for surfaces made up of different materials.[7]

Color constancy. The human perceptual system has the remarkable ability to assign stable colors to object surfaces despite changes in the spectral distribution of illuminating sources. This well-known ability is called "color constancy." However, the spectral distribution of received radiation at the sensor is a function of both the spectral reflectance properties of the object surfaces in the scene and the spectral composition of the illuminating sources. Maloney and Wandell[8] describe an algorithm for estimating the spectral reflectance functions of surfaces even when information about the spectral distribution of ambient light is incomplete. Funt and Ho[9] describe a technique that exploits the chromatic aberration in an optical-imaging system to separate the reflectance component from the illumination component. This method, in principle, solves the color constancy problem.

Range Images

Extraction of the three-dimensional structure of objects from images is an important task for a computer vision system. Although three-dimensional information can be extracted from images with two-dimensional intensity — using image cues such as shading, texture, and motion — the problem is greatly simplified by range imaging. Range imaging is acquiring images for which the value at each pixel is a function of the distance of the corresponding point of the object from the sensor. Besl[10] describes in detail various methods of range imaging and compares their relative merits. An earlier survey by Jarvis[11] includes not only direct range-measuring techniques, but also techniques in which the range is calculated from two-dimensional image cues. Range-image sensing, processing, interpretation, and applications are described in detail in Jain and Jain.[12] Two of the most commonly used principles for range imaging — imaging radar and triangulation — are briefly described in the following sections.

Imaging radar. In a time-of-flight pulsed radar, the distance to the object is computed by observing the time difference between the transmitted and received electromagnetic pulses. Range information can also be obtained by detecting the phase difference between the transmitted and received waves of an amplitude-modulated beam or by detecting the beat frequency in a coherently mixed transmitted-and-received signal in a frequency-modulated beam. Several commercial laser beam imaging systems have been built using these principles.

Triangulation. In an active triangulation-based range-imaging system, a light projector and a camera aligned along the z-axis are separated by a baseline distance b, as shown in Figure 1.5.[10] The object coordinates (X,Y,Z) are related to the measured image coordinates (x,y) and the projection angle θ by

$$[X\ Y\ Z] = \frac{b}{f\ \cot\theta - x}\ [x\ y\ f]. \tag{1.8}$$

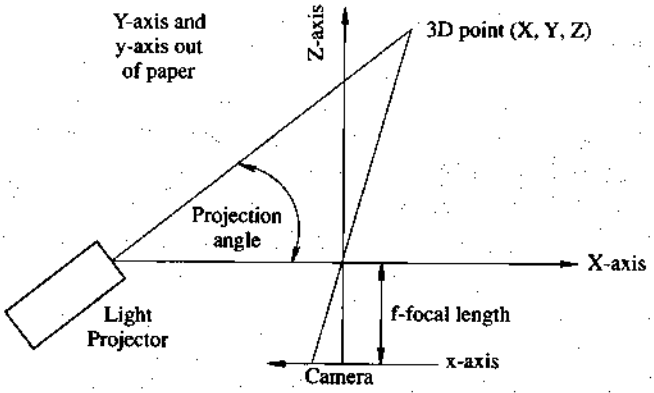


Figure 1.5: Camera-centered triangulation geometry (from Besl).[10]

The accuracy with which the angle θ and the horizontal position x can be measured determines the range resolution of such a triangulation system. This system of active triangulation using a point source of light is an example of structured lighting, a technique that is described in more detail in the following section.

Structured lighting

Imaging using structured lighting refers to systems in which the scene is illuminated by a known geometric pattern of light. In a simple point projection system — like the triangulation system discussed above — the scene is illuminated, one point at a time, at each point in a two-dimensional grid pattern. The depth at each point is then calculated using equation 1.8 above to obtain a two-dimensional-range image. Because of the sequential nature of this system, it is slow and not suitable for use with dynamically changing scenes.

In a typical structured-lighting system, either planes of light or two-dimensional patterns of light are projected on the scene. A camera (displaced spatially from the source of illumination) observes the patterns of light projected on object surfaces. The observed images of the light patterns contain distortions that are determined by the shape and orientation of surfaces of objects on which the patterns are projected, as illustrated in Figure 1.6.[11] Note that the light pattern as seen by the camera contains discontinuities and changes in orientation and curvature. The three-dimensional object coordinate corresponding to any point in the image plane can be calculated by computing the intersection of the line-of-sight with the light plane. To obtain the complete object description, parallel planes of light are projected in sequence, and the corresponding stripe images are acquired. Different surfaces in the object are detected by clustering stripes of light having similar spatial attributes. In dynamically changing situations in which projecting light stripes in sequence is not practical, a set of multiple stripes of light — in which each stripe is uniquely encoded — is projected. For example, using a binary-encoding scheme, a complete set of data can be acquired by projecting only $\log_2 N$ patterns, where $(N\text{-}1)$ is the total number of stripes. Boyer and Kak[13] describe a method in which color coding is used to acquire the range information using a single image. The classical paper by Will and Pennington[14] describes using grid coding and Fourier domain processing techniques to locate different planar surfaces in a scene.

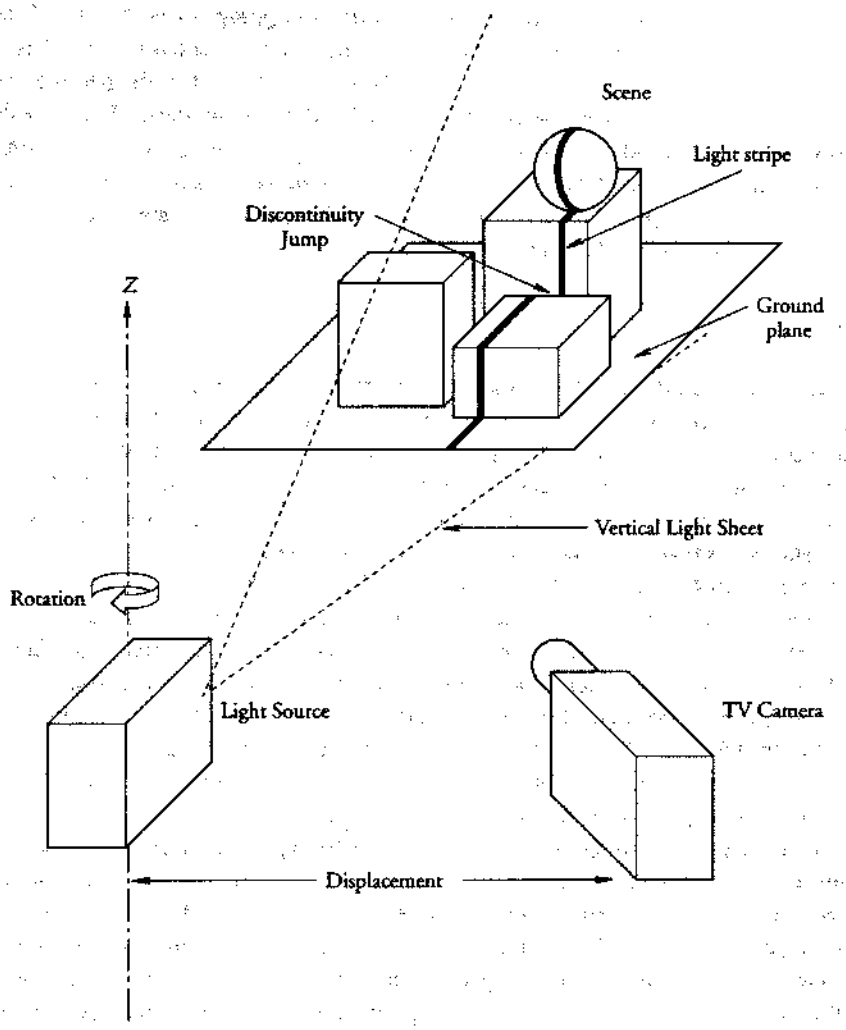


Figure 1.6: Illustration of striped lighting technique (from Jarvis).[11]

Structured-lighting techniques have been used extensively in industrial vision applications in which the illumination of the scene can be easily controlled. In a typical application, objects on a conveyor belt pass through a plane of light, creating a distortion in the image of the light stripe. The profile of the object at the plane of the light beam is then calculated. This process is repeated at regular intervals to recover the shape of the object. The primary drawback of structured-lighting systems is that data cannot be obtained for object points that are not visible to either the light source or the imaging camera.

Structured lighting is useful in binocular stereopsis of surfaces that contain few dominant features, thereby making the finding of corresponding points in a stereo pair of images difficult. In such a structured-lighting system, the stereo pair of images of a projected pattern are matched to find the disparity between the images and hence calculate the depth.

Active vision

Most computer vision systems rely on data captured by systems with fixed characteristics. These systems include passive sensing systems — such as video cameras — and active sensing systems — such as laser range finders. Bajcsy[15] has argued that, in contrast to these systems of data capture, an active vision system — in which the parameters and characteristics of data capture are dynamically controlled by the scene interpretation system — is crucial for perception. Active vision systems may employ either passive or active sensors; however, in an active vision system, the state parameters of the sensors — such as focus, aperture, vergence, and illumination — are controlled to acquire data that will facilitate scene interpretation. The concept of active vision is not new. Biological systems routinely acquire data in an active fashion. The advantages of accommodation in acquiring images for computer vision tasks were elaborated by Tenenbaum.[16] Aloimonos et al.[17] described the advantages of active vision systems

for performing many computer vision tasks, including shape recovery from image cues. Krotkov[18] described a stereo-image-capture system in which the focus, zoom, aperture, vergence, and illumination are actively controlled to obtain depth maps. In this system, information from several low-level-vision modules is combined to control the parameters of the imaging system. Control strategies for an active vision system are based on both local and global models.[15] Local models contain physical and geometric properties, whereas global models contain interaction among local models. Active perception is essentially an intelligent data acquisition process controlled by the measured and calculated parameters and errors from the scene. Precise definition of these scene- and context-dependent parameters requires a thorough understanding of not only the properties of the imaging and processing systems, but also of their interdependence.

Research trends

Computer vision research has been dominated by the study of imaging and image analysis techniques for monochromatic intensity images (except for multispectral images used in remote-sensing applications). Analyses of true color images and range images have become more common only recently. We have selected a set of seven papers that are representative of papers dealing with the topics covered in this chapter. Two are included in this book and the remaining five can be found in the companion book, *Computer Vision: Advances and Applications*. We begin this collection with the classic paper, "Understanding Image Intensities," by Horn in *Principles*. Relationships between image intensity, surface reflectance properties and orientation, and illumination are derived in this paper. In our companion book *Advances and Applications*, "Modeling Light Reflection for Computer Color Vision," by Lee et al., extends these concepts to include spectral reflectance properties of surfaces. This paper also contains experimental results of the application of their model to surfaces made up of different materials. Also in *Advances and Applications*, in "Color Constancy: A Method for Recovering Surface Spectral Reflectance," Maloney and Wandell describe an algorithm for estimating the spectral reflectance functions of surfaces even when information about the spectral distribution of ambient light is incomplete. In *Advances*, "Color From Black and White," by Funt and Ho, describes a technique that exploits the chromatic aberration in an optical-imaging system in order to separate the reflectance component from the illumination component. This technique, in principle, solves the color constancy problem.

In the last few years, range images (like structured light earlier) have been popular due to explicit specification of depth values. At one time, it was believed that if depth information were explicitly available, later processing would be easy. Research in the last few years has shown that — although depth information helps — the basic task of image interpretation retains all of its problems. Nevertheless, range images are finding increasing applications in many industrial applications, particularly in surface inspection.[12] The next three papers pertain to obtaining depth information. In *Principles*, "Active, Optical Range-Imaging Sensors," by Besl describes principles of operation of various range-imaging techniques. The paper also includes a comprehensive survey and an objective comparison of sensing methodologies and sensors, many of which are commercially available. Will and Pennington[14] introduced to computer vision the concept of structured lighting and many systems have been built using the basic principles they describe. In *Advances and Applications*, "Color-Encoded Structured Light for Rapid Active Ranging," by Boyer and Kak describes a range measurement system that exploits the additional degree of freedom provided by color to obtain a depth map from a single color-encoded structured light image.

Remote-sensing[19,20] and medical-imaging engineers have implemented many computer vision techniques to interpret images obtained using different sensors. Basic techniques, developed in computer vision for processing in early stages, can be applied to disparate sensory data used in various applications. Later processing stages require the embedding of image formation knowledge in interpretation programs. Current computer vision systems embed image formation knowledge implicitly during interpretation, making these programs ineffective for any other sensor. The trend to represent image formation knowledge explicitly is increasing, as shown by so-called "active vision approaches."[15] We conclude this chapter in *Advances and Applications* with "Active Perception," by Bajcsy, which strongly advocates integrating image acquisition and image interpretation within the framework of a cooperating control strategy. Bajcsy argues that such a system would simplify many difficult problems encountered by computer vision systems that operate upon data acquired by passive systems.

Many applications require that properties be represented in three-dimensional space. In such applications, three-dimensional sensory information is obtained. Such information differs from that obtained from range images, which are really two-dimensional. Three-dimensional information is common in various applications, including medical imaging,[21] oceanography, space, and fluid flow. The increasing use of computer vision techniques in these applications is sure to result in many new early-processing techniques in computer vision. Also, representation and processing of multisensory information will become more commonplace in computer vision research in the next decade. The current trend suggests that computer vision is fast becoming computer perception.

References Cited
Chapter 1

1. R.C. Gonzalez and P. Wintz, *Digital Image Processing*, second edition, Addison-Wesley, Reading, Mass., 1987.
2. B.K.P. Horn and R.W. Sjoberg, "Calculating the Reflectance Map," *Applied Optics*, Vol. 18, No. 11, 1979, pp. 1770-1779.
3. B.K.P. Horn, *Robot Vision*, McGraw-Hill, New York, N.Y., 1986.
4. A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, N.J., 1989.
5. D.H. Ballard and C.M. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, N.J., 1982.
6. S.A. Shafer, "Using Color to Separate Reflection Components," *Color: Research and Applications*, Vol. 10, No. 4, 1985, pp. 210-218.
7. H.C. Lee, E.J. Breneman, and C.P. Schulte, "Modeling Light Reflection for Computer Color Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, No. 4, 1990, pp. 402-409.
8. L.T. Maloney and B.A. Wandell, "Color Constancy: A Method for Recovering Surface Spectral Reflectance," *J. Opt. Soc. Am. A*, Vol. 3, No. 1, 1986, pp. 29-33.
9. B. Funt and J. Ho, "Color from Black and White," *Int'l. J. Computer Vision*, Vol. 3, No. 2, 1989, pp. 109-117.
10. P.J. Besl, "Active, Optical Range Imaging Sensors," *Machine Vision and Applications*, Vol. 1, No. 2, 1988, pp. 127-152.
11. R.A. Jarvis, "A Perspective on Range Finding Techniques for Computer Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 5, No. 2, 1983, pp. 122-139.
12. R.C. Jain and A.K. Jain, eds., *Analysis and Interpretation of Range Images*, Springer-Verlag, New York, N.Y., 1990.
13. K.L. Boyer and A.C. Kak, "Color-Encoded Structured Light for Rapid Active Ranging," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 9, No. 1, 1987, pp. 14-28.
14. P.M. Will and K.S. Pennington, "Grid Coding: A Novel Technique for Image Processing," *Proc. IEEE*, Vol. 60, No. 6, IEEE Press, New York, N.Y., 1972, pp. 669-680.
15. R. Bajcsy, "Active Perception," *Proc. IEEE*, Vol. 76, No. 8, IEEE Press, New York, N.Y., 1988, pp. 996-1005.
16. J.M. Tenenbaum, *Accommodation in Computer Vision*, PhD thesis, Stanford University, 1970.
17. J.Y. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active Vision," *Int'l. J. Computer Vision*, Vol. 1, 1988, pp. 333-356.
18. E. Krotkov, *Exploratory Visual Sensing for Determining Spatial Layout with an Agile Stereo Camera System*, PhD thesis, University of Pennsylvania, 1987.
19. R. Bernstein, *Digital Image Processing for Remote Sensing*, IEEE Press, New York, N.Y., 1978.
20. R.M. Hord, *Remote Sensing Methods and Applications*, John Wiley & Sons, New York, N.Y., 1986.
21. A.C. Kak and M. Slaney, eds., *Principles of Computerized Tomographic Imaging*, IEEE Press, New York, N.Y., 1988.

"Understanding Image Intensities" by B.K.P. Horn from *Artificial Intelligence*, Volume 8, 1977, pages 201-231.

ARTIFICIAL INTELLIGENCE

Understanding Image Intensities[1]

Berthold K. P. Horn

Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

Recommended by Max Clowes

ABSTRACT

Traditionally, image intensities have been processed to segment an image into regions or to find edge-fragments. Image intensities carry a great deal more information about three-dimensional shape, however. To exploit this information, it is necessary to understand how images are formed and what determines the observed intensity in the image. The gradient space, popularized by Huffman and Mackworth in a slightly different context, is a helpful tool in the development of new methods.

0. Introduction and Motivation

The purpose of this paper is to explore some of the puzzling phenomena observed by researchers in computer vision. They range from the effects of mutual illumination to the characteristic appearance of metallic surfaces—subjects which at first glance may seem to take us away from the central issues of artificial intelligence. But surely if artificial intelligence research is to claim victory over the vision problem, then it has to embrace the whole domain, understanding not only the problem solving aspects, but also the physical laws that underlie image formation and the corresponding symbolic constraints that enable the problem solving.

One reason for previous neglect of the image itself was the supposition that the work must surely already have been done by researchers in image processing, pattern recognition, signal processing and allied fields. There are several reasons why this attitude was misadvised:

Image processing deals with the conversion of images into new images, usually for human viewing. Computer image understanding systems, on the other hand, must work toward symbolic descriptions, not new images.

[1] This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's research is provided in part by the Advanced Research Projects Agency of the Department of Defence under Office of Naval Research contract N00014-75-C-0643.

Artificial Intelligence 8 (1977), 201–231

Pattern recognition, when concerned with images, has concentrated on the classification of images of characters and other two-dimensional input, often of a binary nature. Yet the world we want to understand is three-dimensional and the images we obtain have many grey-levels.

Signal processing studies the characteristics of transformations which are amenable to mathematical analysis, not the characteristics imposed on images by nature. Yet in the end, the choice of what to do with an image must depend on it alone, not the character of an established technical discipline.

Although we can borrow some of the techniques of each of these approaches we must still understand how the world forms an image if we are to make machines see. Yet I do not mean to suggest analysis by synthesis. Nothing of the sort! I propose only that if we are to solve the problem of creating computer performance in this domain, we must first thoroughly understand that domain.

This is, of course, not without precedent. The line of research beginning with Guzman and continuing through Clowes, Huffman, Waltz, Mackworth and others, was a study of how the physical world dictates constraints on what we see—constraints that once understood can be turned around and used to analyze what is seen with great speed and accuracy relative to older techniques which stressed problem solving expertise at the expense of domain understanding.

1. Developing the Tools for Image Understanding

An understanding of the visual effects of edge imperfections and mutual illumination will be used to suggest interpretations of image intensity profiles across edges, including those that puzzled researchers working in the blocks world. We shall see that a "sharp peak" or edge-effect implies that the edge is convex, a "roof" or triangular profile suggests a concave edge, while a step-transition or discontinuity accompanied by neither a sharp peak nor a roof component is probably an obscuring edge. This last hypothesis is strengthened significantly if an "inverse peak" or negative edge-effect is also present. (See Section 3.)

Next, it will be shown that the image intensities of regions meeting at a joint corresponding to an object's corner determine fairly accurately the orientation of each of the planes meeting at the corner. Thus we can establish the three-dimensional structure of a polyhedral scene without using information about the size or, support or nature of the objects being viewed. (See Section 3.4.)

Finally, we will turn to curved objects and show that their shape often can be determined from the intensities recorded in the image. The approach given here is supported by geometric arguments and does not depend on methods for solving first-order non-linear partial differential equations. It combines my previous shape-from-shading method [4, 2] with geometric arguments in gradient-space (Huffman and Mackworth [1, 2, 3, 9]). This approach to the image analysis problem enables us to establish whether or not certain features can be extracted from images. (See Sections 4 and 5.)

1.1. Image formation

The visual world consists of opaque bodies immersed in a transparent medium. The dimensionality of the two domains match: since only the object's surfaces are important for recognition and description purposes. On one hand we have two-dimensional surfaces plus depth and on the other, a two-dimensional image plus intensity. There are two parts to the problem of exploiting this observation to understand what is being imaged: one deals with the geometry of projection, the other with the intensity of light recorded in the image.

The relation between object coordinates and image coordinates is given by the well-known perspective projection equations derived from a diagram such as Fig. 1, where f if the focal length and,

$$x' = (x/z)f \quad \text{and} \quad y' = (y/z)f$$

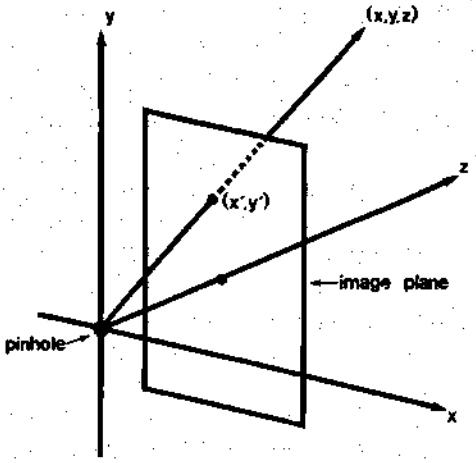


FIG. 1. Image projection geometry.

For the development presented here it will be convenient to concentrate on the case where the viewer is very far from the objects relative to their size. The resultant scene occupies a small visual angle as if viewed by a telephoto lens. This corresponds to orthographic projection, where z is considered constant in the equations above.

1.2. Surface orientation

We must understand the geometry of the rays connecting the lightsource(s), the object and the viewer in order to determine the light flux reflected to the viewer from a particular element of the object. The surface orientation in particular, plays a major role. There are, of course, various ways of specifying the surface orientation of a plane. We can give, for example, the equation defining the plane or the direction of a vector perpendicular to the surface. If the equation for the plane is $ax+by+cz = d$, then a suitable surface normal is (a, b, c).

We extend this method to curved surfaces simply by applying it to tangent planes. A local normal to a smooth surface is $(z_x, z_y, -1)$, where z_x and z_y are the first partial derivatives of z with respect to x and y. It is convenient to use the abbreviations p and q for these quantities. The local normal then becomes $(p, q, -1)$. It is clear then that the surface orientation thus defined has but two degrees to freedom. The quantity (p, q) is called the *gradient* (Fig. 2).

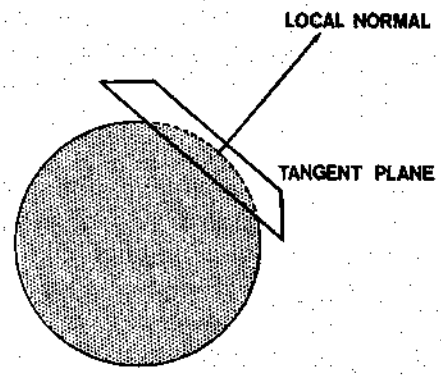


FIG. 2. Definition of surface orientation for curved objects.

1.3. Image intensity

The amount of light reflected by a surface element depends on its micro-structure and the distribution of incident light. Constructing a tangent plane to the object's surface at the point under consideration, we see that light may be arriving from directions distributed over a hemisphere. We first consider the contributions separately, from each of these directions and then superimpose the results.

For most surfaces there is a *unique value of reflectance and hence image intensity for a given surface orientation.* No matter how complex the distribution of light sources. We shall spend some time exploring this and develop the reflectance map in the process.

The simplest case is that of a single point-source where the geometry of reflection is governed by three angles: the incident, the emittance and the phase angles (Fig. 3). The incident angle, i, is the angle between the incident ray and the local normal,

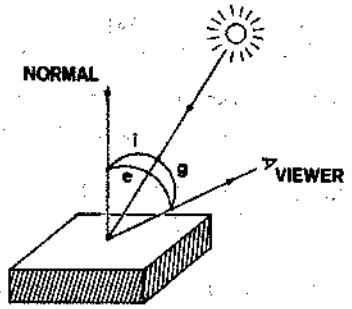


FIG. 3. The reflectivity is a function of the incident, emittance, and phase angles.

the emittance angle, e, is the angle between the ray to the viewer and the local normal, while the phase angle, g, is the angle between the incident and the emitted ray [4]. The reflectivity function is a measure of how much of the incident light is reflected in a particular direction. Superficially, it is the fraction of the incident light reflected per unit surface area, per unit solid angle in the direction of the viewer. More precisely:

Let the illumination be E (flux/area) and the resulting surface luminance in the direction of the viewer by B (flux/steradian/projected area). (The projected area is the equivalent foreshortened area as seen by the viewer.) *The reflectivity is then defined to be* B/E. It may be written $\phi(i, e, g)$, to indicate its dependence on the three angles involved.

Note that an infinitesimal surface element, $\mathrm{d}A$, captures a flux $E\cos(i)\,\mathrm{d}A$, since its surface normal is inclined i relative to the incident ray. Similarly, the intensity I(flux/steradian) equals $B\cos(e)\,\mathrm{d}A$, since the projected area is foreshortened by the inclination of the surface normal relative to the emitted ray.

1.4 Reflectivity function

For some surfaces, mathematical models have been constructed that allow an analytical determination of the reflectivity function. Since such techniques have rarely proved successful, reflectivity functions are usually determined empirically. Often there will be more than one source illuminating the object. In this case one has to integrate the product of reflectivity and the incident light intensity per unit solid angle over the hemisphere visible from the point under consideration. This determines the total light flux reflected in the direction of the viewer.

The normal to the surface relates object geometry to image intensity because it is defined in terms of the surface geometry, yet it also appears in the equation for the reflected light intensity. Indeed two of the three angles on which the reflectivity function depends are angles between the normal and other rays. Although we could now proceed to develop a partial differential equation based on this observation, it is more fruitful to introduce first another tool—gradient space.

1.5. Gradient space

Gradient-space can be derived as a projection of dual-space or of the Gaussian sphere but it is easier here to relate it directly to surface orientation [2]. We will concern ourselves with orthographic projection only, although some of the methods can be extended to deal with perspective projection as well.

The mapping from surface orientation to gradient-space is made by constructing a normal $(p, q, -1)$ at a point on an object and mapping it into the point (p, q) in gradient-space. Equivalently, one can imagine the normal placed at the origin and find its intersection with a plane at unit distance from the origin.)

We should look at some examples in order to gain a feel for gradient-space. Parallel planes map into a single point in gradient-space. Planes perpendicular to the view-vector map into the point at the origin of gradient-space. Moving away from the origin in gradient-space, we find that the distance from the origin equals the tangent of the emittance angle e, between the surface normal and the view-vector.

If we rotate the object-space about the view-vector, we induce an equal rotation of gradient-space about the origin. This allows us to line up points with the axes and simplify analysis. Using this technique, it is easy to show that the angular position of a point in gradient-space corresponds to the direction of steepest descent on the original surface.

Let us call the orthogonal projection of the original space the image-plane. Usually this is all that is directly accessible. Now consider two planes and their intersection. Let us call the projection of the line of intersection the image-line. The two planes, of course, also correspond to two points in gradient space. Let us call the line connecting these two points the gradient-line. Thus, a line maps into a line. The perpendicular distance of the gradient-space line from the origin equals the tangent of the inclination of the original line of intersection with respect to the image plane. We show by superimposing gradient-space on the image-space [2, 11] that the gradient-space line and the image-line are *mutually perpendicular*. Mackworth's scheme for scene analysis of line drawings of polyhedra depends on this observation [2].

1.6. Trihedral corners

The points in gradient-space corresponding to the three planes meeting at a trihedral corner must satisfy certain constraints. The lines connecting these points must be perpendicular to the corresponding lines in the image-plane (Fig. 4). This provides us with three constraints but that is not enough to fix the position of three points in gradient-space. Three degrees of freedom—the *position* and *scale* of the triangle—remain undetermined. We see later that measurement of image intensities for the three planes provides enough information to specify their orientations, thus allowing a determination of the three-dimensional structure of a polyhedral scene.

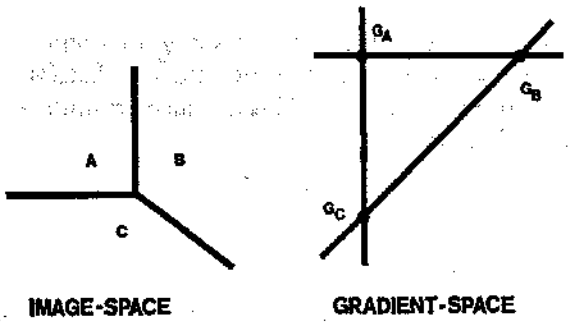


FIG. 4. Constraints on the gradient-space points corresponding to the planes meeting at a trihedral corner. The gradient-lines must be perpendicular to the image-lines.

2. The Reflectance Map

The amount of light reflected by a given surface element depends on the orientation of the surface and the distribution of light sources around it, as well as on the nature of the surface. For a given type of surface and distribution of light sources, there is a fixed value of reflectance for every orientation of the surface normal and hence for every point in *gradient-space*. Thus image intensity is a single-valued function of p and q. We can think of this as a map in gradient-space. This is *not* a transform of the image seen by the viewer. It is, in fact, independent of the scene and instead, only a function of the surface properties and the light source distribution (but see Section 2.6).

What can we do with this strange "image" of the world surrounding the object? If we measure a certain intensity at a given point on the object, we know that the orientation of the surface at that point is restricted to a subset of all possible orientations; we cannot, however, uniquely determine the orientation. The one constraint is that it be one of the points in gradient-space where we find this same value of intensity.

The use of the gradient-space diagram is analogous to the use of the hodogram or velocity-space diagram. The latter provides insight into the motion of particles in force field that is hard to obtain by algebraic reasoning alone. Similarly, the gradient-space allows geometric reasoning about surface orientation and image intensities.

2.1. Matte surfaces and a point-source near the viewer

A perfect lambertian surface or diffuser looks equally bright from all directions; the amount of light reflected depends only on the cosine of the incident angle. In order to postpone the calculation of incident, emittance and phase angles from p and q, we first place a single light source near the viewer. The incident angle then equals the emittance angle, the angle between the surface normal and the view-vector. The cosine of the incident angle is the dot product of the corresponding unit vectors:

$$\cos(i) = \frac{(p, q, -1) \cdot (0, 0, -1)}{|(p, q, -1)|\,|(0, 0, -1)|} = \frac{1}{\sqrt{1+p^2+q^2}}.$$

We obtain the same result by remembering that the distance from the origin in gradient-space is the tangent of the angle between the surface normal and the view-vector:

$$\sqrt{p^2+q^2} = \tan(e),$$

$$\cos(e) = \frac{1}{\sqrt{1+\tan^2(e)}},$$

and

$$e = i.$$

If we plot reflectance as a function of p and q, we obtain a central maximum of 1 at the origin and a circularly symmetric function that falls smoothly to 0 as we approach infinity in gradient-space. This is a nice, smooth reflectance map, typical of matte surfaces (Fig. 5).

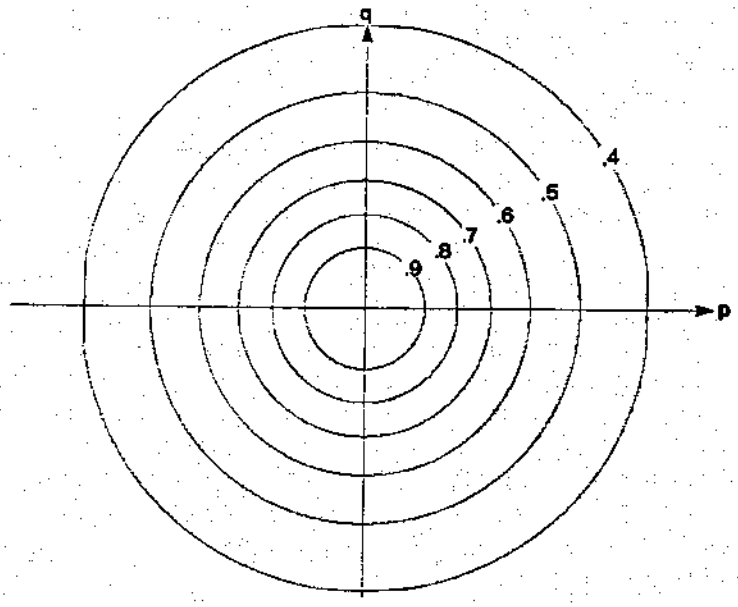


FIG. 5. Contours of $E = \cos(e)$. This is the reflectance map for objects with lambertian surfaces when there is a single light source and the light source is near the viewer.

A given image intensity corresponds to a simple locus in gradient-space, in this case a circle centered on the origin. A measurement of image intensity tells us that the surface gradient falls on a certain circle in gradient-space.

Since the light-source is not always likely to be near the viewer we now explore the more complicated geometry of incident and emitted rays for arbitrary directions of incident light at the object.

2.2. Incident, emittance, and phase angles

For many surfaces the reflectance is a smooth function of the angles of incidence, emittance and phase. It is convenient to work with the cosines of these angles, $I = \cos(i)$, $E = \cos(e)$, and $G = \cos(g)$. (These can be obtained easily from the dot products of the unit vectors.) If we have a single distant light source whose direction is given by a vector $(p_s, q_s, -1)$, and note that the view-vector is $(0, 0, -1)$, then,

$$G = \frac{1}{\sqrt{1+p_s^2+q_s^2}}, \qquad E = \frac{1}{\sqrt{1+p^2+q^2}},$$

and

$$I = \frac{(1+p_s p+q_s q)}{\sqrt{1+p^2+q^2}\sqrt{1+p_s^2+q_s^2}} = (1+p_s p+q_s q)EG.$$

It is simple to calculate I, E, and G for any point in gradient-space. In fact G is *constant* given our assumption of orthogonal projection and distant light source. We saw earlier that the contours of constant E are circles in gradient-space centered on the origin. Setting I constant gives us a second-order polynomial in p and q and suggests that loci of constant I may be conic sections. The terminator—the line separating lighted from shadowed regions, for example, is a straight line obtained by setting $i = \pi/2$. That is, $I = 0$; or $1+p_s p+q_s q = 0$. Similarly, the locus of $I = 1$ is the single point $p = p_s$ and $q = q_s$.

A geometrical way of constructing the loci of constant I is to develop the cone generated by all directions that have the same incident angle. The axis of the cone is the direction to the light-source $(p_s, q_s, -1)$. We find the corresponding points in gradient-space by intersecting this cone with a plane at unit distance from the origin. Varying values of I produce cones with varying angles. The cones form a nested sheaf. The intersection of this nested sheaf with the unit plane is a nested set of conic sections (Fig. 6). Note that our previous example (Fig. 5) is merely a special case in which the axis of the sheaf of nested cones points directly at the viewer.

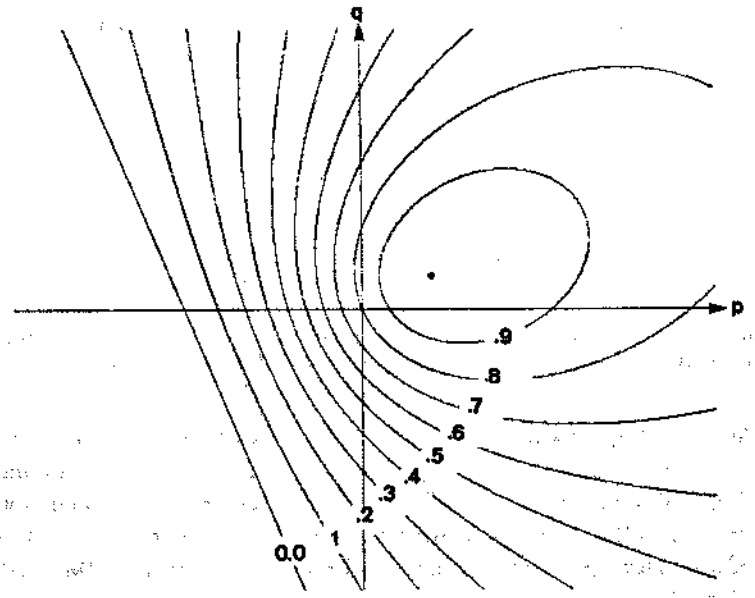


FIG. 6. Contours of $I = \cos(i)$. The direction to the single source is $(p_s, q_s) = (0.7, 0.3)$. This is the reflectance map for objects with lambertian surfaces when the light source is not near the viewer.

If we measure a particular image intensity, we know that the gradient of the corresponding surface element must fall on a particular conic section. The possible normals are confined to a cone—in this case, merely a circular cone. In the case of more general reflectivity functions, the locus of possible normals constitutes a more general figure called the *Monge cone.*

2.3. Specularity and glossy surfaces

Many surfaces have some specular or mirror-like reflection from the outermost layers of their surface, and thus are not completely matte. This is particularly true of surfaces that are smooth on a microscopic scale. For specular reflection $i = e$, and the incident, emitted, and normal vectors are all in the same plane. Alternatively, we can say that $i+e = g$. In any case, only one surface orientation is correct for reflecting the light source towards the viewer. That is, a perfect specular reflection contributes an impulse to the gradient-space image at a particular point.

In practice, few surfaces are perfectly specular. Glossy surfaces reflect some light in directions slightly away from the geometrically correct direction [8]. It can be shown that the cosine of the angle between the direction for perfectly specular reflection and any other direction is $(2IE-G)$ [11]. This clearly equals 1 in the ideal direction and falls off towards 0 as the angle increases to a right-angle. By taking various functions of $(2IE-G)$, such as high powers, one can construct more or less compact specular contributions.

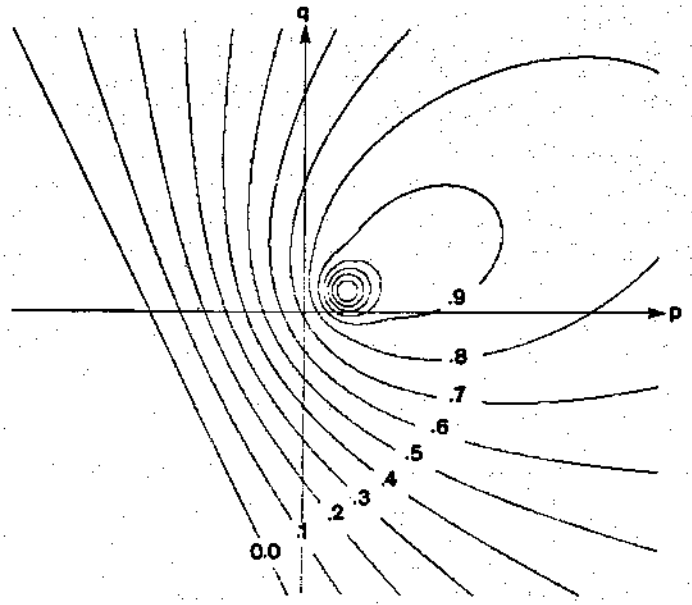


FIG. 7. Contours for $\phi(I, E, G) = \frac{1}{2}s(n+1)(2IE-G)^n+(1-s)I$. This is the reflectance map for a surface with both a matte and a specular component of reflectivity illuminated by a single point-source.

For example, a good approximation for some glossy white paint can be obtained by combining the usual matte component with a specular component defined in this way—$\phi(I, E, G) = \frac{1}{2}s(n+1)(2IE-G)^n+(1-s)I$. Here s lies between 0 and 1 and determines the fraction of incident light reflected specularly before penetrating the surface, while n determines the sharpness of the specularity peak in the gradient-space image (Fig. 7).

2.4. Finding the gradient from the angles

In order to further explore the relation between the specification of surface orientation in gradient-space and the angles involved, we solve for p and q, given I, E, and G. We have already shown that the opposite operation is simple to perform. One approach to this problem is to solve the polynomial equations in p and q derived from the equations for I, E, and G. It can be shown that [11]:

$$p = p' \cos(\theta) - q' \sin(\theta),$$
$$q = p' \sin(\theta) + q' \cos(\theta),$$

where

$$p' = \frac{(I/E-G)}{\sqrt{1-G^2}} \quad \text{and} \quad q' = \frac{(\Delta/E)}{\sqrt{1-G^2}},$$
$$\Delta^2 = 1+2IEG-(I^2+E^2+G^2),$$
$$\cos(\theta) = \frac{p_s}{\sqrt{p_s^2+q_s^2}} \quad \text{and} \quad \sin(\theta) = \frac{q_s}{\sqrt{p_s^2+q_s^2}}.$$

It is immediately apparent that there are two solution points in gradient space for most values of I, E, and G. Notice that θ is the direction of the light source in gradient-space, that is, the line connecting (p_s, q_s) to the origin makes an angle θ with the p-axis. So p' and q' are coordinates in a new gradient-space obtained after

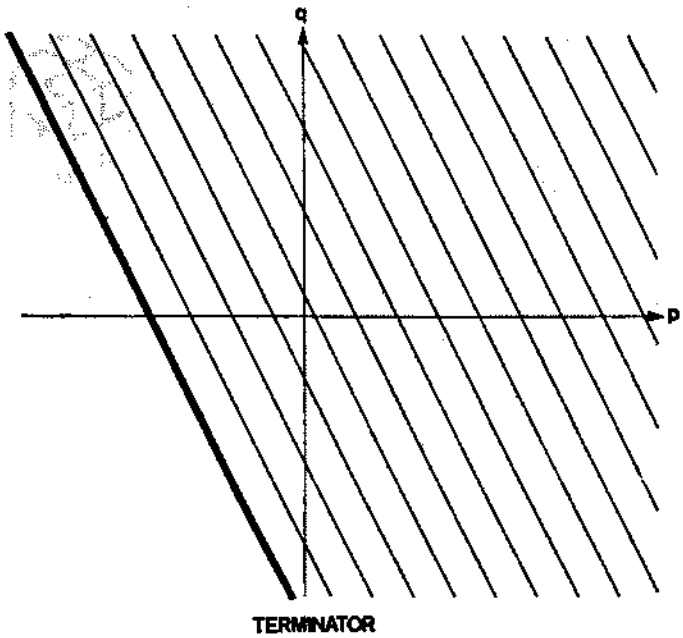


FIG. 8. Contours of $\phi(I, E, G) = I/E$. Contour intervals are 0.2 units wide. The reflectivity function for the material in the maria of the moon is constant for constant I/E (constant luminance longitude).

simplifying matters by rotating the axes until $q_s = 0$. The light source is then in the direction of the x'-axis. Notice that p' is constant if I/E is constant (remembering that G is constant anyway.) Hence the loci of constant I/E are straight lines. These lines are all parallel to the terminator, for which $I = 0$. This turns out to be important since some surfaces have constant reflectance for constant I/E (Fig. 8).

2.5. Smooth metallic surfaces

Consider a smooth metallic surface: a surface with a purely specular or mirror-like reflectance. Each point in gradient-space corresponds to a particular direction of the surface normal and defines a direction from which incident light has to approach the object in order to be reflected towards the viewer. In fact, we can produce a complete map of the sphere of possible directions as seen from the object. At the origin, for example, we have the direction towards the viewer. Now for each incident direction there is a certain light-intensity depending on what objects lie that way. Consider recording these intensities at the corresponding points in gradient-space. Clearly one obtains some kind of image of the world surrounding the given metallic object. In fact, one develops a stereographic projection, a plane projection of a sphere with one of the poles as the center of projection. Another way of looking at it is that the image we construct in this fashion is like one we obtain by looking into a convex mirror—a metallic paraboloid to be precise (Fig. 9).

In order to construct reflectance maps for various surfaces and distributions of light sources, we superimpose the results in gradient-space for each light source in turn. We now examine a flaw in this approach and attempt a partial analysis of mutual illumination.

2.6. Mutual illumination

The reflectance map is based on the assumption that the viewer and all light sources are distant from the object. Only under these assumptions can we associate a

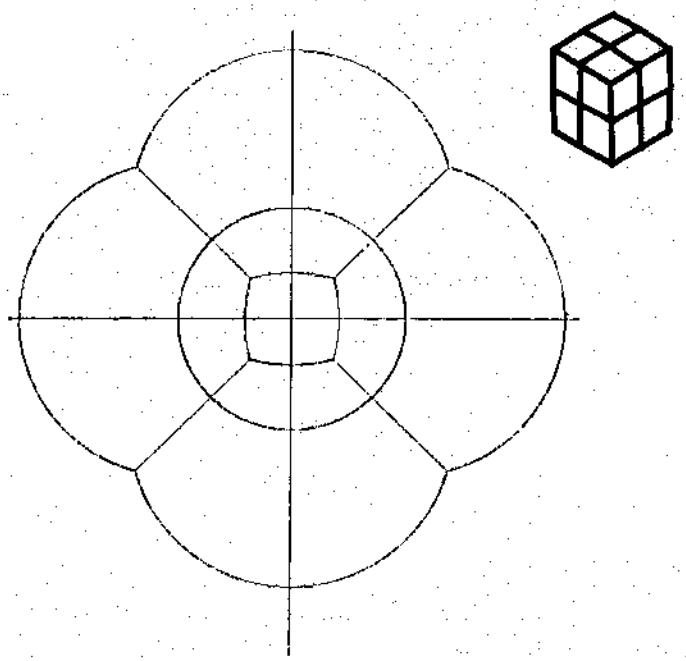


FIG. 9. Reflectance map for a metallic object in the center of a very large wire cube. Equivalently one can think of it as the reflection of the wire cube in a paraboloid with a specularly-reflecting surface.

unique value of image intensity with every surface orientation. If the scene consists of a single convex object these assumptions will be satisfied, but when there are several highly reflective objects placed near one another, mutual illumination may become important. That is, the distribution of incident light no longer depends only on direction, but is a function of position as well. The general case is very difficult, and we shall only study some idealized situations applicable to scenes consisting of polyhedra.

Two important effects of mutual illumination are a reduction in *contrast* between faces, and the appearance of *shading* or gradation of light on images of plane surfaces. In the absence of this effect, we would expect plane surfaces to give rise to images of uniform intensity since all points on a plane surface have the same orientation.

2.7. Two semi-infinite planes

First, let us consider a highly idealized situation of two semi-infinite planes joined at right angles and a distant light source. Let the incident rays make an angle α with respect to one of the planes (Fig. 10). Assume further that the surfaces reflect

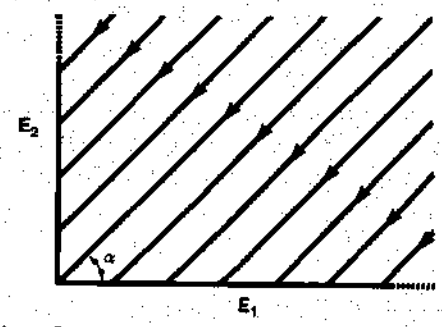


FIG. 10. Mutual illumination of two infinite lambertian planes of reflectance r, when illuminated by a single distant source.

a fraction r of the light falling on them, and that the illumination provided by the source is E (light flux/unit area). Picking any point on one of the half-planes, we find that one-half of its hemisphere of directions is occupied by the other plane, one-half of the light radiated from this point hits the other plane, while one-half is lost. Since both planes are semi-infinite, the geometry of this does not depend on the distance from the corner. The light incident at any point is made up of two components: that received directly from the source and that reflected from the other plane. The intensity on one plane does not vary with distance from the corner—a point receives light from one-half of its hemisphere of directions no matter what its distance from the corner. Put another way, there is no natural scale factor for a fluctuation in intensity. Let the illumination of the plane be E_1 and E_2 (light flux/unit area); then,

$$E_1 = \tfrac{1}{2}rE_2 + E\cos(\alpha),$$
$$E_2 = \tfrac{1}{2}rE_1 + E\sin(\alpha).$$

Solving for E_1 and E_2, we get:

$$E_1 = E[\cos(\alpha) + \tfrac{1}{2}r\sin(\alpha)]\frac{1}{[1-(\tfrac{1}{2}r)^2]},$$

$$E_2 = E[\sin(\alpha) + \tfrac{1}{2}r\cos(\alpha)]\frac{1}{[1-(\tfrac{1}{2}r)^2]}.$$

Had we ignored the effects of mutual illumination we would have found $E_1 = E\cos(\alpha)$ and $E_2 = E\sin(\alpha)$. Clearly the effect increases as reflectance r increases (it is not significant for dark surfaces). When the planes are illuminated equally, for $\alpha = \pi/4$, we find

$$E_1 = E_2 = (E/\sqrt{2})/(1-\tfrac{1}{2}r).$$

When $r = 1$, we obtain *twice* the illumination and hence twice the brightness than that obtained in the absence of mutual illumination. If the angle between the two planes varies, we find that the effect becomes larger and larger as the angle becomes more and more acute. By choosing the angle small enough, we can obtain arbitrary "amplification". Conversely, for angles larger than $\pi/2$, the effect is less pronounced.

In the derivation above, we did not make very specific assumptions about the angular distribution of reflected light, just that it is symmetrical about the normal and that it does not depend on the direction from which the incident ray comes. Hence, a lambertian surface is included, while a highly specular one is not. The effect is indeed less pronounced for surfaces with a high specular component of reflection, since most of the light is bounced back to the source after two reflections. Another important thing to note is that even if the planes are not infinite, the above calculations are approximately valid close to the corner. For finite planes we expect a variation of intensity as a function of distance from the corner; the results derived here apply asymptotically as one approaches the corner.

2.8. Two truncated planes

The geometry becomes quite complex if the planes are of finite extent, but we can develop an integral equation if we allow the planes to be infinite along their line of intersection and truncate them only in the direction perpendicular to this. Suppose two perpendicular planes extend a distance L from the corner, and that $\alpha = \pi/4$. This produces a particularly simple integral equation [11]; nevertheless I have been unable to solve it analytically. Numerical methods show that the resultant illumination falls off monotonically from the corner (Fig. 11), that the value at the corner is

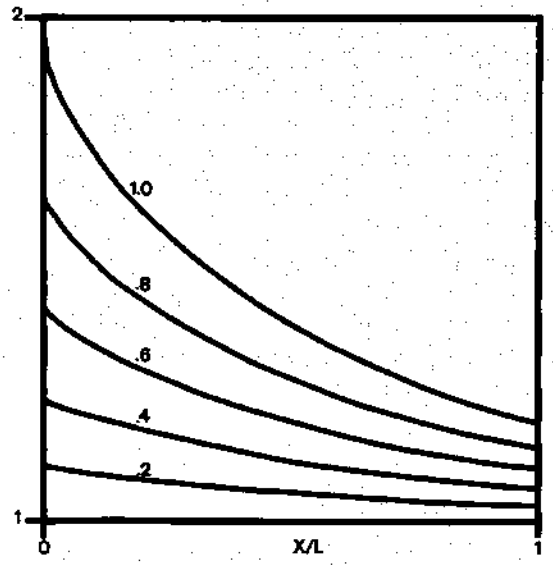


FIG. 11. Surface luminance plotted versus fractional distance from a right-angle corner. The curves are for reflectances of 0.2, 0.4, 0.6, 0.8, and 1.0.

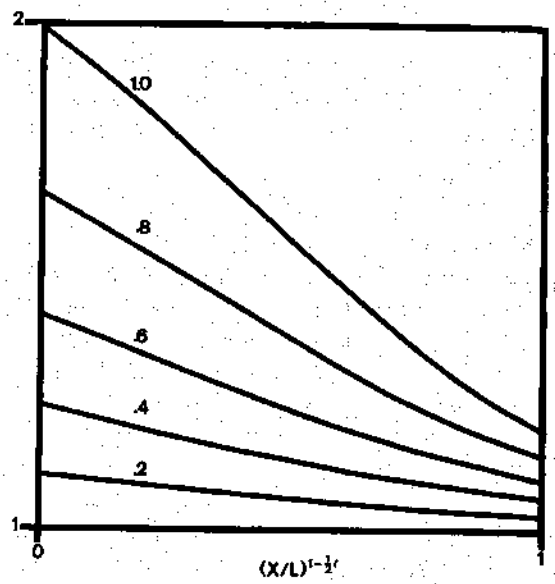


FIG. 12. Surface luminance plotted versus $(x/L)^{(1-\frac{1}{2}r)}$ to illustrate asymptotic behavior near the corner. The curves correspond to reflectances of 0.2, 0.4, 0.6, 0.8, and 1.0.

indeed what was predicted in the previous section, and the fall-off near the corner is governed by a term in $-(x/L)^{(1-\frac{1}{2}r)}$ (Fig. 12). A good approximation appears to be $1+E_{2/r}(x/L)$, where $E_n(x)$ is the elliptical integral and x is the distance along the plane measured from the edge where the planes meet.

3. The Semantics of Edge-Profiles

We are now ready to apply the tools developed so far. First let us consider the interpretation of intensity profiles taken across edges. If polyhedral objects and image sensors were perfect, if there were no mutual illumination, and if light sources were distant from the scene, images of polyhedral objects would be divided into polygonal areas, each of uniform intensity. It is well known that in real images, image intensity varies within the polygonal areas and that an intensity profile taken across an edge separating two such polygonal regions does not have a simple step-shaped intensity transition. Herskovitz and Binford determined experimentally that the most common edge transitions are step-, peak-, and roof-shaped [7] (Fig. 13). So

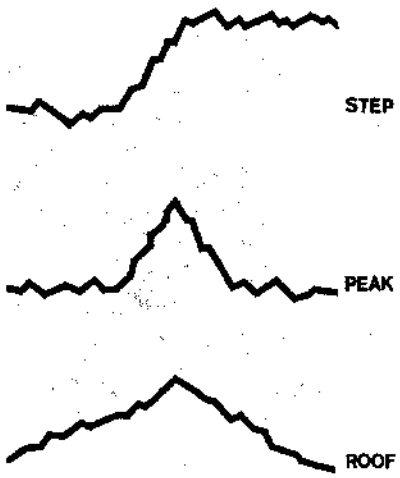


FIG. 13. Most common intensity profiles across images of polyhedral edges.

far this has been considered no more than a nuisance, because it complicates the process of finding edges. We now discuss the interpretation of these profiles in terms of the three-dimensional aspects of the scene.

3.1. Imperfections of polyhedral edges

A perfect polyhedron has a discontinuity in surface normal at an edge. In practice, edges are somewhat rounded off. A cross-section through the object's edge show that the surface normal varies smoothly from one value to the other and takes on values that are linear combinations of the surface normals of the two adjoining planes (Fig. 14). What does this mean in terms of reflected light intensity? Intensity varies smoothly at the edge, instead of jumping from one surface normal value to

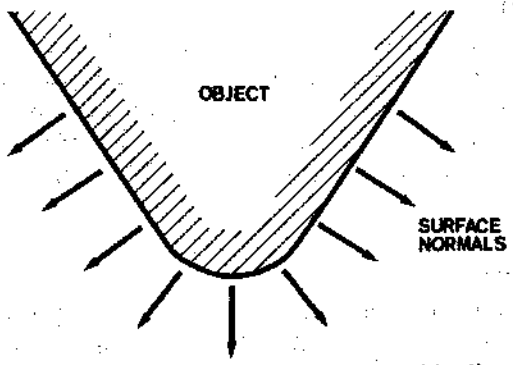


FIG. 14. The normals at an imperfect, rounded off edge are positive linear combinations of the normals of the two adjoining faces.

another. The important point is that it may take on values *outside* the range of those defined by the two planes. The best way to see this is to consider the situation in gradient-space. The two planes defined two points in gradient-space. Tangent planes on the corner correspond to points on the line connecting these two points. If the image intensity is higher for a point somewhere on this line, we will see a peak in the intensity profile across the edge. (Fig. 15.)

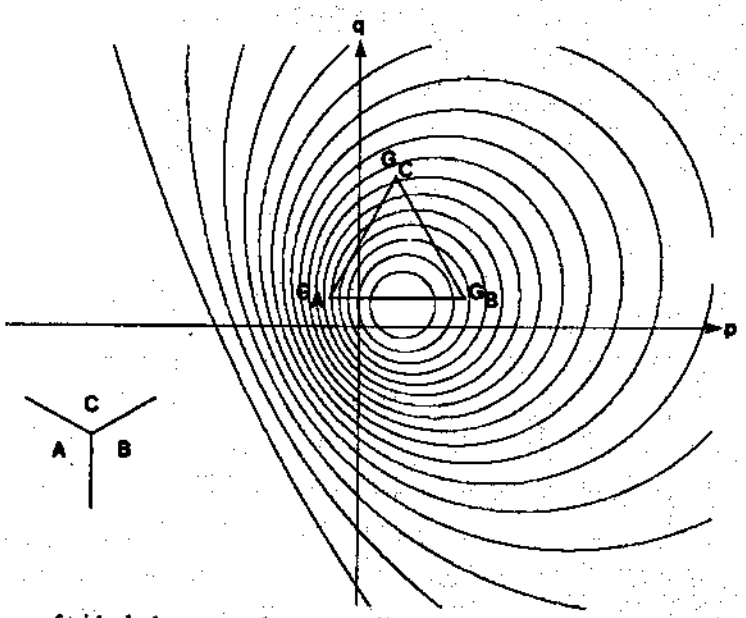


FIG. 15. Image of tri-hedral corner and corresponding gradient-space diagram. The image intensity profile across the edge between face *A* and face *B* will have a peak or highlight. The others will not.

If we find an edge profile with a peak shape or step with a peak superimposed, it is most likely that the corresponding edge is convex. The converse is not true (an edge may be convex and not give rise to a peak) if the line connecting the two points in gradient space has intensity varying monotonically along its length. The identification is also not completely certain since under peculiar lighting conditions with objects that have acute angles between adjacent faces, a peak may appear at an obscuring edge. Notice that the peak is quite compact, since it extends only as far as the rounded-off edge.

At a corner, where the planes meet, we find that surface imperfections provide surface normals that are linear combinations of the three normals corresponding to the three planes. In gradient space, this corresponds to points in the triangle connecting the three points corresponding to the planes. If this triangle contains a maximum in image intensity we expect to see a highlight right on the corner (Fig. 15).

3.2. Mutual illumination

We have seen that mutual illumination gives rise to intensity variations on planar surfaces—intensity roughly decreases linearly away from the corner. Notice that this affects the intensity profile over a large distance from the edge, quite unlike the sharp peak found due to edge imperfections. Clearly, if we find a roof-shaped profile or step with a roof-shaped superimposed we should consider labelling the edge concave.

The identification is, however, partly unreliable since some imaging device defects can produce a similar effect. Image dissectors, for example suffer from a great deal of scattering—areas further from a dark background appear brighter. So we may see a smoothed version of a roof-shape in the middle of a bright scene against a dark background. Experimentation with high quality image input devices such as a PIN-diode mirror-deflection system has confirmed that this is an artifact introduced by the image dissector. When the light source is close to the scene, significant gradients can appear on planar surfaces as pointed out by Herskovitz and Binford [7]. Lastly, the roof-shaped profiles on the two surfaces may be due to mutual illumination with other surfaces, not each other. Nevertheless, a roof-shaped profile usually suggests a concave edge.

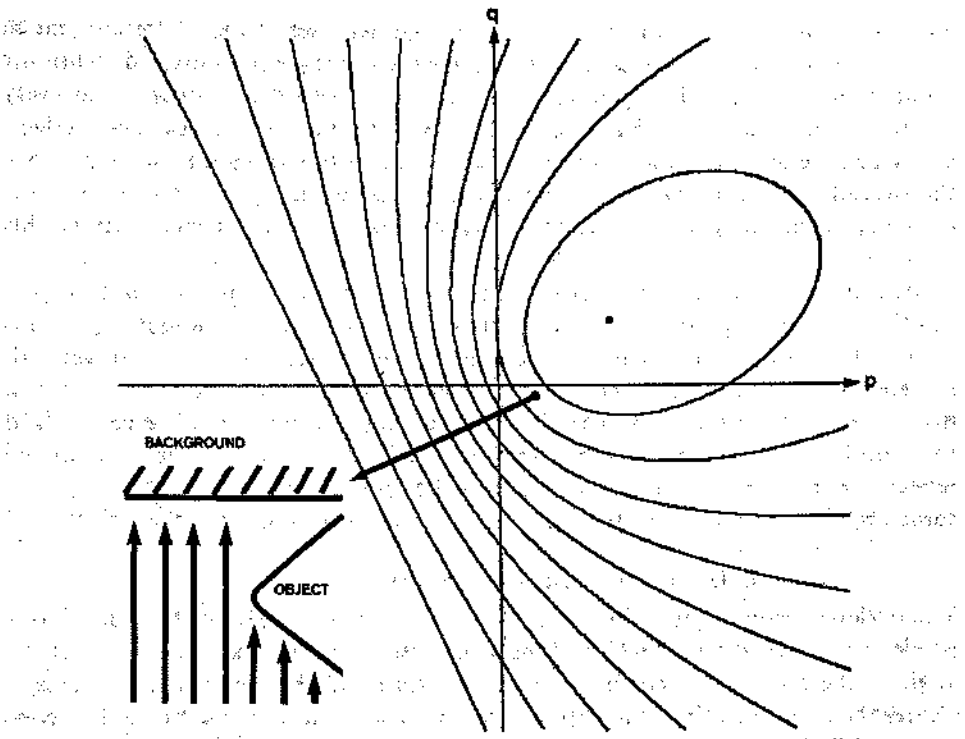


FIG. 16. Generation of a negative peak at an obscuring edge facing away from the light source.

3.3. Obscuring edges

Although they can be found with both convex and concave edges, step-shaped intensity profiles occur most often when objects obscure one another. If the obscuring surface adjoins a self-shadowed surface however, edge imperfections will produce a negative peak on the profile, since the line connecting the points corresponding to the two surfaces of the object passes through the terminator in gradient-space. Hence a negative peak or a step with a superimposed negative peak strongly suggests obscuration. Unfortunately it is difficult to tell on which side the obscuring plane lies (Fig. 16).

3.4. Determining the three-dimensional structure of polyhedral scenes

Mackworth's approach to understanding line drawings of polyhedra allows us to take into account some of the quantitative aspects of the three-dimensional geometry of scenes [2]. It does not, however, allow us to fully determine the orientation of all the planes. The scale and position of the gradient space diagram is undetermined by his technique. To illustrate this, consider a single trihedral corner. Here we know that the three points in gradient space representing the three planes meeting at the corner have to satisfy certain constraints: they must lie on three lines perpendicular to the image lines (see fig. 4).

It takes six parameters to specify the position of three points on a plane, leaving three degrees of freedom after the introduction of these constraints. Measuring the three image intensities of the planes supplies another three. The constraints are due to the fact that the points in gradient-space have to lie on the correct contours of image intensity. The triangle can be stretched and moved until the points correspond to the correct image intensities as measured for the three planes.

Since this process corresponds to solving three non-linear equations for three unknowns, we can expect a finite number of solutions. Often there are but one or two—prior knowledge often eliminates some. (For a numerical example, see Appendix).

When more than three planes meet at a corner, the equations are over-determined,

and the situation is even more constraint. Conversely, we cannot determine much with just two planes meeting at an edge—there are too few equations and an infinite number of solutions. The possible ambiguity at a trihedral corner is not very serious when we consider that in a typical scene there are many "connect" edges, either convex or concave as determined by Mackworth's program. In such a case the overall constraints may allow only one consistent interpretation. A practical difficulty is that it is unclear which search strategy leads most efficiently to this interpretation.

Measurements of image intensity are not very precise and surfaces have properties that vary from point to point as well as with handling. We cannot expect this method to be extremely accurate in pinning down surface orientation. In general, the equations for a typical scene are over-determined; a least-squares approach may improve matters slightly. The idea of stretching and shifting can be generalized to smooth surfaces. We know that the image of a paraboloid is the same as the reflectance map. If we can stretch and shift the reflectance map to fit the image of some object, then the same deformations turn the paraboloid into the object.

4. Determining Lunar Topography

When viewed from a great distance, the material in the maria of the moon has a particularly interesting reflectivity function. First, note that the lunar phase is the angle at the moon between the light source (sun) and the viewer (earth). This is clearly the angle we call g, and explains why we use the term phase angle. For constant phase angle, detailed measurements using surface elements whose projected area as seen from the source is a constant multiple of the projected area as seen by the viewer have shown that all such surface elements have the same reflectance. But the area appears foreshortened by $\cos(i)$ and $\cos(e)$ as seen by the source and the viewer respectively. Hence the reflectivity function is constant for constant $\cos(i)/\cos(e) = I/E$.

In this case each surface element scatters light uniformly into its hemisphere of directions, quite unlike the lambertian surface which favors directions normal to its surface. This is not an isolated incident. The surfaces of other rocky, dusty objects when viewed from great distances appear to have similar properties. For example, the surface of the planet Mercury and perhaps Mars as well as some asteroids and atmosphere-free satellites fit this pattern. Surfaces with reflectance a function of I/E thus form a third species we should add to matte surfaces where the reflectance is a function of I and glossy surfaces where the reflectance is a function of $(2IE-G)$.

4.1. Lunar reflectivity function

Returning to the lunar surface, we find an early formula due to Lommel Seelinger [6].

$$\phi(I, E, G) = \frac{\Gamma_0(I/E)}{(I/E)+\lambda(G)}.$$

Here Γ_0 is a constant and the function $\lambda(G)$ is defined by an empirically determined table. A somewhat more satisfactory fit to the data is provided by a formula of Fesenkov's [6]:

$$\phi(I, E, G) = \frac{\Gamma_0(I/E)[1+\cos^2(\alpha/2)]}{(I/E)+\lambda_0[1+\tan^2(\lambda/2)]}.$$

Where Γ_0 and λ_0 are constants and $\tan(\alpha) = -p' = -(I/E-G)/\sqrt{1-G^2}$. This formula is also supported by a theoretical model of the surface due to Hapke. Note that given I, E, and G, it is straightforward to calculate the expected reflectance. We need to go in the reverse direction and solve for I/E given G and the reflectance as measured by the image intensity. While it may be hard to invert the above equation analytically, it should be clear that by some iterative, interpolation, or hill-climbing scheme one can solve for I/E. We shall ignore for now the ambiguities that arise if there is more than one solution.

4.2. Lunar reflectance map

Next, we ask what the reflectance map looks like for the lunar surface illuminated by a single point source. The contours of constant intensity in gradient-space will be lines of constant I/E. But the contours of constant I/E are straight lines! So the gradient-space image can be generated from a single curve by shifting it along a straight-line—the shadow-line, for example (see Fig. 8). The contour lines are perpendicular to the direction defined by the position of the source (that is, the line from the origin to p_s, q_s).

Now what information does a single measurement of image intensity provide? It tells us that the gradient has to be on a particular straight line. Again, we ignore for the moment the possible existence of more than one contour for a given intensity.

What we would like to know, of course, is the orientation of the surface element. We cannot completely determine the local orientation, but we *can* determine its component in the direction perpendicular to the contour lines in gradient-space.

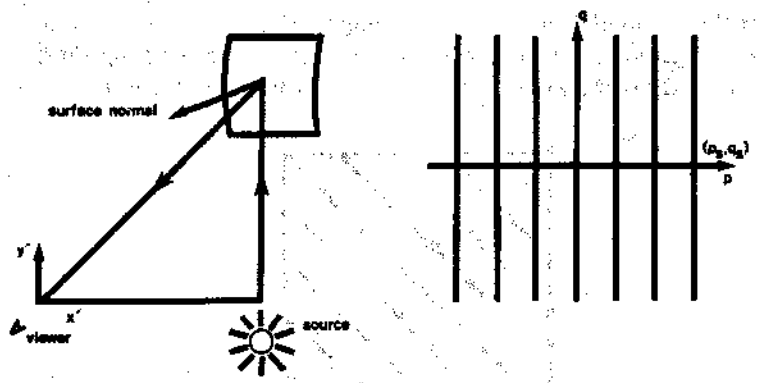


FIG. 17. Rotation of coordinate system to simplify gradient-space geometry.

We can tell nothing about it in the direction at right angles. That is, knowing I/E and G determines p', as previously defined, and tells us nothing about q'.

This favored direction lies in the plane defined by the source, the viewer, and the surface element under consideration. If we wish, we can simplify matters by rotating the viewer's coordinate system until the x axis lies in this plane as well. Then $q_s = 0$, and the contours of constant intensity in gradient-space are all vertical lines. Evidently, an image intensity measurement determines the slope of the surface in the x' direction, without telling us anything about the slope in the y' direction (Fig. 17). We are now ready to develop the surface by advancing in the direction in which we can determine locally the surface slope.

4.3. Finding a surface profile by integration

We have:

$$p' = \frac{dz}{ds} = \frac{I/E-G}{\sqrt{1-G^2}}.$$

The distance s from some starting point is measured in the object coordinate system and is related to the distance along the curve's projection in the image by $s' = s(f/z_0)$.

$$\frac{dz}{ds'} = \frac{f}{z_0}\frac{I/E-G}{\sqrt{1-G^2}}.$$

Integrating, we get:

$$z(s') = z_0 + \frac{f}{z_0}\int_0^{s'} \frac{I/E-G}{\sqrt{1-G^2}}\,ds',$$

where I/E is found from G and the image intensity $b(x', y')$. Starting anywhere in the image, we can integrate along a particular line and find the relative elevation of the corresponding points on the object.

The curves traced out on the object in this fashion are called *characteristics*, and their projections in the image-plane are called *base characteristics*. It is clear that here the base characteristics are parallel straight lines in the image, independent of the object's shape.

4.4. Finding the whole surface

We can explore the whole image by choosing sufficient starting points along a line at an angle to the favored direction. In this way we obtain the surface shape over the whole area recorded in the image (Fig. 18). Since we cannot determine the gradient at right angles to the direction of the characteristics, there is nothing to relate adjacent characteristics in the image. We have to know an initial curve, or use assumptions about reasonable smoothness. Alternatively, we can perform a second surface calculation from an image taken with a different source-surface-observer geometry. In this case, we obtain solutions along lines crossing the surface at a different angle, tying the two solutions together. This is not

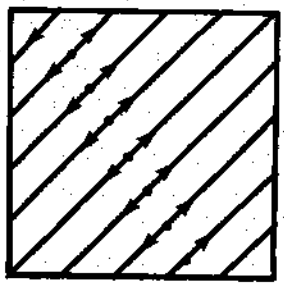


FIG. 18. Finding the shape of the surface by integrating along several base characteristics starting from different initial points.

quite as useful as one might first think, because it does not apply to pictures taken from earth. The plane of the sun, moon and earth varies little from the ecliptic plane. The lines of integration in the image vary little in inclination. This idea however *does* work for pictures taken from near the moon.

4.5. Ambiguity in local gradient

What if more than one contour in gradient-space corresponds to a given intensity? Then we cannot tell locally which gradient to apply. If we are integrating along some curve however, this is no problem, since we may assume that there is little change in gradient over small distances, and pick one close to the gradient last used. This assumption of smoothness leaves us with one remaining problem: what happens if we approach a maximum of intensity in gradient-space and then enter areas of lower intensity (Fig. 19). Which side of the local maximum do we slide down?

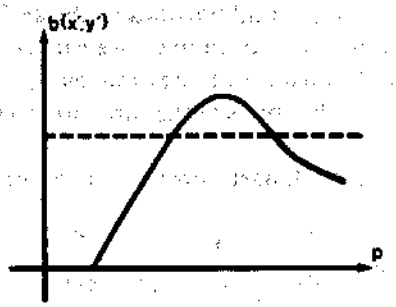


FIG. 19. Problem of ambiguity caused by non-uniqueness of slope for a particular observed image intensity.

This is an ambiguity which cannot be resolved locally, and the solution has to be terminated at this point. Under certain lighting conditions the image is divided into regions inside each of which we can find a solution. The regions are separated by ambiguity edges, which cannot be crossed without making an arbitrary choice [4].

4.6. Low sun angles

This problem can be avoided entirely if one deals only with pictures taken at low sun angles, where the gradient is a single-valued function of image intensity. This is a good idea in any case, since the accuracy of the reconstruction depends on how accurately one can determine the gradient, which in turn depends on the spacing of the contour lines in gradient-space. If they are close together, this accuracy is high (near a maximum, on the other hand, it is low). It is easy to convince oneself that pictures taken at low sun angle have "better contrast," show the "relief in more detail," and are "easier to interpret." An additional reason for interest in low sun angle images is that the contours of constant intensity near the shadow-line in gradient-space are nearly straight lines even if we are *not* dealing with the special reflectivity function for the lunar material! An early solution to the problem of determining the shape of lunar hills makes use of this fact by integrating along lines perpendicular to the terminator [5].

Working at low sun angles introduces another problem, of course, since shadows are likely to appear. Fortunately, they are easy to deal with since we simply trace the line in the image until we see a lighted area again. Knowing the direction of the source's rays, we easily determine the position of the first lighted point. The integration is then continued from there (Fig. 20). In fact, no special attention to this

FIG. 20. Geometry of grazing ray needed to deal with shadow gaps in solution.

problem is needed since a surface element oriented for grazing incidence of light already has the correct slope. Thus, simply looking up the slope for zero intensity and integrating with this value will do. Some portion of the surface, of course, is not explored because of shadows. If we take one picture just after "sunrise" and one just before "sunset," most of this area is covered.

4.7. Generalization to perspective projection

All along we have assumed orthographic projection—looking at the surface from a great distance with a telephoto lens. In practice, this is an unreasonable assumption for pictures taken by artificial satellites near the surface. The first thing that

changes in the more general case of perspective projection is that the sun-surface-viewer plane is no longer the same for all portions of the surface images. Since it is this plane which determines the integration lines, we expect that these lines are no longer parallel. Instead they all converge on the anti-solar point in the image which corresponds to a direction directly opposite the direction towards the source (Fig. 21).

The next change is that z is no longer constant in the projection equation. So $s' = f(s/z)$. Hence,

$$p' = \frac{dz}{ds} = \frac{f}{z}\frac{dz}{ds'} = \frac{I/E - G}{\sqrt{1-G^2}}.$$

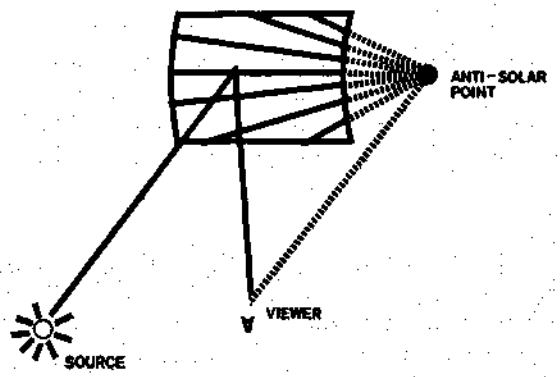


FIG. 21. The base-characteristics converge on the anti-solar point.

We can no longer simply integrate. It is easy, however, to solve the above differential equation for z by separating terms:

$$\log(z) = \frac{1}{f}\int \frac{I/E - G}{\sqrt{1-G^2}}\, ds',$$

and so

$$z(s') = z_0 \exp\left(\frac{1}{f}\int_0^{s'} \frac{I/E - G}{\sqrt{1-G^2}}\, ds'\right).$$

Finally, note that the phase angle g is no longer constant. This has to be taken into account when calculating I/E from the measured image intensity. On the whole, the process is still very simple. The paths of integration are predetermined straight lines in the image radiating from the anti-solar point. At each point we measure the image intensity, determining which value of I/E gives rise to this image intensity. Then we calculate the corresponding slope along the straight line and take a small step. Repeating the process for all lines crossing the image we obtain the surface elevation at all points in the image. The same result can be obtained by a very complex algebraic method [6].

4.8. A note on accuracy

Since image intensities are determined only with rather limited precision, we must expect the calculation of surface coordinates to suffer from errors that accumulate along characteristics. A "sharpening" method that relates adjacent characteristics reduces these errors somewhat [4].

It also appears that an object's shape is better described by the orientations of its surface normals than by the distances from the viewer to points on its surface. In part, this may be because distances to the surface undergo a more complicated transformation when the object is rotated than do surface normal directions. Note that the calculation of surface normals is *not* subject to the cumulative errors mentioned.

Finally, we should point out that the precise determination of the surface is not the main impetus for the development presented here. The understanding of how image intensities are determined by the object, the lighting, and the image forming system is of more importance and may lead to more interesting heuristic methods.

5. The Shape of Surfaces with Arbitrary Reflectance Maps

The simple method developed for lunar topography is inapplicable if the contours of constant intensity in gradient-space are not parallel straight lines. We will still be able to trace along the surface, but the direction we take at each point now depends on the image and changes along the profile. The base characteristics no longer are predetermined straight lines in the image. At each point on a characteristic curve we find that the solution can be continued only in a particular direction. It also appears that we need more information to start a solution and will have to carry along more information to proceed. Reasoning from the gradient-space diagram is augmented here by some algebraic manipulation.

Let $a(p, q)$ be the intensity corresponding to a surface element with a gradient (p, q). Let $b(x, y)$ be the intensity recorded in the image at the point (x, y). Then, for a particular surface element, we must have:

$$a(p, q) = b(x, y).$$

Now suppose we want to proceed in a manner analogous to the method developed earlier by taking a small step $(\mathrm{d}x, \mathrm{d}y)$ in the image. It is clear that we can calculate the corresponding change in z as follows:

$$\mathrm{d}z = z_x \mathrm{d}x + z_y \mathrm{d}_y = p\,\mathrm{d}x + q\,\mathrm{d}y.$$

To do this we need the values of p and q. We have to keep track of the values of the gradient as we integrate along the curve. We can calculate the increments in p and q by:

$$\mathrm{d}p = p_x\,\mathrm{d}x + p_y\,\mathrm{d}y \quad \text{and} \quad \mathrm{d}q = q_x\,\mathrm{d}x + q_y\,\mathrm{d}y.$$

At first, we appear to be getting into more difficulty, since now we need to know $p_x, p_y = q_x$ and q_y. In order to determine these unknowns we will differentiate the basic equation $a(p, q) = b(x, y)$ with respect to x and y:

$$a_p p_x + a_q q_x = b_x \quad \text{and} \quad a_p p_y + a_q q_y = b_y.$$

While these equations contain the right unknowns, there are only two equations, not enough to solve for three unknowns. Note, however, that we do not really need the individual values! We are only after the linear combinations $(p_x\,\mathrm{d}x + p_y\,\mathrm{d}y)$ and $(q_x\,\mathrm{d}x + q_y\,\mathrm{d}y)$.

We have to properly choose the direction of the small step $(\mathrm{d}x, \mathrm{d}y)$ to allow the determination of these quantities. There is only one such direction. Let $(\mathrm{d}x, \mathrm{d}y) = (a_p, a_q)\,\mathrm{d}s$, then $(\mathrm{d}p, \mathrm{d}q) = (b_x, b_y)\,\mathrm{d}s$. This is the solution we were after. Summarizing, we have five ordinary differential equations:

$$\dot{x} = a_p,\ \dot{y} = a_q,\ \dot{z} = a_p p + a_q q,\ \dot{p} = b_x \quad \text{and} \quad \dot{q} = b_y.$$

Here the dot denotes differentiation with respect to s, a parameter that varies along the solution curve.

5.1. Interpretation in terms of the gradient-space

As we solve along a particular characteristic curve on the object, we simultaneously trace out a base characteristic in the image and a curve in gradient-space. At each point in the solution we know to which point in the image and to which point in the gradient-space the surface element under consideration corresponds.

The intensity in the real image and in the gradient space image must, of course, be the same. The paths in the two spaces are related in a peculiar manner. The step we take in the image is perpendicular to the contour in *gradient-space* and the step we take in gradient-space is perpendicular to the intensity contour in the *image-plane*. (See Fig. 22.)

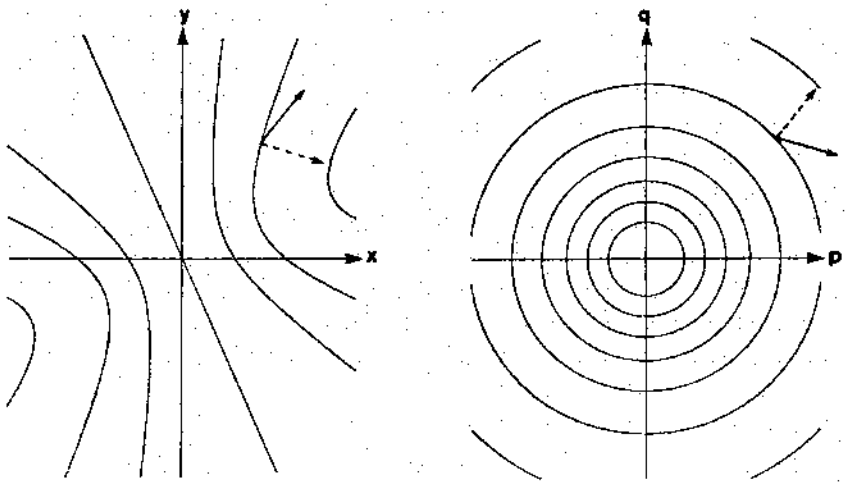


FIG. 22. The solution curve in image-space is along the line of steepest descent in gradient-space and the solution curve in gradient-space is along the line of steepest descent in image-space.

5.2. Generalization to source and viewer near the object

The last solution method, while correct for arbitrary reflectivity functions, still assumes orthographic projection and a distant source. This is a good approximation for many practical cases. In order to take into account the effects of the nearness of the source and the viewer, we discard the gradient-space diagram, since it is based on the assumption of constant phase angle. The problem can still be tackled by algebraic manipulation, much as the last solution. It turns out that we are really trying to solve a first order non-linear partial differential equation in two independent variables. The well-known solution involves converting this equation into five ordinary differential equations, quite like the ones we obtain in the last Section [4].

Appendix

Using the reflectance map to determine three-dimensional structure of polyhedral scenes

What follows is a simple numerical example to illustrate the idea that information about surface reflectance can augment the gradient-space diagram and lead to a solution for the orientation of three planes meeting at a vertex. We will assume a lambertian reflectance for the object and assume that the light-source and viewer are far removed from the object, but close to each other. Suppose now that we are given the partial line-drawing as in Fig. 23 showing edges separating three planes

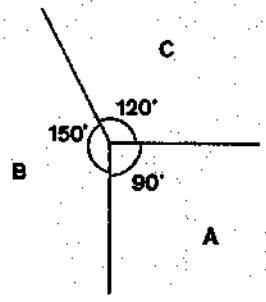


FIG. 23. Image-lines of a tri-hedral vertex. The orientations of the planes *A*, *B* and *C* are to be found.

A, B and C. The gradient-space diagram showing the three points G_A, G_B and G_C corresponding to these three planes will be as in Fig. 24. The scale and position of the indicated triangle are not yet determined. In fact the whole diagram could be flipped around if the scale is negative.

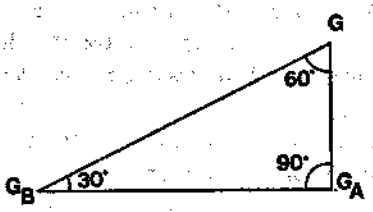


FIG. 24. The gradient-space diagram corresponding to the previous figure. The planes A, B and C map into the points G_A, G_B and G_C. The scale and position of this triangle is as yet undetermined.

Nevertheless, we now have available three linear constraints on the coordinates (p_a, q_a), (p_b, q_b) and (p_c, q_c) of the points G_A, G_B, G_C.

$$p_a = p_c, q_a = q_b, \quad \text{and} \quad (q_c - q_a) = (p_a - p_b)t.$$

Where $t = \tan(30°) = 1/\sqrt{3}$. We are now told that image measurements suggest reflectances of 0.707, 0.807 and 0.577 for the three faces respectively. From the information about the position of the source and viewer, we know that $G = 1$ and that $I = E$. Next, since we are dealing with a lambertian surface we calculate the reflectance from $\phi(I, E, G) = I$, which here equals $E = \cos(e) = 1/\sqrt{1+p^2+q^2}$. We immediately conclude that the surface normals of the three planes are inclined 45°, 36.2° and 54.8° respectively with respect to the view vector.

It also follows that the points G_A, G_B and G_C must lie on circles of radii 1.0, 0.732 and 1.415 in gradient-space, since distance from the origin in gradient space $\sqrt{p^2+q^2}$ equals $\tan(e)$. That is, the points have to lie on the appropriate contours of reflectance in the reflectance map as in Fig. 25.

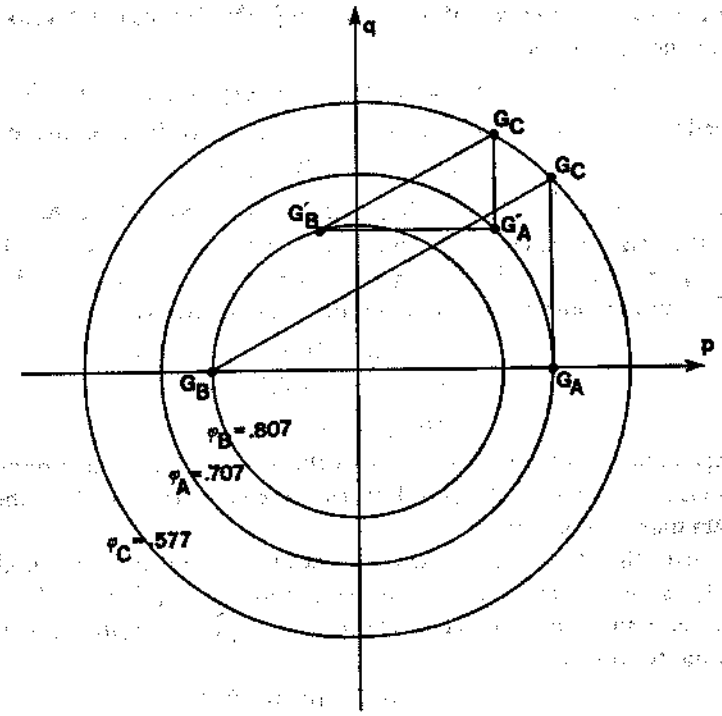


FIG. 25. Contours of constant reflectance corresponding to the reflectances of the three planes. Two solutions are superimposed on the reflectance map.

This gives us three further constraints, unfortunately non-linear ones. Let us call the radii r_a, r_b and r_c respectively, then

$$p_a^2 + q_a^2 = r_a^2, \quad p_b^2 + q_b^2 = r_b^2 \quad \text{and} \quad p_c^2 + q_c^2 = r_c^2.$$

If the source had *not* been near the viewer, these equations would have involved linear terms in p and q as well, since then the contours of equal reflectance would have been conic sections other than circles. In the general case these three equations could be more complicated and in fact it is possible that the reflectance map is only known numerically. Then one will have to proceed iteratively from here. In our simple example however it is possible to solve the three linear equations and the three non-linear ones we have developed directly. As usual one proceeds by judiciously eliminating variables.

Let us use the three linear equations to eliminate the unknowns p_c, q_b and q_c from the three non-linear equations, then,

$$p_a^2+q_a^2 = r_a^2, \quad p_b^2+q_a^2 = r_b^2 \quad \text{and} \quad p_a^2+[q_a+t(p_a-p_b)]^2 = r_c^2.$$

We now have three non-linear equations in three unknowns. First note that

$$(p_a-p_b)(p_a+p_b) = p_a^2-p_b^2 = r_a^2-r_b^2.$$

Then expand the last of the three equations to get:

$$p_a^2+q_a^2+2tq_a(p_a-p_b)+t^2(p_a-p_b)^2 = r_c^2.$$

Now using $p_a^2+q_a^2 = r_a^2$ and the previous equation for (p_a-p_b):

$$[2tq_a+t^2(p_a-p_b)](r_a^2-r_b^2) = (p_a+p_b)(r_c^2-r_a^2).$$

This last question is linear! It is of the form $ap_a+bp_b+cq_a = 0$, where a, b and c can be evaluated and found to be -0.845, -1.155 and 0.536.

The simplest next operation is the elimination of p_a and p_b, using the equations for r_a^2 and r_b^2.

$$-a\sqrt{r_a^2-q_a^2}-b\sqrt{r_b^2-q_a^2} = cq_a.$$

A single equation in a single unknown, at last. Squaring both sides leads to:

$$2ab\sqrt{(r_a^2-q_a^2)(r_b^2-q_a^2)} = (a^2+b^2+c^2)q_a^2-(a^2r_a^2+b^2r_b^2).$$

Letting $e = (a^2+b^2+c^2)/(2ab)$ and $f = (a^2r_a^2+b^2r_b^2)/(2ab)$ and squaring again to get rid of the square-root:

$$(q_a^2)^2(1-d^2)+(q_a^2)[2de-(r_a^2+r_b^2)]+(r_a^2r_b^2-e^2) = 0.$$

This quadratic equation in q_a^2 can be further simplified by evaluating the terms to:

$$(q_a^2)^2-0.5(q_a^2) = 0.0.$$

The solutions are $q_a = 0.0$, -0.7071 and $+0.7071$. Not all of these may be solutions of the original equations, so we will have to check the results. Trying $q_a = 0$, leads to $q_b = 0$, $p_a = \pm 1$, $p_c = \pm 1$, $p_b = \pm 0.732$, and $q_c = 0.577(\pm 1 \pm 0.732)$. Two of these combinations satisfy the original equations.

$$(p_a, q_a) = (1, 0)$$
$$(p_b, q_b) = (-0.732, 0)$$
$$(p_c, q_c) = (1, 1).$$

The other solution is the mirror image of this one, with the same numerical values, but reversed signs. One of these solutions is seen superimposed on the contours of the reflectance map in Fig. 25.

If one tries the other possible set of values for q_a, ± 0.7071 one finds, $p_a = q_b = \pm 0.7071$, $p_b = \pm 0.1895$, $p_c = \pm 0.1895$, and $q_c = \pm 0.7071 + 0.577(\pm 0.7071 \pm 0.1895)$. As before only two combinations satisfy the original equations. One of these is the following:

$$(p_a, q_a) = (0.707, 0.707),$$
$$(p_b, q_b) = (-0.189, 0.707),$$
$$(p_c, q_c) = (0.707, 1.225).$$

The other solution again simply has the signs reversed. One of these solutions is also shown in figure 25. The symmetry of the problem here contributes to the plethora of solutions, more usually one finds but two.

Clearly it would be desirable to avoid this tedious solution of simultaneous non-linear equations. Graphical techniques work and iterative Newton–Raphson techniques are appropriate for computer implementations of this method. For a numerical example see [11].

ACKNOWLEDGMENTS

I wish to thank Blenda Horn and Eva Kampitts for help in the preparation of the paper, Karen Prendergast for the drawings and Kathy Van Sant for an early version of the numerical solution for the truncated-plane mutual-illumination problem. David Marr and Patrick Winston provided much appreciated stimulation and discussion.

REFERENCES

1. Huffman, D. A., Impossible objects as nonsense sentences, *Machine Intelligence* **6**, Meltzer, R., and Michie, D. (Eds.), (Edinburgh University Press, 1971) 295–323.
2. Mackworth, A. K., Interpreting pictures of polyhedral scenes, *Artificial Intelligence* **4** (1973), 121–137.
3. Huffman, D. A., Curvature and creases: a primer on paper, *Proc. Conf. Computer Graphics, Pattern Recognition and Data Structures* (May 1975) 360–370.
4. Horn, B. K. P., Obtaining shape from shading information, in: Winston, P. H., (Ed.), *The Psychology of Machine Vision* (McGraw-Hill, NY, 1975) 115–155.
5. Van Diggelen, J., A photometric investigation of the slopes and heights of the ranges of hills in the maria of the moon, *Bull. Astron. Inst. Netherlands* **11** (1951) 283–289.
6. Rindfleisch, T., Photometric method for lunar topography, *Photogrammetric Eng.* **32** (1966) 262–276.
7. Herskovits, A. and Binford, T. O., On boundary detection, MIT Artificial Intelligence Memo 183 (July 1970) 19, 55, 56.
8. Phong, Bui Tuong, Illumination for Computer Generated Pictures, *CACM* **18** (1975) 311–317.
9. Hilbert, D. and Cohn-Vossen, S., *Geometry and the Imagination* (Chelsea Publishers, New York, 1952).
10. Hildebrand, F. D., *Methods of Applied Mathematics* (Prentice-Hall, New Jersey, 1952) 222–294.
11. Horn, B. K. P., Image intensity understanding, MIT Artificial Intelligence Memo 335 (August 1975).

Received January 1976; final version received August 1976

Machine Vision and Applications (1988) 1:127–152

Machine Vision and Applications

Active, Optical Range Imaging Sensors

Paul J. Besl
Computer Science Department, General Motors Research Laboratories, Warren, Michigan 48090-9055 USA

Abstract: Active, optical range imaging sensors collect three-dimensional coordinate data from object surfaces and can be useful in a wide variety of automation applications, including shape acquisition, bin picking, assembly, inspection, gaging, robot navigation, medical diagnosis, and cartography. They are unique imaging devices in that the image data points explicitly represent scene surface geometry in a sampled form. At least six different optical principles have been used to actively obtain range images: (1) radar, (2) triangulation, (3) moire, (4) holographic interferometry, (5) focusing, and (6) diffraction. In this survey, the relative capabilities of different sensors and sensing methods are evaluated using a figure of merit based on range accuracy, depth of field, and image acquisition time.

Key Words: range image, depth map, optical measurement, laser radar, active triangulation

1. Introduction

Range-imaging sensors collect large amounts of three-dimensional (3-D) coordinate data from visible surfaces in a scene and can be used in a wide variety of automation applications, including object shape acquisition, bin picking, robotic assembly, inspection, gaging, mobile robot navigation, automated cartography, and medical diagnosis (biostereometrics). They are unique imaging devices in that the image data points explicitly represent scene surface geometry as sampled points. The inherent problems of interpreting 3-D structure in other types of imagery are not encountered in range imagery although most low-level problems, such as filtering, segmentation, and edge detection, remain.

Most active optical techniques for obtaining range images are based on one of six principles: (1) radar, (2) triangulation, (3) moire, (4) holographic interferometry, (5) lens focus, and (6) Fresnel diffraction. This paper addresses each fundamental category by discussing example sensors from that class. To make comparisons between different sensors and sensing techniques, a performance figure of merit is defined and computed for each representative sensor if information was available. This measure combines image acquisition speed, depth of field, and range accuracy into a single number. Other application-specific factors, such as sensor cost, field of view, and standoff distance are not compared.

No claims are made regarding the completeness of this survey, and the inclusion of commercial sensors should not be interpreted in any way as an endorsement of a vendor's product. Moreover, if the figure of merit ranks one sensor better than another, this does not necessarily mean that it is better than the other for any given application.

Jarvis (1983b) wrote a survey of range-imaging methods that has served as a classic reference in range imaging for computer vision researchers. An earlier survey was done by Kanade and Asada (1981). Strand (1983) covered range imaging techniques from an optical engineering viewpoint. Several other surveys have appeared since then (Kak 1985, Nitzan et al. 1986, Svetkoff 1986, Wagner 1987). The goal of this survey is different from previous work in that it provides a simple example methodology for quantitative performance comparisons between different sensing methods which may assist system engineers in performing evaluations. In addition, the state of the art in range imaging advanced rapidly in the past few years and is not adequately documented elsewhere.

This survey is structured as follows. Definitions of range images and range-imaging sensors are given first. Different forms of range images and generic viewing constraints and motion requirements are discussed next followed by an introduction to sensor performance parameters, which are then used to define a figure of merit. The main body sequentially addresses each fundamental ranging method. The figure of merit is computed for each sensor if possible. The conclusion consists of a sen-

sor comparison section and a final summary. An introduction to laser eye safety is included in the appendix. This paper is an abridged version of Besl (1988), which was derived from sections of Besl (1987). Tutorial material on range-imaging techniques may be found in both as well as in the references.

2. Preliminaries

A *range-imaging sensor* is any combination of hardware and software capable of producing a *range image* of a real-world scene under appropriate operating conditions. A *range image* is a large collection of *distance measurements* from a known reference coordinate system to *surface points* on object(s) in a *scene*. If scenes are defined as collections of physical objects and if each *object* is defined by its mass density function, then surface points are defined as the 3-D points in the half-density level set of each object's normalized mass-density function as in Koenderink and VanDoorn (1986). Range images are known by many other names depending on context: range map, depth map, depth image, range picture, rangepic, 3-D image, 2.5-D image, digital terrain map (DTM), topographic map, 2.5-D primal sketch, surface profiles, *xyz* point list, contour map, and surface height map.

If the distance measurements in a range image are listed relative to three orthogonal coordinate axes, the range image is in *xyz* form. If the distance measurements indicate range along 3-D direction vectors indexed by two integers (i, j), the range image is in r_{ij} form. Any range image in r_{ij} form can be converted directly to *xyz* form, but the converse is not true. Since no ordering of points is required in the *xyz* form, this is the more general form, but it can be more difficult to process than the r_{ij} form. If the sampling intervals are consistent in the x- and y-directions of an *xyz* range image, it can be represented in the form of a large matrix of scaled, quantized range values r_{ij} where the corresponding x, y, z coordinates are determined implicitly by the row and column position in the matrix and the range value. The term "image" is used because any r_{ij} range image can be displayed on a video monitor, and it is identical in form to a digitized video image from a TV camera. The only difference is that pixel values represent distance in a range image whereas they represent irradiance (brightness) in a video image.

The term "large" in the definition above is relative, but for this survey, a range image must specify more than 100 (x, y, z) sample points. In Figure 1, the 20 × 20 matrix of heights of surface points above a plane is a small range image. If r_{ij} is the pixel value at the ith row and the jth column of the matrix, then the 3-D coordinates would be given as

$$x = a_x + s_x i \qquad y = a_y + s_y j \qquad z = a_z + s_z r_{ij} \quad (1)$$

where the s_x, s_y, s_z values are scale factors and the a_x, a_y, a_z values are coordinate offsets. This matrix of numbers is plotted as a surface viewed obliquely in Figure 2, interpolated and plotted as a contour map in Figure 3, and displayed as a black and white image in Figure 4. Each representation is an equally valid way to look at the data.

The affine transformation in equation (1) is appropriate for *orthographic* r_{ij} range images where depths are measured along parallel rays orthogonal to the image plane. Nonaffine transformations of (i, j, r_{ij}) coordinates to Cartesian (x, y, z) coordinates are more common in active optical range sensors. In the spherical coordinate system shown in Figure

```
171 160 163 163 166 166 168 166 168 166 163 160 163 163 160 163 166 163 166 163
168 166 166 163 166 163 168 166 166 166 163 163 166 163 166 163 166 160 163 163
163 168 166 166 166 163 160 166 166 171 166 168 168 166 160 163 166 160 160 166
166 163 166 166 163 163 160 163 179 174 185 177 185 179 212 196 185 204 196 185
163 166 160 166 163 163 166 160 174 168 168 182 185 190 201 196 199 182 196 199
166 163 163 163 168 160 163 166 166 163 166 177 190 188 199 188 190 196 193 185
163 166 166 157 160 160 160 171 160 168 166 182 199 199 199 193 199 188 193 193
160 160 160 166 157 160 168 166 166 163 163 182 201 199 190 188 190 190 193 193
163 166 157 163 160 157 160 177 166 160 171 201 215 199 196 201 190 190 188 188
166 160 160 163 160 163 160 166 166 163 163 204 207 207 190 185 193 190 196 196
157 166 163 160 157 157 166 166 166 163 177 188 201 199 196 196 201 182 210 196
157 157 166 157 160 157 163 171 163 157 166 204 185 196 193 188 198 188 193 201
157 160 166 166 157 157 166 166 163 166 166 190 201 201 196 188 190 193 185 193
157 166 160 160 157 157 163 157 157 160 157 182 204 190 185 190 190 188 185 188
167 157 157 160 157 157 152 166 160 163 166 193 196 193 199 190 190 185 190 166
166 157 160 160 160 152 166 152 163 152 168 171 212 212 193 190 168 182 188 185
162 157 166 166 152 166 149 163 160 166 157 166 210 210 212 215 210 185 204 193
166 166 157 152 152 166 166 171 174 166 171 188 188 199 188 204 188 185 215 207
166 157 152 157 149 157 157 168 179 204 182 221 174 193 182 179 212 188 201 182
166 166 166 166 152 149 148 174 188 193 168 185 168 179 171 190 190 193 190 179
```

Figure 1. 20 × 20 matrix of range measurements: r_{ij} form of range image.

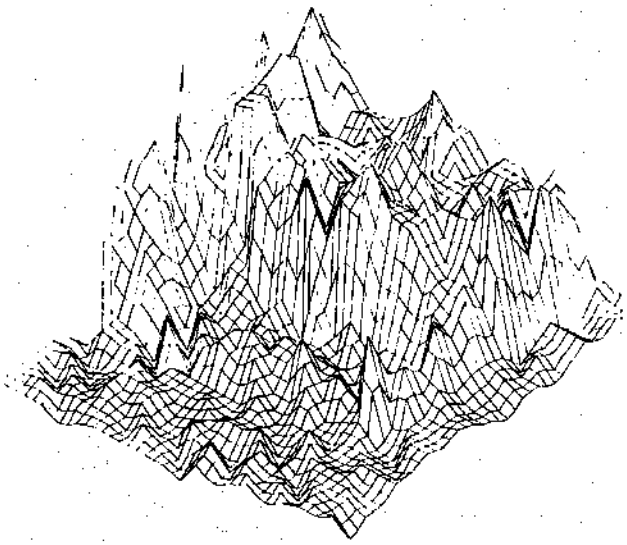


Figure 2. Surface plot of range image in Figure 1.

5, the (i, j) indices correspond to elevation (latitude) angles and azimuth (longitude) angles respectively. The spherical to Cartesian transformation is

$$\begin{aligned} x &= a_x + s_r r_{ij} \cos(is_\phi)\sin(js_\theta) \\ y &= a_y + s_r r_{ij} \sin(is_\phi) \\ z &= a_z + s_r r_{ij} \cos(is_\phi)\cos(js_\theta) \end{aligned} \quad (2)$$

where the s_r, s_ϕ, s_θ values are the scale factors in range, elevation, and azimuth and the a_x, a_y, a_z values are again the offsets. The "orthogonal-axis" angular coordinate system, also shown in Figure 5, uses an "alternate elevation angle" ψ with the spherical azimuth definition θ. The transformation to Cartesian coordinates is

$$\begin{aligned} x &= a_x + s_r r_{ij}\, tan(js_\theta)/\sqrt{1 + \tan^2(is_\theta) + \tan^2(js_\psi)} \\ y &= a_y + s_r r_{ij} \tan(is_\psi)/\sqrt{1 + \tan^2(is_\theta) + \tan^2(js_\psi)} \\ z &= a_z + s_r r_{ij}/\sqrt{1 + \tan^2(is_\theta) + \tan^2(js_\psi)}. \end{aligned} \quad (3)$$

The alternate elevation angle ψ depends only on y and z whereas ϕ depends on x, y, and z. The differences in (x, y, z) for equations (2) and (3) for the same values of azimuth and elevation are less than 4% in x and z and less than 11% in y, even when both angles are as large as ± 30 degrees.

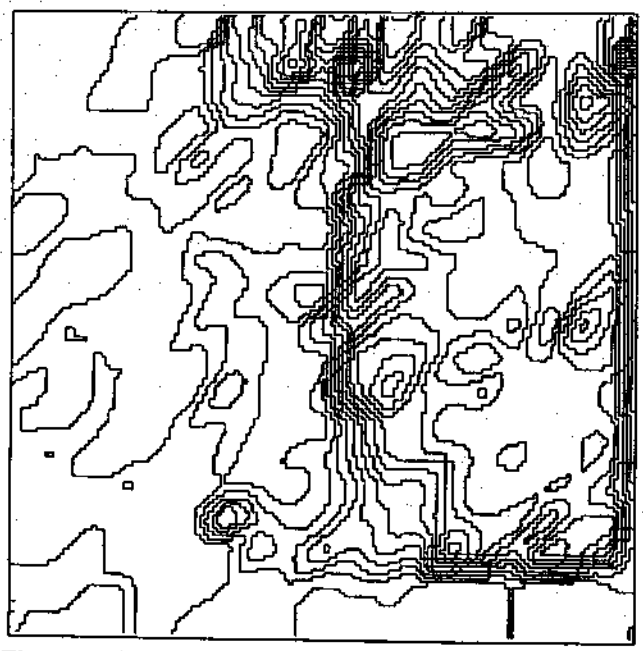


Figure 3. Contour plot of range image in Figure 1.

2.1 Viewing Constraints and Motion Requirements

The first question in range imaging requirements is *viewing constraints*. Is a single view sufficient, or are multiple views of a scene necessary for the given application? What types of sensors are compatible with these needs? For example, a mobile robot can acquire data from its on-board sensors only at its current location. An automated modeling system may acquire multiple views of an object with many sensors located at different viewpoints. Four basic types of range sensors are distinguished based on the viewing constraints, scanning mechanisms, and object movement possibilities:

1. A *Point Sensor* measures the distance to a single visible surface point from a single viewpoint along a single ray. A point sensor can create a range image if (1) the scene object(s) can be physically moved in two directions in front of the point-ranging sensor, (2) if the point-ranging sensor can be scanned in two directions over the scene, or (3) the scene object(s) are stepped in

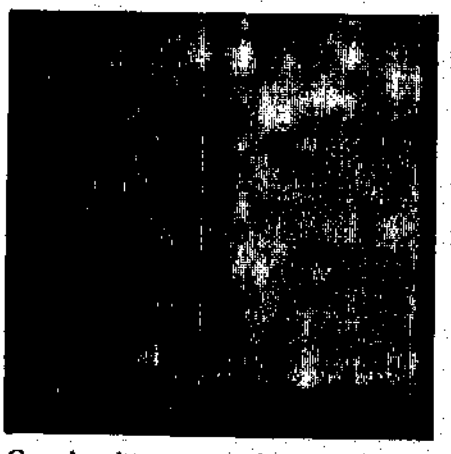


Figure 4. Gray level representation of range image in Figure 1.

one direction while the point sensor is scanned in the other direction.

2. A *Line* or *Circle Sensor* measures the distance to visible surface points that lie in a single 3-D plane or cone that contains the single viewpoint or viewing direction. A line or circle sensor can create a range image if (1) the scene object(s) can be moved orthogonal to the sensing plane or cone or (2) the line or circle sensor can be scanned over the scene in the orthogonal direction.
3. A *Field of View Sensor* measures the distance to many visible surface points that lie within a given field of view relative to a single viewpoint or viewing direction. This type of sensor creates a range image directly. No scene motion is required.
4. A *Multiple View Sensor System* locates surface points relative to more than one viewpoint or viewing direction because all surface points of interest are not visible or cannot be adequately measured from a single viewpoint or viewing direction. Scene motion is not required.

These sensor types form a natural hierarchy: a point sensor may be scanned (with respect to one sensor axis) to create a line or circle sensor, and a line or circle sensor may be scanned (with respect to the orthogonal sensor axis) to create a field of view sensor. Any combination of point, line/circle, and field of view sensors can be used to create a multiple view sensor by (1) rotating and/or translating the scene in front of the sensor(s); (2) maneuvering the sensor(s) around the scene with a robot; (3) using multiple sensors in different locations to capture the appropriate views; or any combination of the above.

Accurate sensor and/or scene object positioning is achieved via commercially available translation stages, $xy(z)$-tables, and $xy\theta$ tables (translation repeatability in submicron range, angular repeatability in microradians or arc-seconds). Such methods are preferred to mirror scanning methods for high precision applications because these mechanisms can be controlled better than scanning mirrors. Controlled 3-D motion of sensor(s) and/or object(s) via gantry, slider, and/or revolute joint robot arms is also possible, but is generally much more expensive than table motion for the same accuracy. Scanning motion internal to sensor housings is usually rotational (using a rotating mirror), but may also be translational (using a precision translation stage). Optical scanning of lasers has been achieved via (1) motor-driven rotating polygon mirrors, (2) galvanometer-driven flat mirrors, (3) acoustooptic (AO) modulators, (4) rotating holographic scanners, or (5) stepper-motor-driven mirrors (Gottlieb 1983, Marshall 1985). However, AO modulators and holographic scanners significantly attenuate laser power, and AO modulators have a narrow angular field of view ($\approx 10° \times 10°$), making them less desirable for many applications.

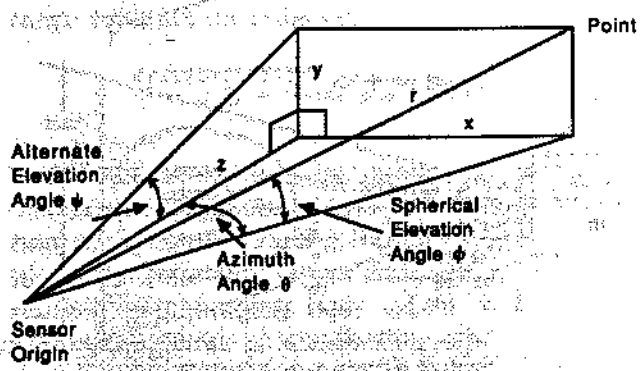


Figure 5. Cartesian, spherical, and orthogonal-axis coordinates.

2.2 Sensor Performance Parameters

Any measuring device is characterized by its measurement resolution or precision, repeatability, and accuracy. The following definitions are adopted here. *Range resolution* or *range precision* is the smallest change in range that a sensor can report. *Range repeatability* refers to statistical variations as a sensor makes repeated measurements of the exact same distance. *Range accuracy* refers to statistical variations as a sensor makes repeated measurements of a known *true value*. Accuracy should indicate the largest expected deviation of a measurement from the true value under normal operating conditions. Since range sensors can improve accuracy by averaging multiple measurements, accuracy should be quoted with measurement time. For our comparisons, a range sensor is characterized by its accuracy over a given measurement interval (the depth of field) and the measurement time. If a sensor has good repeatability, we assume that it is also calibrated to be accurate. Loss of calibration over time (drift) is a big problem for poorly engineered sensors but is not addressed here.

A range-imaging sensor measures point positions (x, y, z) within a specified accuracy or error tolerance. The method of specifying accuracy varies in different applications, but an accuracy specification should include one or more of the following for each 3-D direction given N observations: (1) the mean absolute error (MAE) $(\pm\delta_x, \pm\delta_y, \pm\delta_z)$ where $\delta_x = (1/N)\Sigma|x_i - \mu_x|$ and $\mu_x = (1/N)\Sigma x_i$ (or μ_x = median (x_i)); (2) RMS (root-mean-square) error $(\pm\sigma_x, \pm\sigma_y, \pm\sigma_z)$ where $\sigma_x^2 = (N-1)^{-1}\Sigma(x_i - \mu_x)^2$ and $\mu_x =$

$(1/N)\Sigma x_i$; or (3) maximum error ($\pm\epsilon_x$, $\pm\epsilon_y$, $\pm\epsilon_z$) where $\epsilon_x = \max_i|x_i - \mu_x|$. (Regardless of the measurement error probability distribution, $\delta \leq \sigma \leq \epsilon$ for each direction.) Some specify range accuracy (σ) with $\pm\sigma$ for RMS error as above; others specify $\pm 3\sigma$; others specify the positive width of the normal distribution as 2σ; and others do not say, if any of the above. Whatever specification is used, the sensor should ideally meet the specified error tolerance for any measured point within the working volume V of size $L_x \times L_y \times L_z$.

The foregoing parameters specify the spatial properties of the sensor. The pixel dwell time T is the time required for a single pixel range measurement. For points acquired *sequentially*, the total time to take N_p points in an image frame is N_pT, the frame rate is $1/N_pT$ frames/second, and the pixel data rate is $1/T$ pixels/second. If all points are acquired in *parallel* in time T, the frame rate is $1/T$ and the pixel rate N_p/T.

A system figure of merit maps several system parameters to a single number for comparing different systems. An application independent performance figure of merit M is defined to measure the spatial and temporal performance of range imaging sensors. For xyz-form range imaging, the figure of merit M is defined as

$$M = \frac{1}{\sqrt{T}}\left(\frac{L_xL_yL_z}{\sigma_x\sigma_y\sigma_z}\right)^{1/3}. \tag{4}$$

For r_{ij}-form range imaging, there is usually very little relative uncertainty in the direction of a ray specified by the interger i and j indices compared to the uncertainty in the measured range r. Thus, the uncertainty in the resulting x, y, z coordinates is dominated by the uncertainty in the r_{ij} values. The working volume is a portion of a pyramid cut out by spherical surfaces with the minimum and maximum distance radii. The figure of merit for r_{ij}-form range-imaging sensors is given by the simpler expression

$$M \approx \frac{L_r}{\sigma_r\sqrt{T}} \tag{5}$$

where L_r is the *depth of field* and σ_r is the *RMS range accuracy*. Both quantities are defined along rays emanating from the sensor. The factors of standoff distance, angular field of view, and field of view are other important parameters for range sensors that do not enter into the figure of merit calculations directly, but should be considered for each application. These parameters are shown in Figure 6.

The dimensions of M may be thought of roughly as the amount of good quality range data per second, and of course, the higher the number the better, other things being equal. A doubling of the depth-of-field to range-accuracy ratio is reflected by a doubling of the figure of merit. However, a quadrupling of image acquisition speed is required to double the figure of merit. This expresses a bias for accurate measurement over fast measurement, but also maintains an invariant figure of merit under internal sensor averaging changes. Suppose a system does internal averaging of normally distributed measurement errors during the pixel dwell time T. If T is quartered, the σ-value should only double. If the square root of time T were not used, the figure of merit would double as the data became noisier and the sensor got faster. This was considered undesirable.

The figures of merit quoted in this survey should be taken as examples of the rough order of magnitude of sensor performance, not exact numbers. First, it is difficult to know the actual accuracy for a given application without testing a sensor on typical scenes. Second, it is difficult to know whether quoted accuracy means 1, 2, 3, or 4σ or something else. Third, even if the quoted figure is a valid test result, the surface reflectance, absorption, and transmission properties for the test are not always stated. Sensor performance is often quoted under the most favorable conditions. Fourth, several sensors, especially sensors from conservative commercial companies, are underrated because of conservative accuracy figures. They know about the vast difference between measurements in the lab and in the customer's plant and about customer disappointment. In fact, some sensors are conservatively rated 10 to 20 times less accurate than they would be in the lab. Finally, only resolution is given for some sensors and accuracy had to be estimated.

The sensor cost C can be combined with the performance figure of merit to create a cost-weighted

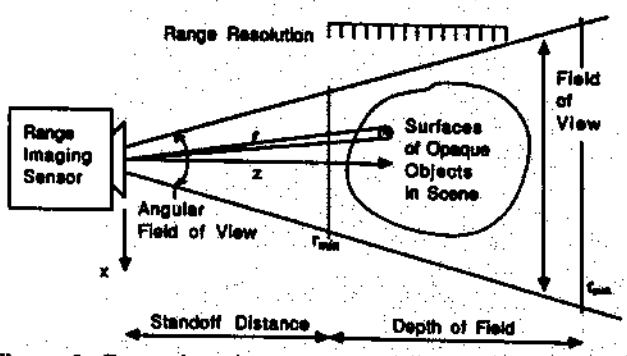


Figure 6. Range imaging sensor with angular scan.

figure of merit $M' = M/C$ where the dimensions are roughly range data per second per unit cost. Cost estimates are not included here because actual costs can vary significantly from year to year depending upon technical developments and market forces, not to mention customized features that are often needed for applications. Cost estimates were also not available for many sensors.

It is likely that these figures of merit M and M' may place no importance on factors that dominate decisions for a particular application. The figures of merit given here are application independent. No figure of merit can represent all factors for all applications. For example, some triangulation or moire range sensors with large source/detector separations may have a significant "missing parts" problem (shadowing problem) for certain applications and not for others. Figures of merit cannot easily reflect this limitation.

Neither can the "scene materials" problem be easily factored into a figure of merit. There are materials in many scenes that almost completely reflect, absorb, or transmit optical radiation. For example, mirrors and shiny metal or plastic surfaces reflect light, black paint may absorb infrared, and glass is transparent. These materials cause scene geometry interpretation problems for optical sensors. Hence, the physical/chemical composition of matter in a scene determines the quality and the meaning of range values. Even though optical range sensors are designed for determining scene geometry directly, a priori information about the optical properties of scene materials is needed for accurate interpretation.

3. Imaging Radars

Bats (Griffin 1958) and porpoises (Kellogg 1961) are equipped by nature with ultrasonic "radars." Electromagnetic radar dates back to 1903 when Hulsmeyer (1904) experimented with the detection of radio waves reflected from ships. The basic time/range equation for radars is

$$v\tau = 2r = \text{round-trip distance} \tag{6}$$

where v is the speed of signal propagation, r is the distance to a reflecting object, and τ is the transit time of the signal traveling from the radar transmitter to the reflecting object and back to the radar receiver. For imaging laser radars, the unknown scene parameters at a reflecting point are the (1) range r, (2) the surface reflection coefficient (albedo) ρ, and (3) the angle $\theta = \cos^{-1}(\hat{n} \cdot \hat{l})$ between the visible surface normal $\hat{n}$ and the direction $\hat{l}$ of the radar beam. Ignoring atmospheric attenuation, all other relevant physical parameters can be lumped into a single function $K(t)$ that depends only on the radar transceiver hardware. The received power $P(t)$ is

$$P(t, \theta, \rho, r) = K(t - \tau)\rho \cos\theta / r^2 \tag{7}$$

This laser radar equation tells us that if 10 bits of range resolution are required on surfaces that may tilt away from the sight line by as much as 60 deg, and if surface reflection coefficients from 1 to 0.002 are possible on scene surfaces, then a radar receiver with a dynamic range of 90 dB is required.

3.1 Time of Flight, Pulse Detection

In this section, several pulse detection imaging laser radars are mentioned. A figure of merit M is assigned to each sensor.

Lewis and Johnston at JPL built an imaging laser radar beginning in 1972 for the Mars rover (Lewis and Johnston 1977). Their best range resolution was 20 mm over a 3-m depth of field and the maximum data rate possible was 100 points per second. It took about 40 seconds to obtain 64 × 64 range images (M = 1520).

Jarvis (1983a) built a similar sensor capable of acquiring a 64 × 64 range image with ±2.5 mm range resolution over a 4 m field of view in 40 s (M = 16,160).

Heikkinen et al. (1986) and Ahola et al. (1985) developed a pulsed time-of-flight range sensor with a depth of field of 1.5 m at a standoff of 2.5 m. The range resolution is about 20 mm at its maximum data rate (10,000 points/s) at a range of 3.5 m (M = 7500).

Ross (1978) patented a novel pulsed, time-of-flight imaging laser radar concept that uses several fast camera shutters instead of mechanical scanning. For a range sensor with 30 cm resolution over a 75-m depth of field, the least significant range-bit image is determined by a 2-ns shutter (the fastest shutter required). Assuming a conservative frame rate of 15 Hz and eight, 512 × 512 cameras, M = 500,000 if constructed.

An imaging laser radar is commercially available for airborne hydrographic surveying (Banic et al. 1987). The system can measure water depths down to 40 m with an accuracy of 0.3 m from an aerial standoff of 500 m. Two hundred scan lines were acquired covering 2000 km^2 with two million

"soundings" in 30 h (M = 596). This number is low because *application specific capabilities* (e.g., standoff) are not included.

3.2 Amplitude Modulation

Rather than sending out a short pulse, waiting for an echo, and measuring transit time, a laser beam can be amplitude-modulated by varying the drive current of a laser diode at a frequency $f_{AM} = c/\lambda_{AM}$. An electronic phase detector measures the phase difference $\Delta\phi$ (in radians) between the transmitted signal and the received signal to get the range: $r(\Delta\phi) = c\Delta\phi/4\pi f_{AM} = \lambda_{AM}\Delta\phi/4\pi$. Since relative phase differences are only determined modulo 2π, the range to a point is only determined within a range ambiguity interval r_{ambig}. In the absence of any ambiguity-resolving mechanisms, the depth of field of an AM laser radar is the ambiguity interval: $L_r = r_{ambig} = c/2f_{AM} = \lambda_{AM}/2$ which is divided into $2^{N_{bits}}$ range levels where N_{bits} is the number of bits of quantization at the output of the phase detector. Finer depth resolution and smaller ambiguity intervals result from using higher modulating frequencies.

The ambiguity interval problem in AM CW radars can be resolved either via software or more hardware. If the imaged scene is limited in surface gradient relative to the sensor, it is possible in software to unwrap phase ambiguities because the phase gradient will always exceed the surface gradient limit at phase wraparound pixels. This type of processing is done routinely in moire sensors (see Halioua and Srinivasan 1987). In hardware, a system could use multiple modulation frequencies simultaneously. In a simple approach, each range ambiguity is resolved by checking against lower modulation frequency measurements. Other methods are possible, but none are commercially available at the current time.

Nitzan et al. (1977) built one of the first nonmilitary AM imaging laser radars. It created high-quality registered range and intensity images. With a 40-dB signal-to-noise ratio (SNR), a range accuracy of 4 cm in an ambiguity interval of 16.6 m was obtained. With a 67 dB SNR, the accuracy improved to 2 mm. The pixel dwell time was variable: 500 ms per pixel dwell times were common and more than 2 h was needed for a full 128 × 128 image (M = 3770 at 67 dB). The system insured image quality by averaging the received signal until the SNR was high enough.

The Environmental Research Institute of Michigan (ERIM) developed three AM imaging laser radars: (1) the Adaptive Suspension Vehicle (ASV) system, (2) the Autonomous Land Vehicle (ALV) system, and (3) the Intelligent Task Automation (ITA) system. Zuk and Dell'Eva (1983) described the ASV sensor. The range accuracy is about 61 mm over 9.75 m at a frame rate of two 128 × 128 images per second (M = 28,930). The ALV sensor generates two 256 × 64 image frames per second. The ambiguity interval was increased to 19.5 m, but M = 28930 is identical to the ASV sensor since pixel dwell time and depth of field to range accuracy ratios stayed the same. The new ERIM navigation sensor (Sampson 1987) uses lasers with three different frequencies and has 2-cm range resolution (M = 353,000 assuming depth of field is doubled). The ERIM ITA sensor is programmable for up to 512 × 512 range images (Svetkoff et al. 1984). The depth of field can change from 150 mm to 900 mm. As an inspection sensor, the laser diode is modulated at 720 MHz. The sensor then has a range accuracy of 100 μ at a standoff of 230 mm in a 76-mm × 76-mm field of view over a depth of field of 200 mm. The latest system of this type claims a 100-kHz pixel rate (M = 632,500).

A commercially available AM imaging laser radar is built by Odetics (Binger and Harris 1987). Their sensor has a 9.4-m ambiguity interval with 9-bit range resolution of 18 mm per depth level. The pixel dwell time is 32 μsec (M = 71,720). This sensor features an *auto-calibration feature* that calibrates the system *every frame* avoiding thermal drift problems encountered in other sensors of this type. It is currently the smallest (9 × 9 × 9 in.), lightest weight (33 lbs.), and least power hungry (42 W) sensor in its class. Class I CDRH eye safety requirements (see the appendix) are met except within a 0.4 m radius of the aperture.

Another commercially available AM imaging laser radar is built by Boulder Electro-Optics (1986). The ambiguity interval is 43 m with 8-bit resolution (about 170 mm). The frame rate was 1.4 256 × 256 frames/sec (M = 27,360).

Perceptron (1987) reports they are developing an AM imaging laser radar with a 360-kHz data rate, a 1.87-m ambiguity interval, a 3-m standoff, and 0.45-mm (12-bit) range resolution (M = 153,600 assuming 8-bit accuracy).

Cathey and Davis (1986) designed a system using multiple laser diodes, one for each pixel, to avoid scanning. They obtained a 15-cm range accuracy at a range of 13 m with a 2-diode system. For N^2 laser diodes fired four times a second, $M = 512N$. If the sensor cost is dominated by N^2 laser diode cost, the cost-weighted figure of merit M' would decrease as $1/N$. A full imaging system has not been built.

Miller and Wagner (1987) built an AM radar unit using a modulated infrared LED. The system scans 360 deg in azimuth, digitizing about 1000 points in a second. The depth of field is about 6 m with a range accuracy of about 25 mm (M = 7590). This system is very inexpensive to build and is designed for mobile robot navigation.

The Perkin-Elmer imaging airborne laser radar (Keyes 1986) scans 2790 pixels per scan line in 2 ms (M = 302,360 assuming 8-bit range accuracy). Aircraft motion provides the necessary scanning motion in the flight direction of the aircraft.

Wang et al. (1984) and Terras (1986) discussed the imaging laser radar developed at General Dynamics. The 12 × 12-deg angular field of view is scanned by dual galvanometers. It ranges out to 350 m, but the ambiguity interval is 10 m yielding lots of phase transition stripes in uncorrected range images.

Other work in AM imaging laser radars has been done at Hughes Aircraft, MIT Lincoln Labs (Quist et al. 1978), Raytheon (Jelalian and McManus 1977), as well as United Technologies and other defense contractors.

3.3 Frequency Modulation, Heterodyne Detection

The optical frequency of a laser diode can also be tuned thermally by modulating the laser diode drive current (Dandridge 1982). If the transmitted optical frequency is repetitively swept linearly between $\nu \pm \Delta\nu/2$ to create a total frequency deviation of $\Delta\nu$ during the period $1/f_m$ (f_m is the linear sweep modulation frequency), the reflected return signal can be mixed coherently with a reference signal at the detector (Teich 1968) to create a beat frequency f_b signal that depends on the range to the object r (Skolnick 1962). This detection process is known as FM coherent heterodyne detection. Range is proportional to the beat frequency in an FM CW radar: $r(f_b) = cf_b/4f_m\Delta\nu$. One method for measuring the beat frequency is counting the number of zero-crossings N_b of the beat signal during a ramp of the linear sweep frequency modulation. This zero-crossing count must satisfy the relationship $2N_b = \lfloor f_b/f_m \rfloor$ which yields the range equation $r(N_b) = cN_b/2\Delta\nu$. The range values in this method are determined to within $\delta r = \pm c/4\Delta\nu$ since N_b must be an integer. The maximum range should satisfy the constraint that $r_{max} \ll c/f_m$. Since it is difficult to ensure the exact optical frequency deviation $\Delta\nu$ of a laser diode, it is possible to measure range indirectly by comparing the N_b value with a known reference count N_{ref} for an accurately known reference distance r_{ref} using the relationship $r(N_b) = N_b r_{ref}/N_{ref}$. Hersman et al. (1987) reported results for two commercially available FM imaging laser radars: a vision system and a metrology system (Digital Optronics 1986). The vision system measures a 1-m depth of field with 8-bit resolution at four 256 × 256 frames/second (M = 3770 using a quoted value of 12 mm for RMS depth accuracy after averaging 128 frames in 32 s). A new receiver is being developed to obtain similar performance in 0.25 s. The metrology system measures to an accuracy of 50 μ in 0.1 s over a depth of field of 2.5 m (M = 30,430). Better performance is expected when electronically tunable laser diodes are available.

Beheim and Fritsch (1986) reported results with an in-house sensor. Points were acquired at a rate of 29.3/s. The range accuracy varied with target to source distance. From 50 to 500 mm, the range accuracy was 2.7 mm; from 600 to 1000 mm, σ_z = 7.4 mm; and from 1100 to 1500 mm, σ_z = 15 mm (approximately M = 1080).

4. Active Triangulation

Triangulation based on the law of sines is certainly the oldest method for measuring range to remote points and is also the most common. A simple geometry for an active triangulation system is shown in Figure 7. A single camera is aligned along the z-axis with the center of the lens located at (0, 0, 0). At a baseline distance b to the left of the camera (along the negative x-axis) is a light projector sending out a beam or plane of light at a variable angle θ relative to the x-axis baseline. The point (x, y, z) is projected into the digitized image at the pixel (u, v) so $uz = xf$ and $vz = yf$ by similar triangles where f is the focal length of the camera in pixels. The measured quantities (u, v, θ) are used to compute the (x, y, z) coordinates:

$$[x\ y\ z] = \frac{b}{f\cot\theta - u}[u\ v\ f] \tag{8}$$

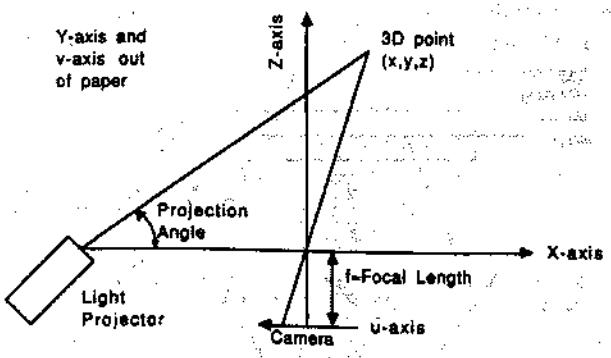


Figure 7. Camera-centered active triangulation geometry.

4.1 Structured Light: Point

It is commonly believed that a large baseline distance b separating the light source and the detector is necessary for accurate ranging. However, for any fixed focal length f and baseline distance b, the range resolution of a triangulation system is only limited by the ability to accurately measure the angle θ and the horizontal position u.

Rioux (1984) has patented a synchronized scanner concept for active triangulation in which the horizontal position detector and the beam projector are *both* scanned. The angle θ is coupled with the u measurement yielding high-range resolution with a small baseline by making more efficient use of the finite resolution of the horizontal position detector. The basic concept is that if one uses the available resolution to measure differences from the mean rather than absolute quantities, the effective resolution can be much greater. As shown in Figure 8, the beam leaves the source, hits the mirror currently rotated at a position θ, bounces off a fixed (source) mirror and impinges on an object surface. The illuminated bright spot is viewed via the opposite side of the mirror (and a symmetrically positioned fixed detector mirror). The average range is determined by the angular positioning of the fixed mirrors. The sensor creates a 128 × 256 range image in less than a second. The angular separation of the fixed mirrors is only 10 deg. For a total working volume of 250 mm × 250 mm × 100 mm, the x, y, z resolutions are 1, 2, and 0.4 mm, respectively (M = 45,255).

Servo-Robot (1987) manufactures the Saturn and the Jupiter line scan range sensors. Both are based on synchronous scanning. The Saturn system measures a 60 mm × 60 mm × 60 mm working volume from a standoff of 80 mm. The volume-center resolution is 0.06 mm in x and 0.05 mm in z (M = 32,860 for 3000 points/s). The Jupiter system measures a 1 m × 1 m × 1 m volume from a standoff of 0.1 m. The volume-center resolution is 1 mm in x and 0.3 mm in z (M = 91,290 for 3000 points/s).

Hymarc (1987) also makes a line scan sensor based on synchronous scanning. The sensor is accurate to 0.25 mm in a 500 mm × 500 mm × 500 mm working volume at a 600-mm standoff with a 3000 point per second data rate (M = 109,540).

Photonic Automation, Inc. (1987) is developing a commercially available sensor for fast ranging in a shallow depth of field. They claim a range accuracy of 25 μ over a depth of field of 6.25 mm at a speed of 10 million pixels per second (M = 790,570). The angular separation between source and detector is about 5 deg. Synthetic Vision Systems of Ann Arbor, Michigan has a competing unit.

Bickel et al. (1984) independently developed a mechanically coupled deflector arrangement for spot scanners similar in concept to the Rioux (1984) design. Bickel et al. (1985) addressed depth of focus problems inherent in triangulation systems for both illumination and detection. They suggest a teleaxicon lens and a laser source can provide a 25-μ spot that is in focus over a 100-mm range at a 500-mm standoff. Detection optics should be configured to satisfy the Scheimpflug (tilted detector plane) condition (Slevogt 1974) shown in Figure 9: $\tan \theta_{tilt} = 1/M \tan \theta_{sep}$ where θ_{sep} is the separation angle of the illumination direction and the detector's viewing direction, θ_{tilt} is the tilt angle of the photosensitive surface in the focusing region of the lens relative to the viewing direction, and $M = (w_c - f)/w_c$ is the on-axis magnification of the lens where w_c is the distance from the center of the lens to the center of the detector plane and f is the focal length of the lens. All points in the illumination plane are in exact focus in the detector plane. Using a 4000-element linear array detector, they get 25-μ range resolution, 13-μ lateral resolution, over a depth of field of 80 mm (M = 17,530 assuming 30 points/s rate). Tilted detector planes are used by some commercial

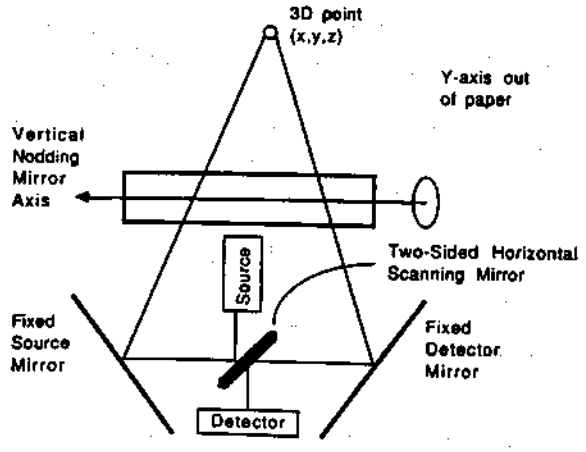


Figure 8. Synchronous scanning of source and detector.

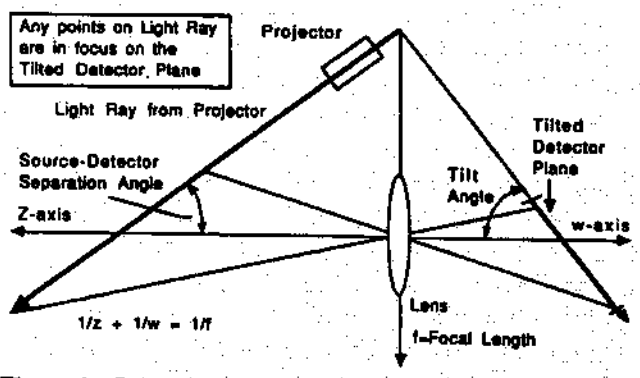


Figure 9. Scheimpflug condition: tilted detector to maintain focus for all depths.

vendors. Hausler and Maul (1985) examined the use of telecentric scanning configurations for point scanners. A telecentric system positions optical components at the focal length of the lens (or mirror).

Faugeras and Hebert (1986) used an in-house laser scanner. Their sensor uses a laser spot projector and two horizontal position detectors. Objects are placed on a turntable, and points are digitized as the object rotates. Scans are taken at several different heights to define object shape. No numbers were available to compute the figure of merit.

CyberOptics Corp. (1987) manufactures a series of point range sensors. For example, the PRS-30 measures a 300 μ depth of field from a standoff of 5 mm with 0.75-μ accuracy (1 part in 400). A precision xy-table (0.25 μ) provides object scanning under a stationary sensor at a rate of 15 points/s (M = 1550).

Diffracto, Ltd. (1987) also makes a series of point range sensors. Their Model 300 LaserProbe measures a depth of field of 2 mm from a standoff of 50 mm with an accuracy of 2.5 μ in 5 ms (M = 11,300). The detector handles a 50,000:1 dynamic range of reflected light intensities and works well for a variety of surfaces.

Kern Instruments (1987) has developed the System for Positioning and Automated Coordinate Evaluation (SPACE) using two automated Kern theodolites. This system measures points in a 3 m × 3 m × 3 m working volume to an accuracy of 50 μ (1 part in 60,000) at a rate of about 7.5 s per point (M = 21,910).

Lorenz (1984, 1986) has designed an optical probe to measure range with a repeatability of 2.5 μ over a depth of field of 100 mm (1 part in 40,000). He uses split-beam illumination and optimal estimation theory. The probe was tested on the z-axis of a CNC machining center. Even at one point per second, M = 40000.

The Selcom Opticator (1987) series are among the highest performance commercially available ranging point probes. They measure with one part in 4000 resolution at 16,000 points/s (M = 126,490 for 1 part in 1000 accuracy). The resolution of different models ranges from 2 to 128 μ in powers of two.

Pipitone and Marshall (1983) documented their experience in building a point scanning system. They measured with an accuracy of about 1 part in 400 over a depth of field of about 7.6 m (M = 8940 for 500 pts/s).

Haggren and Leikas (1987) have developed a four-camera photogrammetric machine-vision system with accuracy of better than 1 part in 10,000. The system generates one 3-D point every 1.5 s (M = 8160). Earlier similar photogrammetry work is found in Pinckney (1978) and Kratky (1979).

4.2 Structured Light: Line

Passing a laser beam through a cylindrical lens creates a line of light. Shirai (1972) and Will and Pennington (1972) were some of the first researchers to use light striping for computer vision. Nevatia and Binford (1973), Rocker (1974), and Popplestone et al. (1975) also used light striping. The General Motors Consight System (Holland et al. 1979) was one of the first industrial systems to use light stripe principles.

Technical Arts Corp. (1987) produces the 100X White Scanner. The camera and laser are typically separated by 45 deg or more. The system can measure up to a range of 2.4 m with a resolution of about 0.5 mm (M = 87,640 for 3000 pts/s and accuracy of 1.5 mm).

The IMAGE Lab at ENST in France developed a light stripe laser ranging system (Schmitt et al. 1985), commercially available from Studec. Schmitt et al. (1986) show a range image of a human head sculpture obtained with this sensor.

Cotter and Batchelor (1986) describe a depth map module (DMM) based on light striping techniques that produces 128 × 128 range images in about 4 s (M = 8192 assuming 7-bit resolution).

Silvaggi et al. (1986) describe a very inexpensive triangulation system (less than $1000 in component cost) that is accurate to 0.25 mm over a 50-mm depth of field at a standoff of 100 mm. A photosensitive RAM chip is used as the camera.

CyberOptics Corp. (1987) also manufactures a series of line range sensors. The LRS-30-500 measures a 300 μ depth of field and an 800-μ field of view from a standoff of 15 mm with 0.75 μ range accuracy (1 part in 400). A precision xy-table (0.25 μ) provides object scanning under the stationary sensor head at a rate of 5 lines/s (M = 7155 assuming only 64 points per line).

Perceptron (1987) makes a contour sensor that uses light striping and the Scheimpflug condition to obtain 25-μ accuracy over a 45-mm depth of field at a rate of 15 points/s (M = 6970).

Diffracto, Ltd. (1987) manufactures a Z-Sensor series of light stripe range sensors. Their Z-750 can measure a 19 mm depth of field with an accuracy of 50 μ from a standoff of 762 mm (M = 6100 assuming one 256 point line/s).

Landman and Robertson (1986) describe the capabilities of the Eyecrometer system available from Octek. This system is capable of 25 μ 3σ accuracy in the narrow view mode with a 12.7 mm depth

of field. The time for a high-accuracy scan is 9.2 s (M = 2680 assuming 256 pixels/scan).

Harding and Goodson (1986) implemented a prototype optical guillotine system that uses a high-precision translation stage with 2-μ resolution to obtain an accuracy of 1 part in 16,000 over a range of 150 mm. The system generates a scan in about 1 s (M = 256,000 assuming a 256-point scan).

The APOMS (Automated Propeller Optical Measurement System) built by RVSI (Robotic Vision Systems, Inc.) (1987) uses a high precision point range sensor mounted on the arm of a 5-axis inspection robot arm. The large working volume is 3.2 m $\times$ 3.5 m $\times$ 4.2 m. The accuracy of the optical sensor (x, y, z) coordinates is 64 μ in an 81 mm $\times$ 81 mm field of view. The linear axes of the robot are accurate to 2.5 μ, and the pitch and roll axes are accurate to 2 arc-seconds. The system covers 60 square feet per hour. Assuming 4 points per square millimeter, the data rate is about 6000 points/s (M = 3,485,700). The RVSI Ship Surface Scanner is a portable tripod mounted unit that has a maximum 70 deg $\times$ 70 deg field of view. The line scanner scans at an azimuthal rate of 8 deg/s. The range accuracy is about 1 part in 600 or about 5.7 mm at 3.66 m. The RVSI RoboLocator sensor can measure depths to an accuracy of 50 μ in a 25 mm $\times$ 25 mm field of view and a 50 mm depth of field. The RVSI RoboSensor measures about 1 part in 1000 over up to a 1-m depth of field in a 500 mm $\times$ 500 mm field of view. Assuming 3000 points/s, M = 54,000.

4.3 Structured Light: Miscellaneous

Kanade and Fuhrman (1987) developed an 18 LED light-source optical proximity sensor that computes 200 local surface points in 1 s with a precision of 0.1 mm over a depth of field of 100 mm (M = 14,140). Damm (1987) has developed a similar but smaller proximity sensor using optical fibers.

Labuz and McVey (1986) developed a ranging method based on tracking the multiple points of a moving grid over a scene. Lewis and Sopwith (1986) used the multiple-point-projection approach with a static stereo pair of images.

Jalkio et al. (1985) use multiple light stripes to obtain range images. The field of view is 60 mm $\times$ 60 mm with at least a 25-mm depth of field. The range resolution is about 0.25 mm with a lateral sampling interval of 0.5 mm. The image acquisition time was dominated by software processing of 2 min (M = 1170).

Mundy and Porter (1987) describe a system designed to yield 25-μ range resolution within 50 μ $\times$ 50 μ pixels at a pixel rate of 1 MHz while tolerating a 10 to 1 change in surface reflectance. The goals were met except the data acquisition speed is about 16 kHz (M = 32,380 assuming 8-bit accuracy).

Range measurements can be extracted from a single projected grid image, but if no constraints are imposed on the surface shapes in the scene, ambiguities may arise. Will and Pennington (1972) discussed grid-coding methods for isolating planar surfaces in scenes based on vertical and horizontal spatial frequency analysis. Hall et al. (1982) described a grid-pattern method for obtaining sparse range images of simple objects. Potmesil (1983) used a projected grid method to obtain range data for automatically generating surface models of solid objects. Stockman and Hu (1986) examined the ambiguity problem using relaxation labeling. Wang et al. (1985) used projected grids to obtain local surface orientation.

Wei and Gini (1983) proposed a structured light method using circles. They propose a spinning mirror assembly to create a converging cone of light that projects to a circle on a flat surface and an ellipse on a sloped surface. Ellipse parameters determine the distance to the surface as well as the surface normal (within a sign ambiguity).

If the light source projects two intersecting lines (X), it is easier to achieve subpixel accuracy at the point. The cross is created by a laser by using a beamsplitter and two cylindrical lenses. Pelowski (1986) discusses a commercially available Perceptron sensor that guarantees a $\pm 3\sigma$ accuracy in (x, y, z) of 0.1 mm over a depth of field of 45 mm in less than 0.25 s. Nakagawa and Ninomiya (1987) also uses the cross structure.

Asada et al. (1986) project thick stripes to obtain from a single image a denser map of surface normals than is possible using grid projection. The thickness of the stripes limits ambiguity somewhat because of the signed brightness transitions at thick stripe edges.

4.4 Structured Light: Coded Binary Patterns

Rather than scan a light stripe over a scene and process N separate images or deal with the ambiguities possible in processing a single gray scale multistripe image, it is possible to compute a range image using $N' = T\log_2 NT$ images where the scene is illuminated with binary stripe patterns. In an appropriate configuration, a range image can be computed from intensity images using lookup tables. This method is fast and relatively inexpensive.

Solid Photography, Inc. (1977) made the first use of gray-coded binary patterns for range imaging. A gantry mounted system of several range cameras acquired range data from a 2π solid angle around an

object. The system was equipped with a milling machine so that if a person had his or her range picture taken, a 3-D bust could be machined in a matter of minutes. The point accuracy of the multisensor system was about 0.75 mm in a 300 mm × 300 mm × 300 mm volume (M = 100,000 assuming 64K points/s).

Altschuler et al. (1981) and Potsdamer and Altschuler (1983) developed a numerical stereo camera consisting of a laser and an electrooptic shutter synchronized to a video camera. They used standard binary patterns and also performed experiments using two crossed electrooptic shutters (grid-patterns).

Inokuchi et al. (1984) and Sato and Inokuchi (1985) showed results from their system based on the gray-code binary pattern concept. More recently, Yamamoto et al. (1986) reported another approach based on binary image accumulation. A variation on the binary pattern scheme is given in Yeung and Lawrence (1986).

Rosenfeld and Tsikos (1986) built a range camera using 10 gray-code patterns on a 6-in. dia disk that rotates at 5 revolutions per second. Their system creates a 256 × 256 8-bit range image with 2-mm resolution in about 0.7 s (M = 78,330).

Vuylsteke and Oosterlinck (1986) developed another binary coding scheme. They use a projection of a specially formulated binary mask where each local neighborhood of the mask has its own signature. A 64 × 64 range image was computed from a 604 × 576 resolution intensity image in about 70 CPU s (VAX 11/750) (M = 1260 assuming 7-bit accuracy).

4.5 Structured Light: Color Coded Stripes

Boyer and Kak (1987) developed a real-time light striping concept that requires only one image frame from a color camera (no mechanical operations). If many stripes are used to illuminate a scene and only one monochrome image is used, ambiguities arise at depth discontinuities because it is not clear which image stripe corresponds to which projected stripe. However, when stripes are color coded, unique color subsequences can be used to establish the correct correspondence for all stripes. Although no figures are given, 128 × 128 images with 8-bit accuracy at a 7.5-Hz frame rate would yield M = 89,000.

4.6 Structured Light: Intensity Ratio Sensor

The intensity ratio method, invented by Schwartz (1983), prototyped by Bastuschek and Schwartz (1984), researched by Carrihill (1986), and documented by Carrihill and Hummel (1985), determines range unambiguously using the digitization and analysis of only three images: an ambient image, a projector-illuminated image, and a projected lateral attenuation filter image. The depth of field was 860 mm with a range resolution of 12 bits at a standoff of 80 cm, but an overall range repeatability of 2 mm. The total acquisition and computation time for a 512 × 480 image with a Vicom processor was about 40 s (M = 33,700).

4.7 Structured Light: Random Texture

Schewe and Forstner (1986) developed a precision photogrammetry system based on random texture projection. A scene is illuminated by a texture projector and photographed with stereo metric cameras onto high-resolution glass plates. Registered pairs of subimages are digitized from the plates, and a manually selected starting point initializes automated processing. The range accuracy of the points is about 0.1 mm over about a 1-m depth of field and a several-meter field of view. A complete wireframe model is created requiring a few seconds per point on a microcomputer (M = 10,000).

5. Moire Techniques

A moire pattern is a low spatial frequency interference pattern created when two gratings with regularly spaced patterns of higher spatial frequency are superimposed on one another. Mathematically, the interference pattern $A(x)$ from two patterns A_1, A_2 is

$$A(x) = A_1\{1 + m_1 \cos[\omega_1 x + \phi_1(x)]\} \cdot A_2\{1 + m_2 \cos[\omega_2 x + \phi_2(x)]\} \quad (9)$$

where the A_i are amplitudes, the m_i are modulation indices, the ω_i are spatial frequencies, and the $\phi_i(x)$ are spatial phases. When this signal is low-pass filtered (LPF) (blurred), only the difference frequency and constant terms are passed:

$$A'(x) = \mathrm{LPF}[A(x)] = A_1A_2(1 + m_1m_2 \cos[[\omega_1 - \omega_2)x + \phi_1(x) - \phi_2(x)]] . \quad (10)$$

For equal spatial frequencies, only the phase difference term remains. In moire range-imaging sensors, surface depth information is encoded in and recovered from the phase difference term. Reviews and bibliographies of moire methods may be found in Pirodda (1982), Sciammarella (1982), and Oster (1965). Theocaris (1969) provides some history of moire techniques (e.g., Lord Rayleigh 1874).

Moire range-imaging methods are useful for measuring the *relative distance* to surface points on a smooth surface $z(x, y)$ that does not exhibit depth

discontinuities. The magnitude of surface slope as viewed from the sensor direction should be bounded $\|\nabla z\| < K$. Under such constraints, absolute range for an entire moire image can be determined if the distance to one reference image point is known.

Moire methods for surface measurement use line gratings of alternating opaque and transparent bars of equal width (Ronchi gratings). The *pitch* P of a grating is the number of opaque/transparent line-pairs per millimeter (LP/mm). The period $p = 1/P$ of the grating is the distance between the centers of two opaque lines.

5.1 Projection Moire

Khetan (1975) gives a theoretical analysis of projection moire. In a projection moire system, a precisely matched pair of gratings is required. The projector grating is placed in front of the projector and the camera grating is placed in front of the camera as shown in Figure 10. The projector is located at an angle θ_l and the camera is located at an angle θ_v relative to the z-axis. The projected light is spatially amplitude modulated by the pitch of the projector grating, creating a spatial "carrier" image. When the projected beam falls on the smooth surface, the surface shape modulates the phase of the spatial carrier. By viewing these stripes through the camera grating, interference fringes are created at the camera. The camera grating "demodulates" the modulated carrier yielding a "baseband" image signal whose fringes carry information about surface shape. If p_o is the period of the projected fringes at the object surface, then the change in z between the centers of the interference fringes viewed by the camera is given by

$$\Delta z = \frac{p_o}{\tan(\theta_l) + \tan(\theta_v)}. \tag{11}$$

The angular separation of source and detector is

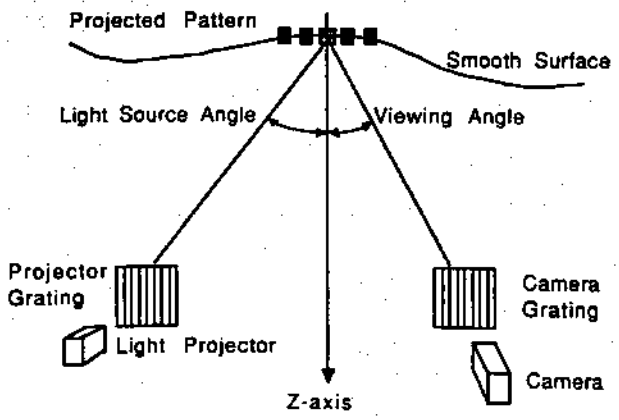


Figure 10. Projection moire configuration.

critical to range measurement and thus, moire may be considered a triangulation method (Perrin and Thomas 1979).

It is relatively inexpensive to set up a moire system using commercially available moire projectors, moire viewers, matched gratings, and video cameras (Newport Corp. 1987). The problem is accurate calibration and automated analysis of moire fringe images. Automated fringe analysis systems are surveyed in Reid (1986). The limitations of projection moire automated by digital image processing algorithms are addressed by Gasvik (1983). The main goal of such algorithms is to track the ridges or valleys of the fringes in the intensity surface to create 1-pixel wide contours. Phase unwrapping techniques are used to order the contours in depth assuming adequate spacing between the contours. It is not possible to correctly interpolate the phase (depth) between the fringes because between-fringe gray level variations are a function of local contrast, local surface reflectance, and phase change due to distance.

5.2 Shadow Moire

If a surface is relatively flat, shadow moire can be used. A single grating of large extent is positioned near the object surface. The surface is illuminated through the grating and viewed from another direction. Everything is the same as projection moire except that two matched gratings are not needed. Cline et al. (1982, 1984) show experimental results where 512 × 512 range images of several different surfaces were obtained automatically using shadow moire methods.

5.3 Single frame moire with reference

The projected grating on a surface can be imaged directly by a camera without a camera grating, digitized, and "demodulated" via computer software provided that a reference image of a flat plane is also digitized. As a general rule of thumb, single frame systems of this type are able to resolve range proportional to about 1/20 of a fringe spacing. Idesawa et al. (1977, 1980) did early work in automated moire surface measurement.

Electro-Optical Information Systems, Inc. (1987) has a commercially available range-imaging sensor of this type. On appropriate surfaces, the system creates a 480 × 512 range image in about 2 s using two array processors and has 1 part in 4000 resolution (M = 350,540 assuming accuracy of 1 part in 1000).

5.4 Multiple-frame phase-shifted moire

Multiple-frame (N-frame) phase-shifted moire is similar to single-frame moire except that after the

first frame of image data is acquired, the projector grating is precisely shifted laterally in front of the projector by a small distance increment that corresponds to a phase shift of $360/N$ degrees and subsequent image frames are acquired. This method, similar to quasi-heterodyne holographic interferometry, allows for an order of magnitude increase in range accuracy compared to conventional methods. Halioua and Srinivasan (1987) present a detailed description of the general moire concept. Srinivasan et al. (1985) show experimental results for a mannequin head using $N = 3$. They obtained 0.1-mm range accuracy over a 100-mm depth of field ($M = 46{,}740$ assuming 2 min. computation time for 512×512 images). Other research in this area has been reported by Andersen (1986).

Boehnlein and Harding (1986) implemented this approach on special hardware. The computations take less than 3.5 s for a 256×256 image, but the high-accuracy phase-shifting translation device (accurate to 0.1 μ) limited them to about 10 s for complete range image acquisition. The range resolution of the system is 11 μ over a 64-mm depth of field ($M = 121{,}430$ assuming 1 part in 1500 accuracy).

6. Holographic Interferometry

Holography was introduced in 1961 by Leith and Upatnieks (1962). The principles of holographic interferometry were discovered soon after (see Vest 1979, Schuman and Dubas 1979). Holographic interferometers use coherent light from laser sources to produce interference patterns due to the optical-frequency phase differences in different optical paths. If two laser beams (same polarization) meet at a surface point $\mathbf{x}$, then the electric fields *add* to create the net electric field:

$$E(\mathbf{x}, t) = E_1 \cos(\omega_1 t - \mathbf{k}_1 \cdot \mathbf{x} + \phi_1(\mathbf{x})) + E_2 \cos(\omega_2 t - \mathbf{k}_2 \cdot \mathbf{x} + \phi_2(\mathbf{x})) \quad (12)$$

where the $\mathbf{k}_i$ are 3-D wave vectors pointing in the propagation directions with magnitude $\|\mathbf{k}_i\| = 2\pi/\lambda_i$, the $\omega_i = \|\mathbf{k}_i\|c$ are the radial optical frequencies, and $\phi_i(\mathbf{x})$ are the optical phases. Since photodetectors respond to the square of the electric field, the detectable irradiance (intensity) is $I(\mathbf{x}, t) = E^2(\mathbf{x}, t)$. Photodetectors themselves act as low-pass filters of the irradiance function I to yield the detectable interference signal $I'(\mathbf{x}, t) = \mathrm{LPF}[I(\mathbf{x}, t)]$, or

$$I'(\mathbf{x}, t) = E_a\{1 + E_b \cos[\Delta\omega t + \Delta\mathbf{k} \cdot \mathbf{x} + \Delta\phi(\mathbf{x})]\} \quad (13)$$

where

$$E_a = E_1^2 + E_2^2/2 \text{ and}$$
$$E_b = 2E_1E_2/(E_1^2 + E_2^2),$$

$\Delta\omega = \omega_1 - \omega_2$ is the difference frequency, $\Delta\mathbf{k} = \mathbf{k}_2 - \mathbf{k}_1$ is the difference wave vector, and $\Delta\phi(\mathbf{x}) = \phi_1 - \phi_2$ is the phase difference. This equation is of the exact same form as the moire equation (10) above for $A'(x)$ except that a time-varying term is included. Since phase changes are proportional to optical path differences in holographic interferometry, fraction of a wavelength distances can be measured. For equal optical frequencies and equal (wave vector) spatial frequencies, only the phase difference term remains. In holographic interferometric range sensors, surface depth information is encoded in and recovered from the phase difference term. Just as the z-depth spacing of moire fringes is proportional to the period of grating lines, the z-depth spacing of holographic interference fringes is proportional to the wavelength of the light. Measured object surfaces must be very flat and smooth.

6.1 Conventional Holography

Conventional interferometry is somewhat like conventional projection moire in that the frequencies of the interfering beams are equal and between-fringe ranging is not possible. There are three types of conventional holographic interferometry used in industrial applications: (1) real-time holography, which allows observers to see instantaneous microscopic changes in surface shape, (2) double-exposure holographic systems, which provide permanent records of surface shape changes, (3) time-average holography, which produces vibration mode maps useful for verifying finite element analyses.

Conventional holographic interferometry is used to visualize stress, thermal strains, pressure effects, erosion, microscopic cracks, fluid flow, and other physical effects in nondestructive testing. Tozer et al. (1985), Mader (1985), Wuerker and Hill (1985), and Church et al. (1985) provide a sampling of industrial uses of holographic interferometry. The Holomatic 8000 (Laser Technology 1986) and the HC1000 Instant Holographic Camera (10-s development time on erasable thermoplastic film) (Newport Corp. 1987) are commercially available holographic camera systems.

6.2 Heterodyne Holography

Heterodyne holographic interferometers cause two coherent beams of slightly different optical frequencies (less than 100 MHz generates RF beat frequencies) to interfere creating time-varying holographic

fringes in the image plane. Optical frequency shifts are achieved by acoustooptic modulators, rotating quarter wave plates, rotating gratings, and other methods. Optical phase measurements corresponding to optical path differences are made at each point by electronically measuring the phase of the beat frequency signal relative to a reference using a phasemeter. The time-varying interference fringe image is mechanically scanned with a high-speed detector to obtain a range image. Heterodyne holographic interferometers can make out-of-plane surface measurements with nanometer resolution over several microns, but they are typically slow. The general rule of thumb is that $\lambda/1000$ resolution is possible using heterodyne methods.

Pantzer et al. (1986) built a heterodyne profilometer that has a mechanical-vibration–limited range resolution of 5 nm and a lateral resolution of 3 μ. The theoretical resolution of this method is 0.4 nm if mechanical instabilities were removed. It took about 20 s to linearly scan 1 mm to get 330 points. (M = 2450 assuming a 3-μ depth of field).

Dandliker and Thalman (1985) obtained 0.2-nm range resolution over a depth of field of 3 μ at a rate of 1 point per second over a lateral range of 120 mm using a double-exposure heterodyne interferometer (M = 7500 assuming 0.4 nm accuracy).

Pryputniewicz (1985) used heterodyne interferometry to study the load-deformation characteristics of surface mount components on a printed circuit board. The reported 3σ range accuracy was 2 nm.

Sasaki and Okazaki (1986) developed a variation on frequency-shift heterodyne methods. The reference path mirror is mounted on a piezoelectric transducer (PZT) modulated at about 220 Hz. This phase modulation provides the needed small frequency shift for heterodyne accuracy. This is slow enough that image sensors can be used to collect the video signals. They obtained repeatable range measurements at less than 1 nm resolution. Over a 250 × 250 μ field of view, the lateral resolution is about 5 μ.

6.3 Quasi-Heterodyne (Phase-Shifted) Methods

Phase-shifted holographic interferometers are referred to as quasi-heterodyne since their $\lambda/100$ range resolution is not quite heterodyne performance, but is much better than conventional. Quasi-heterodyne systems can be much simpler, much cheaper, and much faster than heterodyne systems by trading off some range resolution. Standard video cameras can be used to image several frames of holographic fringes. Phase-shifts can be achieved at every pixel in parallel in real-time using a piezoelectric translator to move a mirror. (Compare this to the lateral shifting of a grating in front of a projector in phase-shifted moire.) Other phase-shifting methods are possible. The computations are very similar to those described in the previous section on multiple frame phase-shifted moire.

Hariharan (1985) used a 100 × 100 camera to digitize the holographic fringes needed to compute the range image. The measurement cycle for each fringe image was about 150 ms, and the total computation time was 10 s using a machine-language program. They used the same formulas as Boehnlein and Harding (1986) discussed above. Results are shown for a large 50 mm × 100 mm field of view (M = 8095 assuming 8-bit accuracy).

Thalman and Dandliker (1985) and Dandliker and Thalmann (1985) examine two-reference beam interferometry and two-wavelength contouring for quasi-heterodyne and heterodyne systems.

Chang et al. (1985) did experiments in digital phase-shifted holographic interferometry to eliminate the need to calibrate the phase shifter as in Hariharan et al. (1983). They claim an accuracy of 2 nm over a 300-nm depth of field.

6.4 Microscopic Interferometry

Peterson et al. (1984) measured VHS video tape surfaces with an interferometer obtaining 1 μ lateral resolution and 1 nm range repeatability.

Matthews et al. (1986) describe a phase-locked loop interferometric method where the two arms of a confocal interference microscope are maintained in quadrature by using an electrooptic phase modulator. Results are shown where the system scanned a 3-μ × 3-μ field of view over a depth of field of 300 nm in 2 s with a range accuracy of 1 nm (M = 27,150).

7. Focusing

Horn (1968), Tenenbaum (1970), Jarvis (1976), and Krotkov (1986) have discussed focusing for range determination. Figure 11 shows basic focusing relationships. Pentland (1987), Grossman (1987), Krotkov and Martin (1986), Schlag et al. (1983), Jarvis (1976), and Harvey et al. (1985) discuss passive methods to determine range from focus.

The autofocus mechanisms in cameras act as range sensors (Denstman 1980, Goldberg 1982), but most commercially available units do not use focusing principles to determine range. The Canon "Sure-Shot" autofocusing mechanism is an active triangulation system using a frequency modulated infrared beam. Jarvis (1982) used this Canon sensor

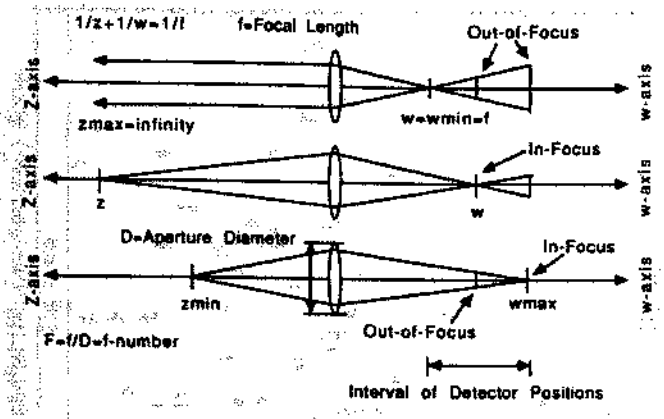


Figure 11: Thin lens relationships.

module to create a 64 × 64 range image in 50 min. The Honeywell Visitronic module for Konica, Minolta, and Yashica cameras is a passive triangulation system that correlates photocell readouts to achieve a binocular stereomatch and the corresponding distance. The Polaroid autofocusing mechanism is a broad beam sonar unit.

Rioux and Blais (1986) developed two techniques based on lens focusing properties. In the first technique, a grid of point sources is projected onto a scene. The range to each point is determined by the radius of the blur in the focal plane of the camera. The system was capable of measuring depths to 144 points with 1-mm resolution over a 100-mm depth of field. The second technique uses a multistripe illuminator. If a stripe is not in focus, the camera sees split lines where the splitting distance between the lines is related to the distance to the illuminated surface. Special purpose electronics process the video signal (Blais and Rioux 1986) and detect peaks to obtain line splitting distances on each scan line and hence range. The system creates a 256 × 240 range image in less than 1 s by analyzing 10 projected lines in each of 24 frames. The projected lines are shifted between each frame. A resolution of 1 mm over a depth of 250 mm is quoted at a 1-m standoff for a small robot-mountable unit (M = 63,450).

Kinoshita et al. (1986) developed a point range sensor based on a projected conical ring of light and focusing principles. A lens is mechanically focused to optimize the energy density at a photodiode. The prototype system measured range with a repeatability of 0.3 mm over a depth of field of 150 mm (9 bits) with a standoff distance of 430 mm.

Corle et al. (1987) measured distances with accuracies as small as 40 nm over a 4-μ depth of field using a type II confocal scanning optical microscope.

8. Fresnel Diffraction

Talbot (1836) first observed that if a line grating $T(x, y) = T(x + p, y)$ with period p is illuminated with coherent light, exact in-focus images of the grating are formed at regular periodic (Talbot) intervals D. This is the self-imaging property of a grating. Lord Rayleigh (1881) first deduced that $D = 2p^2/\lambda$ when $p \gg \lambda$. The Talbot effect has been analyzed more recently by Cowley and Moodie (1957) and Winthrop and Worthington (1965). For cosine gratings, grating images are also reproduced at $D/2$ intervals with a 180-deg phase shift. Thus, the ambiguity interval for such a range sensor is given by $I_r = p^2/2\lambda = D/4$. Ambiguity resolving techniques are needed for larger depths of field. The important fact is that the grating images are out of focus in a predictable manner in the ambiguity interval such that local contrast depends on the depth z. Figure 12 shows the basic configuration for measuring distance with the Talbot effect.

The Chavel and Strand (1984) method illuminates an object with laser light that has passed through a cosine grating. A camera views the object through a beam-splitter so that the grating image is superimposed on the returned object image that is modulated by (1) the distance to object surface points and by (2) the object surface reflectivity. The contrast ratio of the power in the fundamental frequency p^{-1} to the average (dc) power is proportional to depth and can be determined in real-time by analog video electronics. The analog range-image signal was digitized to create an 8-bit 512 × 512 image representing a 20 mm × 20 mm field of view approximately. The ambiguity interval was 38 mm. The digitizer averaged 16 frames so that the frame time is about 0.5 s (M = 92,680 assuming 7-bit accuracy).

Leger and Snyder (1984) developed two techniques for range imaging using the Talbot effect.

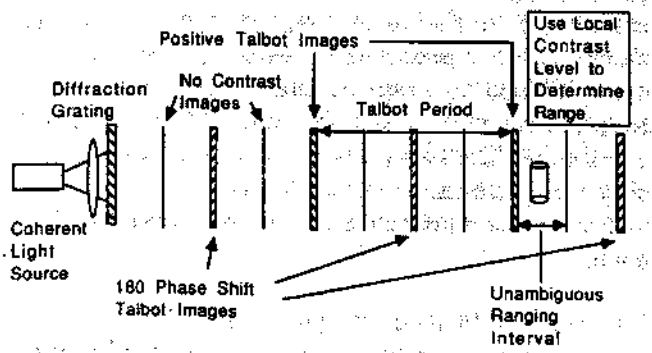


Figure 12. Talbot effect or self-imaging property of gratings for ranging.

The first method used two gratings crossed at right angles to provide two independent channels for depth measurement. The second method uses a modulated grating created by performing optical spatial filtering operations on the original signal emanating from a standard grating. Two prototype sensors were built to demonstrate these methods. The ambiguity intervals were 7.3 mm and 4.6 mm. The figure of merit is similar to the Chavel and Strand sensor. Speckle noise (Goodman 1986, Leader 1986) is a problem with coherent light in these methods, and good range resolution is difficult to obtain from local contrast measures. Other research in this area has been pursued by Hane and Grover (1985).

9. Sensor Comparisons

The key performance factors of any range-imaging sensor are listed in the following table:

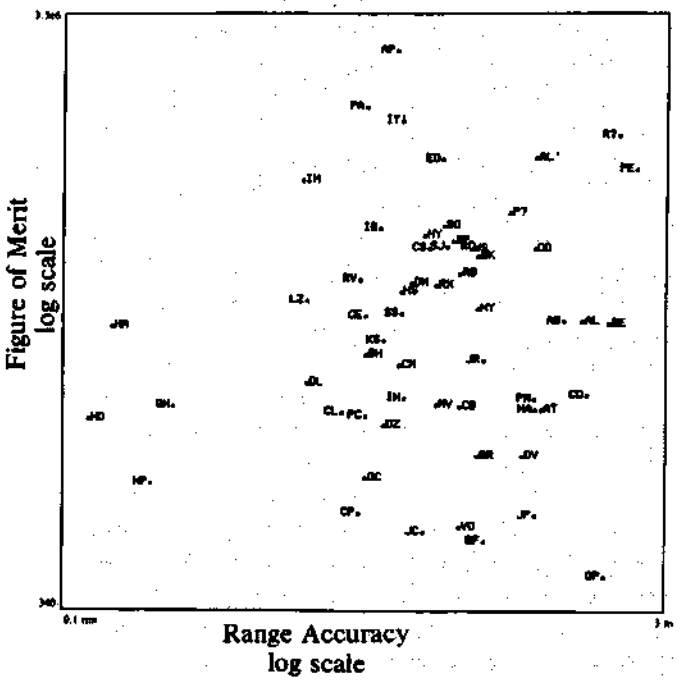
Depth of field	L_r
Range accuracy	σ_r
Pixel dwell time	T
Pixel rate	$1/T$
Range resolution	N_{bits}
Image size	$N_x \times N_y$
Angular field of view	$\theta_x \times \theta_y$
Lateral resolution	$\theta_x/N_x \times \theta_y/N_y$
Standoff distance	L_s
Nominal field of view	$(L_s + L_r/2)\theta_x \times (L_s + L_r/2)\theta_y$
Frame time	$T \times N_x \times N_y$
Frame rate	$1/(T \times N_x \times N_y)$

The figure of merit M used to evaluate sensors in this survey only uses the first three values. A full evaluation for a given application should consider all sensor parameters.

Different types of range imaging sensors are compared by showing the rated sensors in the survey in two scatterplots. In Figure 13, range-imaging sensors are shown at the appropriate locations in a plot of (log) figure of merit M versus (log) range accuracy σ. In Figure 14, range imaging sensors are shown at the appropriate locations in a plot of (log) depth-of-field to range-accuracy ratio (number of accurate range bits) as a function of the (log) pixel dwell time. The two plots in Figures 13 and 14 display the quantitative comparisons of rated sensors and show the wide range of possible sensor performance.

9.1 General Method Comparisons

The six optical ranging principles are briefly summarized below. Imaging laser radars are capable of



AL - Erim-ALV
AS - Erim-ASV
BF - Behaim-Fritsch
CB - Cotter-Batch
CM - CMU:Kanade-Fuhrman
DL - Diffracto-LaserProbe
DZ - Diffracto-Z-Series
HA - Heikkinen,Ahola
KP - Panteer-et-al
IR - ITI:Boshaleis-Harding
IT - New-Erim-ITA
JR - Jarvis-83
MV - Mapvision-Haggren
OD - Odetics-Inc
PA - Photonic-Automation
PM - Pipitone-Marshall
RB - NRCC:Rioux-Blais
RX - NRCC:Rioux
SP - Solid-Photography
VD - Vuylsteke,Oosterlinck
AL' - New-Erim-ALV
AT - ATT:Miller,Wagner
BW - Bickel-et-al
CD - Cathey,Davis
CP - CyberOptics-PRS
DM - Herzman-DO-Matrol
DO - EOIS:Fitts
HD - Dandliker-Thalmann
HS - NYIT:Halioua-Srini
IR - ITI:Harding-Goodson
JC - Jalkio-Kim-Case
KS - Kern-SPACE
NY - NYU:Intensity-Ratio
OP - Optech:Larsen500
PC - Perceptron-Contour
QR - Hariharan
RC - RCA:Rosenfeld-Tsikos
RJ - ServoRobot-Jupiter
SR - SRI:Mitsan,et-al
WS - Tech-Arts-White
AP - ERIM-APGMS
BE - Boulder-EO
BK - Boyer-Kak-Color
CL - CyberOptics-LRS
CS - Chavel-Strand
DV - Herzman-DO-Vision
GE - GE:Handy-Porter
HM - Matthews-et-al
HY - Symarc-Hyscan
IN - Forstner-Indusurf
JP - JPL:Lewis-Johnston
LZ - Lorenz
OC - Ostak-Synchrometer
PT - Perceptron-Radar
PE - Perkin-Elmer
RF - ERIM-Rose
RV - ERIM-other
SO - Selcom-Optocator
SS - ServoRobot-Saturn

Figure 13. Figure of merit vs. range accuracy.

range accuracies from about 50 μ to 5 m over depths of field 250 to 25,000 times larger. They benefit from having very small source to detector separations and operate at higher speeds than many other types of range-imaging sensors because range is determined electronically. They are usually quite expensive, with commercially available units starting at around $100,000. Existing laser radars are sequential in data acquisition (they acquire one point at a time) although parallel designs have been suggested.

Triangulation sensors are capable of range accuracies beginning at about 1 μ over depths of field from 250 to 60,000 times larger. In the past, some have considered triangulation systems to be inaccurate or slow. Many believe that large baselines are required for reasonable accuracy. However, triangulation systems have shown themselves to be accurate, fast, and compact mainly owing to the advent of synchronous scanning approaches. Simple triangulation systems start between $1000 and

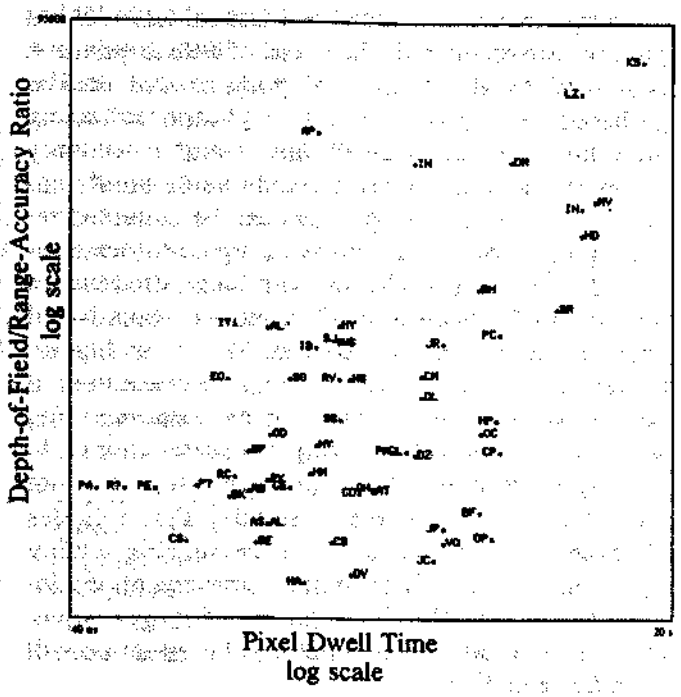


Figure 14. Depth-of-field/range-accuracy ratio versus pixel dwell time.

$10,000 depending on how much you put together yourself and how much needed equipment you already have. Commercially available turnkey systems can easily run upwards of $50,000, and fancier systems can run into the hundreds of thousands if there are requirements for fine accuracy over large working volumes. Triangulation systems go from totally sequential as in point scanners to almost parallel as in the intensity ratio scheme or the color encoded stripe scheme. Triangulation systems have been the mainstay of range imaging and promise to remain so.

Moire systems are limited to about the same accuracies as triangulation sensors (a few microns) and are not applicable unless surface slope constraints are satisfied. The depth of field of a moire system depends on the camera resolution and the object grating period p_o. For a 512 × 512 camera and a minimum of about 5 pixels per fringe, 100 phase transitions can be unwrapped yielding a depth of field on the order of 100 p_o. Optical moire components are a small part of the total system cost if fast computer hardware is used to carry out the necessary computations. Image array processors vary in cost, but a complete moire system with reasonable speed will probably run more than $50,000. Moire techniques are inherently parallel and will benefit from the development of parallel computing hardware.

Holographic interferometer systems can measure with accuracies of less than half a nanometer over as many wavelengths of light as can be disambiguated. Surface slope and smoothness constraints must be met before holographic methods are valid. The most accurate heterodyne methods are also the slowest and the most expensive. The quasi-heterodyne methods are faster and cheaper, but give up about an order of magnitude in accuracy compared to heterodyne. Holographic techniques are also inherently parallel and should benefit from the development of parallel computing hardware. Holographic systems are generally much more specialized than other optical techniques, and are applicable to fine grain surface inspection and nondestructive testing.

The Fresnel diffraction techniques based on the Talbot effect offer video frame rate range images using special-purpose analog video electronics. The range resolution of these systems is limited by the resolution of local contrast measures; it appears to be difficult to get more than seven or eight bits of range. Diffraction ranging is also inherently parallel.

Active focusing methods have great potential for compact, inexpensive range-imaging sensors, but high-precision systems are not likely.

Tactile methods still dominate many potential range-imaging applications where industry needs to exactly specify the shape of a prototype object. The reliability and accuracy of coordinate measuring machines (CMM's) over very large working volumes are hard to beat, but they are inherently slow and very expensive. If flexible noncontact optical methods can provide similar performance with reliability and ease of use, then a significant cost savings will be realized in applications currently requiring CMM's. At very fine scales, the (nonoptical) scanning tunneling microscope (Binnig and Rohrer 1985) is the state-of-the-art in very accurate (0.01 nm) surface studies. It is clear that active, optical ranging sensors have competition from other techniques.

Comments from this section and the survey are summarized in Figure 15. The first range value for each method in this table (ACC) is a good nominal accuracy rounded to the nearest power of ten

Category	ACC/DOF	Notes
Radar (Pulse,AM,FM)	0.1 mm 100 m	Detect Time, Phase, or Frequency Difference Signal Depends on Range, Surface Normal, Reflectance Beam Scanning Usually Required, No Computation History: Since 1903, Well known since 40's, Lasers since 70's Cost: Inexpensive to Extremely Expensive
Triangulation	1 μm 100 m	1 or More Cameras, 1 or more Projectors Scanned Point, Scanned Stripe, Multi-Stripe, Grid Binary Pattern, Color, Texture, Intensity Ratio Terms: Synchronous Scan, Scheimpflug Condition History: Since 300 B.C., Most Popular Method Cost: Inexpensive to Very Expensive
Moire Techniques	1 μm 10 m	Projector, Grating(s), Camera, Computer Fringe Tracking: Projection, Shadow Reference: Single-Frame, Multi-Frame (Phase-Shifted) Surface Slope Constraint, Non-coherent Light Computation Required, No Scanning History: Since 1859, Used Since 1960's in Mech.Eng. Cost: Inexpensive (excluding Computer)
Holographic Interferometry	0.1 mm 100 μm	Detector, Laser, Optics, Electronics, Computer Conventional: Real-Time, 2-Exposure, Time-Avg. Quasi-Heterodyne (Phase-Shifted), Heterodyne Surface Slope Constraint, Coherent Light Computation/Electronics Required, No Scanning History: Not Practical until Laser 1961, Big in NDT Cost: Inexpensive to Expensive
Focusing	1 mm 10 m	Measure Local Contrast, Blur, Displacement Limited Depth-of-Field to Accuracy Ratio History: Since 1800's, Gauss thin lens law Computation/Electronics Required, No Scanning Potential for Inexpensive Systems
Fresnel Diffraction (Talbot Effect)	0.1 μm 10 m	Laser, Grating, Camera / Not Explored by Many Video Rates, Limited Accuracy, Uses Local Contrast Electronics Required, No Scanning History: Discovered 1836, Used 1983 Potential for Inexpensive Systems

Figure 15. General comments on fundamental categories.

whereas the second value is the maximum nominal depth of field. Figure 16 indicates in a brief format the types of applications where the different ranging methods are being used or might be used.

10. Emerging Themes

As in any field, people always want equipment to be faster, more accurate, more reliable, easier to use, and less expensive. Range-imaging sensors are no exception. But compared to the state of the art 10 years ago, range imaging has come a long way. An image that took hours to acquire now takes less than a second. However, the sensors are only one part of the technology needed for practical automated systems. Algorithms and software play an even bigger role, and although research in range-image analysis and object recognition using range images (Besl and Jain 1985) has come a long way in recent years, there is still much to be done to achieve desired levels of performance for many applications.

Application	Radar	Trian	Moire	Holog	Focus	Diffr
Cartography	X	X				
Navigation	X	X			X	
Medical	X	X	X			
Shape Definition	X	X	X	X		
Bin Picking	X	X			X	X
Assembly	X	X	X	X	X	X
Inspection	X	X	X	X		X
Gauging	X	X	X	X		X

Figure 16. Methods and applications of range-imaging sensors.

Image acquisition speed is a critical issue. Since photons are quantized, the speed of data acquisition is limited by the number of photons that can be gathered by a pixel's effective photon collecting area during the pixel dwell time. Greater accuracy or faster frame times are possible using higher energy lasers since more photons can be collected reducing shot noise and improving signal-to-noise ratio. But today's higher-power laser diodes are difficult to focus to a small point size because of irregularities in the beam shapes. Moreover, higher-power lasers are a greater threat to eye safety if people will be working close to the range-imaging sensors (see appendix). Longer wavelengths (1.3–1.55 μ) are desirable for better eye safety, but not enough power is available from today's laser diodes at these wavelengths to obtain reasonable quality range images. The fiber optics communications industry is driving the development of longer wavelength laser diodes, and hopefully this situation will soon be remedied.

Another issue in the speed of data acquisition is scanning mechanisms. Many sensors are limited by the time for a moving part to move from point A to point B. Image dissector cameras are being explored by several investigators to avoid mechanical scanning. Mechanical scanning is a calibration and a reliability problem because moving parts do eventually wear out or break. However, today's mechanical scanners can offer years of reliable service.

Once considered state-of-the-art, 8-bit resolution sensors are giving way to sensors with 10 to 12 bits or more of resolution and possibly accuracy. Processing this information with inexpensive image processing hardware designed for 8-bit images is inappropriate. A few commercial vendors provide 16-bit and floating point image processing hardware, but it is generally more expensive.

Reliable subpixel image location is being achieved in many single light stripe triangulation sensors. It is commonly accepted that a fourth, a fifth, an eighth, or a tenth of a pixel accuracy can realistically be obtained with intensity weighted averaging techniques. Moreover, Kalman filtering (recursive least squares) algorithms (see e.g. Smith and Cheeseman 1987) are beginning to be used in vision algorithms for optimally combining geometric information from different sensing viewpoints or different range sensors. Such efforts will continue to increase the accuracy of sensors and systems.

Although not specifically mentioned, many range sensors also acquire registered intensity images at the same time. Although there is little 3-D metrology information in these images, there is a great

deal of other useful information that is important for automated systems. A few researchers have addressed methods for using this additional information, but commercially available software solutions are more than several years away.

Range-imaging sensors are the data-gathering components of range-imaging systems, and ranging imaging systems are machine perception components of application systems. Algorithms, software, and hardware are typically developed in isolation and brought together later, but there are trends toward developing hardware that can incorporate programmability features that expedite operations common to many applications.

Acknowledgments. The author would like to express his appreciation to R. Tilove and W. Reguiro for their thorough reviews, and to G. Dodd, S. Walter, R. Khetan, J. Szczesniak, M. Stevens, S. Marin, R. Hickling, W. Wiitanen, R. Smith, T. Sanderson, M. Dell'Eva, H. Stern, and J. Sanz.

References

Agin GJ, Highnam, PT (1983) Movable light stripe sensor for obtaining 3D coordinate measurements. Proceedings SPIE Conference on 3-D Machine Perception (360):326

Ahola R, Heikkinen T, Manninen M (1985) 3D image acquisition by scanning time of flight measurements. Proceedings International Conference on Advances in Image Processing and Pattern Recognition

Altschuler MD, Altschuler BR, Toboada J (1981) Laser electro-optic system for rapid 3D topographic mapping of surfaces. Optical Engineering 20(6):953–961

Andresen K (1986) The phase shift method applied to moire image processing. Optik 72:115–119

ANSI 1986. American National Standard for the safe use of lasers. (ANSI Z136.1-1986) American National Standards Institute, New York

Asada M, Ichikawa H, Tsuji S (1986) Determining surface property by projecting a stripe pattern. Proceedings International Conference on Pattern Recognition IEEE-CS, IAPR: 1162–1164

Banic J, Sizgoric S, O'Neill R (1987) Airborne scanning lidar bathymeter measures water depth. Laser Focus/Electro-Optics: 48–52

Bastuschek CM, Schwartz JT (1984) Preliminary implementation of a ratio image depth sensor. Robotics Research Report No. 28, Courant Institute of Mathematical Sciences, New York University, New York

Beheim G, Fritsch K (1986) Range finding using frequency-modulated laser diode. Applied Optics, 25(9):1439–1442

Besl PJ (1987) Range imaging sensors. Tech. Report GMR-6090, Computer Science Dept., General Motors Research Labs, Warren, MI

Besl PJ (1988) Active optical range imaging sensors. In: Advances in Machine Vision: Architectures and Applications. J. Sanz (Ed.), Springer-Verlag, New York

Besl PJ, Jain RC (1985) Three dimensional object recognition. ACM Computing Surveys 17(1):75–145

Bickel G, Hausler G, Maul M (1984) Optics in Modern Science and Technology, Conf. Dig. ICO-13:534

Bickel G, Hausler G, Maul M (1985) Triangulation with expanded range of depth. Optical Engineering 24(6):975–979

Binger N, Harris SJ (1987) Applications of laser radar technology. Sensors 4(4):42–44

Binnig G, Rohrer H (1985) The scanning tunneling microscope. Scientific American 253,2 (Aug), 50–69

Blais F, Rioux M (1986) Biris: a simple 3D sensor. Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision, 728:235–242

Boehnlein AJ, Harding KG (1986) Adaptation of a parallel architecture computer to phase-shifted moire interferometry. Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision, 728:132–146

Boulder Electro-Optics (1986) Product information, Boulder, Colorado. (now Boulder Melles Griot)

Boyer KL, Kak AC (1987) Color encoded structured light for rapid active ranging. IEEE Transactions Pattern Analysis Machine Intelligence PAMI-9, 1:14–28

Brou P (1984) Finding the orientation of objects in vector maps. International Journal of Robot Research 3:4

Bumbaca F, Blais F, Rioux M (1986) Real-time correction of 3D nonlinearities for a laser rangefinder. Optical Engineering 25(4):561–565

Carrihill B (1986) The intensity ratio depth sensor. Ph.D. dissertation, Courant Institute of Mathematical Sciences, New York University, New York

Carrihill B, Hummel R (1985) Experiments with the intensity ratio depth sensor. Computer Vision, Graphics, Image Processing 32:337–358

Case SK, Jalkio JA, Kim RC (1987) 3D vision system analysis and design. In: Three-Dimensional Machine Vision, T. Kanade (Ed.), Kluwer Academic, Boston, pp. 63–96

Cathey WT, Davis WC (1986) Vision system with ranging for maneuvering in space. Optical Engineering 24(7):821–824. See also Imaging system with range to each pixel. Journal of the Optical Society of America A 3(9):1537–1542

CDRH 1985. Federal Register, Part III, Dept. of Health and Human Services, 21 CFR Parts 1000 and 1040 [Docket No. 80N-0364], Laser Products; Amendments to Performance Standard; Final Rule. For further info, Contact Glenn Conklin, Center for Devices and Radiological Health (HFZ-84), U.S. Food and Drug Administration, 5600 Fishers Lane, Rockville, MD 20857

Chang M, Hu CP, Lam P, Wyant JC (1985) High precision deformation measurement by digital phase shifting holographic interferometry. Applied Optics 24(22):3780–3783

Chavel P, Strand TC (1984) Range measurement using Talbot diffraction imaging of gratings. Applied Optics 23(6):862–871

Church EL, Vorburger TV, Wyant JC (1985) Direct comparison of mechanical and optical measurements of the finish of precision machined optical surfaces. Optical Engineering 24(3):388–395

Cline HE, Holik AS, Lorenson WE (1982) Computer-aided surface reconstruction of interference contours. Applied Optics 21(24):4481–4489

Cline HE, Lorenson WE, Holik AS (1984) Automated moire contouring. Applied Optics 23(10):1454–1459

Corle TR, Fanton JT, Kino GS (1987) Distance measurements by differential confocal optical ranging. Applied Optics 26(12):2416–2420

Cotter SM, Batchelor BG (1986) Deriving range maps at real-time video rates. Sensor Review 6(4):185–192

Cowley JM, Moodie AF (1957) Fourier images: I—the point source. Proceedings Physical Society 70:486–496

Cunningham R (1986) Laser radar for the space conscious. Lasers and Applications July: 18–20

Cyberoptics (1987) Product information. Minneapolis, MN

Damm L (1987) A minimum-size all purpose fiber optical proximity sensor. Proceedings Vision'87 Conference: 6-71—6-91

Dandliker R, Ineichen B, Mottier F (1973) Optics Communications 9:412

Dandliker R (1980) Heterodyne holography review. Progress in Optics 17:1

Dandliker R, Thalmann R (1985) Heterodyne and quasi-heterodyne holographic interferometry. Optical Engineering 24(5):824–831

Dandridge A (1982) Current induced frequency modulation in diode lasers. Electron. Letters 18:302

Denstman H (1980) State-of-the-art optics: Automated image focusing. Industrial Photography, July: 33—37

Dereniak EL, Crowe DG (1984) Optical Radiation Detectors. Wiley, New York

Diffracto (1987) Product Literature. Laser probe digital ranging sensor. Diffracto, Ltd., Windsor, Canada

Digital Optronics (1986) Product literature. Springfield, VA

Dimatteo PL, Ross JA, Stern HK (1979) Arrangement for sensing the geometric characteristics of an object. (RVSI) U.S. Patent 4175862

Electro-Optical Information Systems (1987) Product Information. EOIS, Santa Monica, CA

Faugeras OD, Hebert M (1986) The representation, recognition, and locating of 3-D objects. International Journal of Robotic Research 5(3):27–52

Froome KD, Bradsell RH (1961) Distance measurement by means of a light ray modulated at a microwave frequency. Journal of Scientific Instrumentation 38:458–462

Gasvik KJ (1983) Moire technique by means of digital image processing. Applied Optics 22(23):3543–3548

Goldberg N (1982) Inside autofocus: How the magic works. Popular Photography, Feb: 77–83

Goodman JW (1986) A random walk through the field of speckle. Optical Engineering 25(5):610–612

Gottlieb M (1983) Electro-Optic and Acousto-Optic Scanning and Deflection. Marcel-Dekker, New York

Griffin DR (1958) Listening in the dark: The acoustic orientation of bats and men. Yale University Press, New Haven, CT

Grossman P (1987) Depth from Focus. Pattern Recognition Letters 5(1):63–69

Haggren H, Leikas E (1987) Mapvision—The photogrammetric machine vision system. Proceedings Vision'87 Conference: 10-37–10-50

Halioua M, Srinivasan V (1987) Method and apparatus for surface profilometry. New York Institute of Technology, Old Westury, NY. U.S. Patent 4,641,972

Halioua M, Krishnamurthy RS, Liu H, Chiang FP (1983) Projection moire with moving gratings for automated 3D topography. Applied Optics 22(6):850–855

Hall EL, Tio JBK, McPherson CA, Sadjadi FA (1982) Measuring curved surfaces for robot vision. Computer 15(12):42–54

Hane K, Grover CP (1985) Grating imaging and its application to displacement sensing. Journal of Optical Society of America A 2(13):9

Harding KG (1983) Moire interferometry for industrial inspection. Lasers and Applications Nov.: 73

Harding KG, Goodson K (1986) Hybrid high accuracy structured light profiler. Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision 728:132–145

Harding KG, Tait R (1986) Moire techniques applied to automated inspection of machined parts. Proceedings Vision'86 Conference, SME, Dearborn, MI

Hariharan P (1985) Quasi-heterodyne hologram interferometry. Optical Engineering 24(4):632–638

Hariharan P, Oreb BF, Brown N (1983) Applied Optics 22(6):876

Harvey JE, MacFarlane MJ, Forgham JL (1985) Design and performance of ranging telescopes: Monolithic vs. synthetic aperture. Optical Engineering 24(1):183–188

Hausler G, Maul M (1985) Telecentric scanner for 3D sensing. Optical Engineering 24(6):978–980

Heikkinen T, Ahola R, Manninen M, Myllyla R (1986) Recent results of the performance analysis of a 3D sensor based on time of flight. Proceedings SPIE Quebec International Symposium on Optical and Optoelectronic Applied Sciences and Engineering.

Hersman M, Goodwin F, Kenyon S, Slotwinski A (1987) Coherent laser radar application to 3D vision and metrology. Proceedings Vision'87 Conference 3-1–3-12

Holland SW, Rossol L, Ward MR (1979) Consight-1: A vision controlled robot system for transferring parts from belt conveyors. In: Computer Vision and Sensor-Based Robots G.G. Dodd and L. Rossol (Eds.), Plenum Press, New York, pp. 81–97

Horn BKP (1968) Focusing. MIT, Project MAC, AI Memo 160

Hulsmeyer C (1904) Hertzian wave projecting and receiving apparatus adapted to indicate or give warning of the presence of a metallic body, such as a ship or a train, in the line of projection of such waves. U.K. Patent 13,170

HYMARC (1987) Product information. Ottawa, Ontario Canada

Idesawa M, Yatagai Y, Soma T (1976) A method for the automatic measurement of 3D shapes by new type of moire topography. Proceedings 3rd International Conference Pattern Recognition: 708

Idesawa M, Yatagai Y, Soma T (1977) Scanning moire method and automatic measurement of 3D shapes. Applied Optics 16(8):2152–2162

Idesawa M, Yatagai Y (1980) 3D shape input and processing by moire technique. Proceedings 5th International Conference Pattern Recognition, IEEE-CS: 1085–1090

Idesawa M, Kinoshita G (1986) New type of miniaturized optical range sensing methods RORS and RORST. Journal of Robotic Systems 3(2):165–181

Inokuchi S, Sato K, Matsuda F (1984) Range imaging system for 3-D object recognition. Proceedings 7th International Conference Pattern Recognition: 806–808

Jalkio J, Kim R, Case S (1985) 3D inspection using multistripe structured light. Optical Engineering 24(6):966–974

Jalkio J, Kim R, Case S (1986) Triangulation based range sensor design. Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision, 728:132–146

Jarvis RA (1976) Focus optimization criteria for computer image processing. Microscope 24(2):163–180

Jarvis RA (1982) Computer vision and robotics laboratory. IEEE Computer 15(6):9–23

Jarvis RA (1983a) A laser time-of-flight range scanner for robotic vision. IEEE Transactions Pattern Analysis Machine Intelligence PAMI-5, 5:505–512

Jarvis RA (1983b) A perspective on range finding techniques for computer vision. IEEE Transactions Pattern Analysis Machine Intelligence PAMI-5, 2:122–139

Jelalian AV, McManus RG (1977) AGARD Panel Proceeding No. 77. June, Sec. 2.1, pp 1–21

Johnson M (1985) Fiber displacement sensors for metrology and control. Optical Engineering 24(6):961–965

Kak AC (1985) Depth perception for robot vision. In: Handbook of Industrial Robotics, S. Nof (Ed.) Wiley, New York, pp 272–319

Kanade T, Asada H (1981) Noncontact visual 3D range-finding devices. In: Proceedings SPIE 3D Machine Perception, B.R. Altschuler (Ed.):48–53

Kanade T, Fuhrman M (1985) A noncontact optical proximity sensor for measuring surface shape. In: Three-Dimensional Machine Vision, T. Kanade (Ed.), Kluwer Academic Boston, pp 151–194

Karara HM (1985) Close-range photogrammetry: where are we and where are we heading? Photogrammetric Engineering and Remote Sensing 51(5):537–544

Kawata H, Endo H, Eto Y (1985) A study of laser radar. Proceedings 10th International Technical Conference on Experimental Safety Vehicles

Kellogg WN (1961) Porpoises and sonar. University of Chicago Press, Chicago, IL

Kern Instruments (1987) Product Information. Gottwald, R. and Berner, W., The new Kern system for positioning and automated coordinate evaluation; advanced technology for automated 3D coordinate determination. Brewster, NY, and Aarau, Switzerland

Keyes RJ (1986) Heterodyne and nonheterodyne laser transceivers. Review of Scientific Instrumentation 57(4):519–528

Khetan RP (1975) The theory and application of projection moire methods. Ph.D. dissertation. Dept. of Engineering Mechanics, State University of New York, Stony Brook

Kingslake R (1983) Optical system design, Academic Press, New York

Kinoshita G, Idesawa M, Naomi S (1986) Robotic range sensor with projection of bright ring pattern. Journal of Robotic Systems 3(3):249–257

Koenderink JJ, Van Doorn AJ (1986) Dynamic shape. Biological Cybernetics 53:383–396

Kratky V (1979) Real-time photogrammetric support of dynamic 3D control. Photogrammetric Engineering and Remote Sensing 45(9):1231–1242

Krotkov EP (1986) Focusing. Ph.D. Dissertation, U. Penn, Phila, PA

Krotkov E, Martin JP (1986) Range from focus. Proceedings IEEE International Conference on Robotics and Automation, IEEE-CS: 1093–1098

Kurahashi A, Adachi M, Idesawa M (1986) A prototype of optical proximity sensor based on RORS. Journal of Robotic Systems 3(2):183–190

Labuz J, McVey ES (1986) Camera and projector motion for range mapping. Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision 728:227–234

Lamy F, Liegeois C, Meyrueis P (1981) 3D automated pattern recognition using moire techniques. Proceedings SPIE 360:345–351

Landman MM, Robertson SJ (1986) A flexible industrial system for automated 3D inspection. Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision, 728:203–209

Laser Technology (1986) Product information. Norristown, PA

Leader JC (1986) Speckle effects on coherent laser radar detection efficiency. Optical Engineering 25(5):644–650

Leger JR, Snyder MA (1984) Real-time depth measurement and display using Fresnel diffraction and white-light processing. Applied Optics 23(10):1655–1670

Leith E, Upatnieks J (1962) Reconstructed wavefronts and communication theory. Journal of Optical Society America 54:1123–1130

Lewis RA, Johnston AR (1977) A scanning laser rangefinder for a robotic vehicle. Proceedings 5th International Joint Conference on Artificial Intelligence: 762–768

Lewis JRT, Sopwith T (1986) 3D surface measurement by microcomputer. Image and Vision Computing 4(3):159–166

Livingstone FR, Tulai AF, Thomas MR (1987) Application of 3-D vision to the measurement of marine propellers. Proceedings Vision'87 Conference: 10-25–10-36

Livingstone FR, Rioux M (1986) Development of a large field of view 3D vision system. Proceedings SPIE 665

Lord Rayleigh (JW Strutt) (1874) On the manufacture and theory of diffraction gratings. Phil. Mag. 47(81):193

Lord Rayleigh (JW Strutt) (1881) Phil. Mag. 11:196

Lorenz RD (1984) Theory and design of optical/electronic probes for high performance measurement of parts. Ph.D. dissertation, Univ. of Wisconsin-Madison

Lorenz RD (1986) A novel, high-range-to-resolution ratio, optical sensing technique for high speed surface geometry measurements. Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision 728:152–146

Macy WW (1983) Two-dimensional fringe pattern analysis. Applied Optics 22(22):3893–3901

Mader DL (1985) Holographic interferometry of pipes: precision interpretation by least squares fitting. Applied Optics 24(22):3784–3790

Marshall G (1985) Laser Beam Scanning, Marcel-Dekker, New York

Matsuda R (1986) Multifunctional optical proximity sensor using phase modulation. Journal of Robotic Systems 3(2):137–147

Matthews HJ, Hamilton DK, Sheppard CJR (1986) Surface profiling by phase-locked interferometry. Applied Optics 25(14):2372–2374

Mersch SH, Doles JE (1985) Cylindrical optics applied to machine vision. Proceedings Vision'85 Conference, SME, 4-53–4-63

Mertz L (1983) Real-time fringe pattern analysis. Applied Optics 22(10):1535–1539

Miller GL, Wagner ER (1987) An optical rangefinder for autonomous robot cart navigation. Proceedings SPIE Industrial Electronics, Cambridge, MA, (November)

Moore DT, Traux BE (1979) Phase-locked moire fringe analysis for automated contouring of diffuse surfaces. Applied Optics 18(1):91–96

Mundy JL, Porter GB (1986) A three-dimensional sensor based on structured light. In: Three-Dimensional Machine Vision, T. Kanade (Ed.), Kluwer Academic, Boston, pp 3–62

Nakagawa Y, Ninomiya T (1987) Three-dimensional vision systems using the structured light method for inspecting solder joints and assembly robots. Three-Dimensional Machine Vision, T. Kanade (Ed.), Kluwer Academic, Boston, pp 543–565

Nevatia R, Binford TO (1973) Structured descriptions of complex objects. Proceedings 3rd International Joint Conference on Artificial Intelligence: 641–647

Newport Corp (1987) Product Information. Design and testing with holography. Machine vision components. Fountain Valley, CA

Nitzan D, Brain AE, Duda RO (1977) The measurement and use of registered reflectance and range data in scene analysis. Proceedings IEEE 65(2):206–220.

Nitzan D, Bolles R, Kremers J, Mulgaonkar P (May 1986) 3D vision for robot applications. NATO Workshop on Knowledge Engineering for Robotic Applications, Maratea, Italy

Oboshi T (1976) Three-Dimensional Imaging Techniques. Academic Press, New York

Oster G (1965) Moire optics: a bibliography. Journal of Optical Society America 55:1329

Ozeki O, Nakano T, Yamamoto S (1986) Real-time range measurement device for 3D object recognition. IEEE Trans. Pattern Analysis Machine Intelligence. PAMI-8, 4, 550–553

Pantzer D, Politch J, Ek L (1986) Heterodyne profiling instrument for the angstrom region. Applied Optics 25(22):4168–4172

Parthasarathy S, Birk J, Dessimoz J (1982) Laser rangefinder for robot control and inspection. Proceedings SPIE Robot Vision 336:2–11

Pelowski KR (1986) 3D measurement with machine vision. Proceedings Vision'86 Conference: 2-17–2-31

Pentland AP (1987) A new sense of depth of field. IEEE Transaction Pattern Analysis Machine Intelligence PAMI-9, 4:523–531

Perceptron (1987) Product information. Farmington Hills, MI

Perrin JC, Thomas A (1979) Electronic processing of moire fringes: application to moire topography and comparison with photogrammetry. Applied Optics 18(4):563–574

Peterson RW, Robinson GM, Carlsen RA, Englund CD, Moran PJ, Wirth WM (1984) Interferometric measurements of the surface profile of moving samples. Applied Optics 23(10):1464–1466

Photonic Automation, Inc. (1987) Product literature. Improving automated SMT inspection with 3D vision. M. Juha and J. Donahue. Santa Ana, CA

Pipitone FJ, Marshall TG (1983) A wide-field scanning triangulation rangefinder for machine vision. International Journal of Robotics Research 2(1):39–49

Pinckney HFL (1978) Theory and development of an on-line 30 Hz video photogrammetry system for real-time 3D control. International Archives of Photogrammetry, Vol. XXII, Part V. 2:38 pages

Pirodda L (1982) Shadow and projection moire techniques for absolute and relative mapping of surface shapes. Optical Engineering 21:640

Popplestone RJ, Brown CM, Ambler AP, Crawford GF (1975) Forming models of plane-and-cylinder faceted bodies from light stripes. Proceedings 4th International Joint Conference on Artificial Intelligence: 664–668

Potmesil M (1983) Generating models of solid objects by matching 3D surface segments. Proceedings 8th International Joint Conference on Artificial Intelligence: 1089–1093

Potsdamer J, Altschuler M (1982) Surface measurement by space-encoded projected beam system. Computer Graphics Image Processing 18:1–17

Pryputniewicz RJ (1985) Heterodyne holography applications in studies of small components. Optical Engineering 24(5):849–854

Quist TM, Bicknell WE, Bates DA (1978) ARPA Semiannual report: optics research, Lincoln Laboratory, MIT

Reid GT (1986) Automatic fringe pattern analysis: A review. Optics and Lasers in Engineering 7:37–68

Rioux M (1984) Laser range finder based upon synchronized scanners. Applied Optics 23(21):3837–3844

Rioux M, Blais F (1986) Compact 3-D camera for robotic applications. Journal of Optical Society of America A 3(9):1518–1521

Robotic Vision Systems, Inc (1987) Product literature. RVSI, Hauppage, NY

Rocker F (1974) Localization and classification of 3D objects. Proceedings 2nd International Conference Pattern Recognition: 527–528

Rosenfeld JP, Tsikos CJ (1986) High-speed space encoding projector for 3D imaging. Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision, 728:146–151

Ross JA (1978) Methods and systems for 3D measurement. U.S. Patent 4,199,253. (RVSI, Hauppage, NY)

Sampson RE (1987) 3D range sensor via phase shift detection (Insert). IEEE Computer 20(8):23–24

Sasaki O, Okazaki H (1986) Sinusoidal phase modulating interferometry for surface profile measurement. And Analysis of measurement accuracy in sinusoidal phase modulating interferometry. Applied Optics 25(18): 3137–3140,3152–3158

Sato Y, Kitagawa H, Fujita H (1982) Shape measurement of curved objects using multiple-slit ray projection. IEEE Transactions Pattern Analysis Machine Intelligence PAMI-4, 6:641–649

Sato K, Inokuchi S (1985) 3D surface measurement by space encoding range imaging. Journal of Robotic Systems 2(1):27–39

Schewe H, Forstner W (1986) The program PALM for automatic line and surface measurement using image matching techniques. Proceedings Symposium International Society for Photogrammetry and Remote Sensing, Vol. 26, Part 3/2:608–622

Schlag JF, Sanderson AC, Neumann CP, Wimberly FC (1983) Implementation of automatic focusing algorithms for a computer vision system with camera control. CMU-RI-TR-83-14

Schmitt F, Maitre H, Clainchard A, Lopez-Krahm J (1985) Acquisition and representation of real object surface data. SPIE Proceedings Biostereometrics Conf., Vol. 602

Schmitt F, Barsky B, Du W (1986) An adaptive subdivision method for surface-fitting from sampled data. Computer Graphics 20(4):179–188

Schuman W, Dubas M (1979) Holographic Interferometry. Springer-Verlag, Berlin

Schwartz J (1983) Structured light sensors for 3D robot vision. Robotics Research Report No. 8, Courant Institute of Mathematical Sciences, New York University, New York

Sciammarella CA (1982) The moire method—A review. Exp. Mech. 22:418–433

SELCOM (1987) Optocator product information. Valdese, NC, US; Partille, Sweden; Krefeld, West Germany

Servo-Robot (1987) Product information. Boucherville, Quebec, Canada

Shirai Y, Suwa (1972) Recognition of polyhedra with a range finder, Pattern Recognition 4:243–250

Silvaggi C, Luk F, North W (1986) Position/dimension by structured light. Experimental Techniques: 22–25

Skolnick MI (1962) Introduction to Radar Systems. McGraw-Hill, New York

Slevogt H (1974) Technische Optik. Walter de Gruyter, Berlin, pp 55–57

Smith RC, Cheeseman P (1987) On the representation and estimation of spatial uncertainty. International Journal of Robotics Res. 5(4):56–68

Solid Photography, Inc. (1977) (now Robotic Vision Systems, Inc. (RVSI), Hauppage, NY)

Srinivasan V, Liu HC, Halioua M (1985) Automated phase measuring profilometry: A phase-mapping approach. Applied Optics 24(2):185–188

Stockman G, Hu G (1986) Sensing 3D surface patches using a projected grid. Proceedings Computer Vision Pattern Recognition Conference: 602–607

Strand T (1983) Optical three-dimensional sensing. Optical Engineering 24(1):33–40

Svetkoff DJ (Oct. 1986) Towards a high resolution, video rate, 3D sensor for machine vision. Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision, 728:216–226

Svetkoff DJ, Leonard PF, Sampson RE, Jain RC (1984) Techniques for real-time feature extraction using range information. Proceedings SPIE—Intelligent Robotics and Computer Vision 521:302–309

Talbot H (1836) Facts relating to optical science No. IV. Phil. Mag. 9:401–407

Technical Arts Corp (1987) Product Literature. Redmond, WA

Teich MC (1968) Infrared heterodyne detection. Proceedings IEEE 56(1):37–46

Tenenbaum J (1970) Accommodation in computer vision. Ph.D. dissertation, Stanford University, Stanford, CA

Terras R (1986) Detection of phase in modulated optical signals subject to ideal Rayleigh fading. Journal of Optical Society of America A 3(11):1816–1825

Thalmann R, Dandliker R (1985) Holographic contouring using electronic phase measurement. Optical Engineering 24(6):930–935

Theocaris PS (1969) Moire fringes in strain analysis. Pergamon Press, New York

Tozer BA, Glanville R, Gordon AL, Little MJ, Webster JM, Wright DG (1985) Holography applied to inspection and measurement in an industrial environment. Optical Engineering 24(5):746–753

Tsai R (1986) An efficient and accurate camera calibration technique for 3D machine vision. Proceedings Computer Vision Pattern Recognition Conference IEEE-CS:364–374

Vest CM (1979) Holographic interferometry. Wiley, New York

Vuylsteke P, Oosterlinck A (1986) 3D perception with a single binary coded illumination pattern. Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision, 728:195–202

Wagner JW (1986) Heterodyne holographic interferome-

try for high-resolution 3D sensing. Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision, 728:173–182

Wagner JF (June 1987) Sensors for dimensional measurement. Proceedings Vision'87 Conference, pp 13-1–13-18

Wang JY (1984) Detection efficiency of coherent optical radar. Applied Optics 23(19):3421–3427

Wang JY, Bartholomew BJ, Streiff ML, Starr EF (1984) Imaging CO_2 radar field tests. Applied Optics 23(15):2565–2571

Wang JY (1986) Lidar signal fluctuations caused by beam translation and scan. Applied Optics 25(17):2878–2885

Wang YE, Mitiche A, Aggarwal JK (1985) Inferring local surface orientation with the aid of grid coding. IEEE Workshop on Computer Vision: Representation and Control: pp 96–104

Wei D, Gini M (1983) The use of taper light beam for object recognition. In: Robot Vision, R. Pugh (Ed.), IFS Publications, Springer-Verlag, Berlin

Will PM, Pennington KS (1972) Grid coding: a novel technique for image processing. Proceedings IEEE 60(6):669–680

Winthrop JT, Worthington CR (1965) Theory of fresnel images I. Plane periodic objects in monochromatic light. Journal of the Optical Society of America 55(4):373–381

Wuerker RF, Hill DA (1985) Holographic microscopy. Optical Engineering 24(3):480–484

Yamamoto H, Sato K, Inokuchi S (1986) Range imaging system based on binary image accumulation. Proceedings International Conference on Pattern Recognition IEEE:233–235

Yatagai T, Idesawa M, Yamaashi Y, Suzuki M (1982) Interactive fringe analysis system: Applications to moire contourogram and interferogram. Optical Engineering 21(5):901

Yeung KK, Lawrence PD (1986) A low-cost 3D vision system using space-encoded spot projections. Proceedings SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision, 728:160–172

Zuk DM, Dell'Eva ML (1983) Three-dimensional vision system for the adaptive suspension vehicle. Final Report No. 170400-3-F, ERIM, DARPA 4468, Defense Supply Service-Washington

Note added in proofs: Several uncited references have been included to provide a more complete bibliography.

Appendix: Eye Safety

Lasers are used in all types of active optical range imaging sensors. When people are exposed to laser radiation, eye safety is critical. An understanding of eye safety issues is important to the range imaging applications engineer.

Concerning the *sale of laser products* across state lines in the United States, vendors of end-user equipment containing lasers must comply with the requirements of the Food and Drug Administration's Center for Devices and Radiological Health (CDRH). Concerning the *use of laser products*, most organizations follow the ANSI Z136.1 Standard regulations. The ANSI and CDRH regulations are essentially the same except for some fine points. A simplified version of the regulations is given below. The applications engineer should consult the CDRH (1985) regulations or the ANSI (1986) regulations for complete details.

Lasers emit electromagnetic radiation that is either visible (light) or invisible (infrared or ultraviolet). When laser radiation is received by the human eye, damage may occur in the retina or the cornea depending upon the wavelength if the radiation levels exceed the maximum permissible exposure. Visible light regulations are different from invisible regulations because of the aversion response. People will blink or look away in less than 0.25 s when exposed to intense visible radiation. With invisible radiation, no such aversion response occurs although broad spectrum near-infrared laser diodes are visible to many people. Although not listed separately in official documents, the regulations may be viewed as two distinct sets of safety classes, one set for visible and another set for invisible. Within each class, two requirements must be met: (1) the average power through a standard aperture (usually 7-mm dia) must be less than the maximum average power for that class laser at every point in the field of view of the laser, and (2) the energy in any pulse received by the standard aperture must be less than the maximum energy level for that class. One subtlety important to range-imaging sensors is that the pulse repetition frequency (PRF) factor is included in ANSI regulations, but not in CDRH regulations.

For visible light (400–700 nm wavelengths), there are really five classes of lasers. Actual ratings are wavelength dependent, but the following list gives a reasonable indication of allowable average powers through the standard 7 mm diameter aperture with a 5 diopter lens.

- Class I (No Risk, Eye Safe)
  Average Power $< 0.4\ \mu W$.
- Class II (Low Power, Caution)
  $0.4\ \mu W <$ Average Power < 1 mW
- Class IIIa (Medium Low Power, Caution)
  1mW $<$ Average Power $<$ 5mW
- Class IIIb (Medium Power, Danger)
  5 mW $<$ Average Power $<$ 500 mW
- Class IV (High Power, Danger)
  Average Power $>$ 500 mW

Pulse requirements are more complicated and must be computed from equations and tables listed in CDRH and ANSI regulations based on wavelength.

For invisible lasers (UV:200–400 nm, IR:700 nm–1 mm wavelengths), there are three classes:

- Class I (No Risk, Eye Safe)
  Wavelength-Dependent Regulations
- Class IIIb (Medium Power, Danger)
  Average Power < 500 mW, Not Class I
- Class IV (High Power, Danger)
  Average Power > 500 mW

Again, pulse requirements must be computed from equations and tables listed in CDRH and ANSI regulations based on wavelength. There are no low or medium low power categories here. However, ANSI regulations vary slightly from CDRH regulations in that they allow a Class IIIa (Caution) for infrared lasers with powers that exceed the Class I limit by less than a factor of five (Sec. 3.3.3.2, ANZI Z136.1, 1986).

About the Authors

PAUL J. BESL graduated *summa cum laude* in physics from Princeton University in 1978 and received the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of Michigan, Ann Arbor, in 1981 and 1986, respectively. In 1987, he received a Rackham Distinguished Dissertation Award from the University of Michigan for his dissertation on range image understanding.

From 1979 to 1981, he did computer simulations for Bendix Aerospace Systems in Ann Arbor, MI, and from 1981 to 1983, worked on the Geomod solid modeling system at Structural Dynamics Research Corp. in Cincinnati, OH. Since 1986, he has been a research scientist at General Motors Research Laboratories in Warren, MI, where his primary research interest is computer vision, especially range image analysis and geometric modeling for image understanding. Dr. Besl is a member of the Institute of Electrical and Electronics Engineers, the Association for Computing Machinery, the American Association for Artificial Intelligence, and the Machine Vision Association of the Society of Manufacturing Engineers.

YORAM BRESLER received the B.Sc. and M.Sc. degrees from the Technion, Israel Institute of Technology, Haifa, Israel in 1974 and 1981, respectively, and the Ph.D. degree from Stanford University, Stanford, CA, in 1985, all in electrical engineering.

From 1974 to 1979 he served as an electronics engineer in the Israeli Defense Force. From 1979 to 1981 he worked on algorithms for autonomous TV aircraft guidance at the Flight Control Lab, Department of Aeronautical Engineering, Technion, Israel. From 1985 to 1987, he was a Research Associate at the Information Systems Laboratory at Stanford University, where his research involved array signal processing and medical imaging. Since 1987, he has been an Assistant Professor at the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. Dr. Bresler's research interests include statistical methods in imaging and in array signal processing.

JEFFREY A. FESSLER received the B.S. degree from Purdue University, West Lafayette, IN, in 1985 and the M.S. degree from Stanford University, Stanford, CA, in 1986, both in electrical engineering.

Since 1986, he has been a National Science Foundation Graduate Fellow at Stanford, working toward the Ph.D. degree in electrical engineering. His research interests include estimation theory and its applications to medical imaging and bioengineering.

LESTER A. GERHARDT holds a joint professorship with RPI's Electrical, Computer, and Systems Engineering and Computer Science Departments and is Director of the RPI CIM Program.

WILLIAM I. KWAK is a research assistant with the RPI CIM Program and a Ph.D. candidate with the RPI Electrical, Computer, and Systems Engineering Department.

ALBERT MACOVSKI received the B.E.E. degree from C.C.N.Y. in 1950, the M.E.E. degree from the Polytechnic Institute of Brooklyn in 1953, and the Ph.D. degree in electrical engineering from Stanford University in 1968.

From 1950–1957 Dr. Macovski was a member of the technical staff at R.C.A. Laboratories. From 1957–1960 he was an Assistant and then Associate Professor at the Polytechnic Institute of Brooklyn, NY. From 1960–1971 Dr. Macovski was a staff scientist at the Stanford Research Institute. Following one year as a special NIH fellow at the U.C. Medical Center in San Francisco, he joined the Stanford Faculty as an Adjunct Professor and then full Professor of Electrical Engineering and Radiology, his present position.

Dr. Macovski's research has been in a variety of imaging systems including television, facsimile, holography, and interferometry. During the past five years he has been particularly involved with diagnostic imaging techniques in ultrasound, radiography, and magnetic resonance.

Dr. Macovski is a Fellow of the Institute of Electrical and Electronics Engineers and Optical Society of America. He is a member of the American Association of Physicists in Medicine, Eta Kappa Nu, and Sigma Xi. In 1958 he received the award from the IRE professional group on broadcast and television receivers. In 1973 he received the IEEE Zworykin Award.

Dr. Macovski has approximately 150 issued U.S. Patents and 150 publications in the fields of optics, electronics, television, imaging, radiography, and ultrasonics.

CHARLES B. MALLOCH is a resident engineer from United Technologies Corporation with the RPI Computer Integrated Manufacturing Program and a Ph.D. candidate with the RPI Electrical, Computer, and Systems Engineering Department.

RICHARD A. ROBB received the Ph.D. degree in computer science and biophysics from the University of Utah in 1971. He is Professor of Biophysics in the Mayo Graduate School of Medicine and Director of the Mayo Biotechnology Computer Resource. He has been involved in the development and application of computer systems for processing, analysis, and display of biomedical image data for over fifteen years. He is principal investigator on several NIH research grants and has over 150 publications in the field of multi-dimensional biomedical image processing. He is one of the developers of the Dynamic Spatial Reconstructor, an advanced high temporal resolution 3-D image scanner at the Mayo Clinic. His current interests are in development of comprehensive, efficient workstations and networks for biomedical image analysis.

Chapter 2: Segmentation

An image must be analyzed and the features it contains extracted before more abstract representations and descriptions are generated. Careful selection of algorithms for these so-called "low-level-vision" operations is critical for the success of higher level scene interpretation algorithms. One of the first operations that a computer vision system must perform is the separation of objects from the background. This operation, commonly called "segmentation," is approached as either

- An edge-based method for locating discontinuities in certain properties in the image or
- A region-based method of grouping of pixels according to certain similarities.

Success of a segmentation technique is measured by the utility of descriptions of resulting objects. Although many segmentation techniques have been proposed, a general solution still eludes researchers. By introducing enough additional knowledge about a particular application domain, the problem may be solvable. That this additional knowledge is necessary is hardly surprising, since philosophers and psychologists have known for centuries that what we sense when we view an object is a very minor fraction of the information required to see that object. For example, in Figure 2.1, most people do not see the dalmatian dog.

Figure 2.1: A difficult image for segmentation. Can you separate the object from the background?

In an edge-based approach, the boundaries of objects are used to partition an image. Points that lie on the boundaries of an object must be marked. Such points, called "edge points," can often be detected by analyzing the neighborhood of the point. By definition, the regions on either side of an edge point (i.e., the object and the background) have dissimilar characteristics. Thus, in edge detection, the emphasis is on detecting dissimilarities in the neighborhoods of points. Most edge detectors use only intensity characteristics; however, more sophisticated characteristics, which can be derived from intensity values such as texture and motion, may also be used.

In a region-based approach, all pixels corresponding to a single object are grouped together and are marked to indicate that they belong to the same object. The grouping of the points is based on some criterion that distinguishes points belonging to the same object from all other points. Two very important considerations in such a grouping are spatial proximity and intensity similarity. Without any domain-dependent information, we can assume that all points that belong to a given surface of an object will be spatially close and will have similar reflectance characteristics. Clearly, the second assumption is not satisfied in many real situations. But we can initially group points using these simple considerations, and then use domain-dependent knowledge to refine the results of this first grouping operation. Complex images may require the use of more rigorous techniques to perform the grouping operation.

In an ideal image, the edges of each object will define a closed region, and a region will be bounded by connected edge points. Theoretically, both edge-based and region-based approaches should give identical information. When an edged-based approach

is used, regions could be obtained from edges using a simple region-filling algorithm; similarly, when a region-based approach is used, the edges could be obtained by using a simple boundary follower. Unfortunately, in real images, correct edges can rarely be obtained from region information, and vice versa. Due to noise and other factors, neither the dissimilarity measures used by edge detectors nor the similarity measures used by region growers give anywhere near perfect results.

Edge detection techniques are described next, followed by region-growing techniques. Finally, other methods for segmentation — methods that use features such as texture and color — are not described, but references to literature are given. Techniques for representation and description of regions and boundaries, feature extraction, and matching are described in Chapter 3.

Edge-based segmentation

Edge detection has been a very active research area for about two decades, and many edge detectors have been developed. Several types of edges are commonly found in an image. One-dimensional profiles of some of the common edge types are shown in Figure 2.2. The step edge is a good model only in those situations in which (1) objects and the background have very good contrast and (2) the intensity changes at the boundaries are crisp. However, step edges are rare in most real images; rather, ramp edges are common. Because of low-frequency components or the smoothing introduced by most sensing devices, sharp discontinuities rarely exist in real signals. Roof edges and line edges may occur if narrow objects are present in the scene. A line usually becomes a roof in the sensed image.

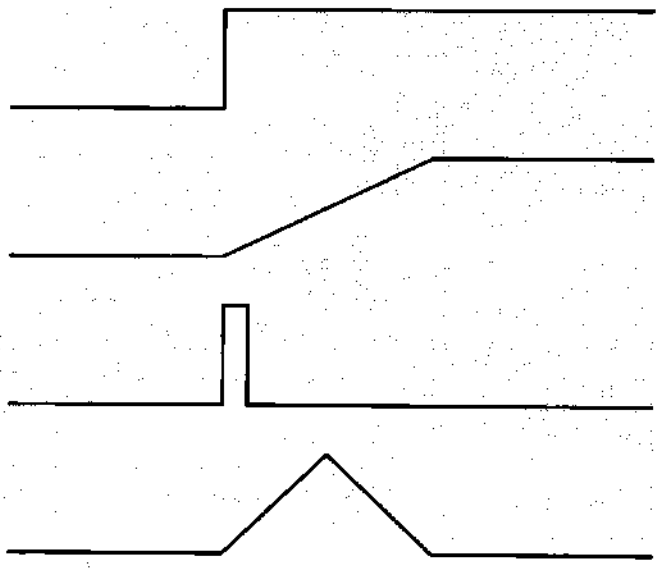


Figure 2.2: One-dimensional profile of edges

Gradient. Edges in an image are detected by computing the gradient at each pixel and then identifying those pixels with gradient magnitudes larger than a threshold. The magnitude of the gradient is used to represent the strength of the edge at each point and the direction of the gradient is used to link points belonging to the same physical edge. The magnitude $E(x,y)$ and the direction $\theta(x,y)$ of the edge at pixel (x,y) in terms of the first partial derivatives of intensities E_x and E_y (along the x- and y-directions) are given by the following equations:

$$E(x,y) = \sqrt{E_x^2(x,y) + E_y^2(x,y)} \tag{2.1}$$

$$\theta = \tan^{-1}\left(\frac{E_y}{E_x}\right) \tag{2.2}$$

In most applications, the following computationally simpler expressions are used to approximate the gradient:

$$E(x,y) = |E_x| + |E_y|$$

or

$$E(x,y) = Max\,(|E_x|, |E_y|) \tag{2.3}$$

Many approximations have been used to compute the partial derivatives in the x- and y-directions. One of the earliest edge detectors used, called "Roberts' cross operator," computes the partial derivatives by taking the difference in intensities of the diagonal pixels in a two-by-two pixel window. One of the most commonly used approximations, defined over a three-by-three window, is attributable to Sobel.[22] Using the following notation to identify neighboring pixels:

A_0	A_1	A_2
A_7	f(x,y)	A_3
A_6	A_5	A_4

this approximation is given by

$$\begin{aligned} E_x &= (A_2 + 2A_3 + A_4) - (A_0 + 2A_7 + A_6) \\ E_y &= (A_0 + 2A_1 + A_2) - (A_6 + 2A_5 + A_4) \end{aligned} \tag{2.4}$$

Occasionally, operators that compute the edge strength in a number of different directions are used in computer vision. These directional operators are defined over three-by-three or larger neighborhoods.[23]

Laplacian. The operators we have just discussed require that E_x and E_y be computed separately and then combined to obtain the edge strength. The Laplacian operator, which responds to changes in intensity in an image, requires much less computation. The Laplacian is defined in terms of the second partial derivatives, as follows:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{2.5}$$

The second derivatives along the x- and y-directions are approximated using the difference equations

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &= (f(x+1,y) - f(x,y)) - (f(x,y) - f(x-1,y)) \\ &= (f(x+1,y) - 2f(x,y) + f(x-1,y)) \end{aligned} \tag{2.6}$$

and

$$\frac{\partial^2 f}{\partial y^2} = (f(x,y+1) - 2f(x,y) + f(x,y-1)) \tag{2.7}$$

Thus, the following three-by-three mask can be used to compute the Laplacian:

0	1	0
1	−4	1
0	1	0

The response of the Laplacian operator can be computed in one step by convolving the image with this mask. Nevertheless, since the Laplacian is an approximation to the second derivative, it reacts more strongly to lines, line ends, and noise pixels than it does to edges, and it generates two responses to each edge — one on either side of the edge. However, the zero-crossings in the output of the Laplacian are useful to localize the edges.

Every edge detector that uses the neighborhood of a point to compute edgeness is prone to responding incorrectly to intensity changes that are attributable to noisy pixels. This problem with such edge detectors occurs because they compute the property of a point, while an edge is actually a property of a region. Edge points of interest are not isolated points, but rather segments of boundaries between regions. Several suggestions have been made to include characteristics of regions in edge detection.

Multiresolution edge detectors. A multiresolution detector detects edges by computing characteristics of the areas of different sizes on either side of a point. In the simplest case, the edge strength may be defined as the difference in the average intensities of the areas on either side of a point. For small neighborhoods, such a detector gives high values close to the actual edge point. For larger neighborhoods, major edges are detected, including texture edges. Many variations on this approach to edge detection exist. That intensity changes in an image may occur at several levels is a very important implicit assumption in this approach. If we are interested in the detection of edges due to disparate physical phenomena in an image, we may need operators of different sizes. Also, several neighborhoods of a point can be considered and their characteristics combined to detect edges.

Natural images contain assorted objects of different sizes; the intensity changes in these images occur over a wide range of scales. Selection of appropriate filters allows an edge detector to respond to edges at different scales. The Marr and Hildreth[24] edge detector is an example of an edge detector in which different smoothing filters are applied to an image at various resolutions and smoothed by different amounts. Then, intensity changes are detected in each smoothed image. Intensity changes at a given resolution will indicate edges that are significant at that level of resolution. The intensity changes detected at multiple levels are combined to allow edges attributable to physical changes in a scene to be detected.

A Gaussian filter is used to average an image. The use of a Gaussian filter, as opposed to some other filter, results in good spatial- and spectral-localization characteristics; it is the only filter that has this capability. The Gaussian filter, excluding the constant multiplicative factor, is given by

$$G(x,y) = e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \tag{2.8}$$

In many edge detection methods, if the first derivative of the image is above a certain threshold, an edge point is assumed. The result is too many edge points. A better approach is to find the image gradient and to consider as edge points only those points with a gradient value that is a local maximum. Then, a peak will be in the first derivative and, equivalently, a zero-crossing will be in the second derivative at edge points. Thus, edge points are detected by finding the zero-crossings of the second derivative of image intensity.

A first step in edge detection is convolution with a Gaussian filter. This smooths the image by removing structures that are smaller than the Gaussian window. Isolated noise points and small structures can be filtered out, making the response of the edge detector more reliable. Since the filtering will result in the blurring of edges, the edge detector only considers to be edge points those points that have a locally maximum gradient. This is accomplished by taking the zero-crossing of the second derivative. The second derivative in two dimensions is approximated by the Laplacian. Thus, zero-crossings are detected in $F(x,y)$, where $F(x,y)$ is given by

$$F(x,y) = \nabla^2(G(x,y) * f(x,y)) \tag{2.9}$$

where ∇^2 and * denote the Laplacian and convolution operators, respectively. Since the Laplacian is an isotropic operator, it will detect edges in all directions.

The two operations — Gaussian smoothing and computation of the Laplacian — can be combined into a single operation: convolving the image with a Laplacian of Gaussian (LOG) function, which is given by

$$\nabla^2 G(x,y) = \left(\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right) e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \tag{2.10}$$

The cross section of the LOG function (inverted) is shown in Figure 2.3. For discrete implementation of this operator, the size of the mask depends on the value of σ. Clearly, the performance of the operator also depends on the value of σ.

In the approach just described, edges are detected at a particular resolution. Determination of the real edges in an image may require that information from operators of several sizes be combined. Real edges in an image are due to a discontinuity in a physical variable in a scene; such edges should survive for more than one resolution of the operator.

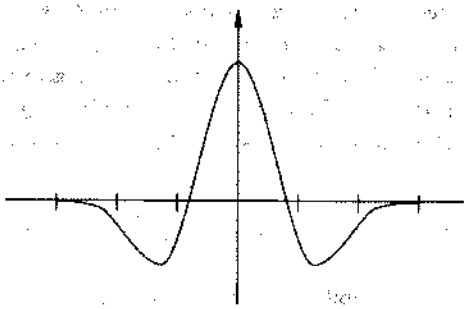


Figure 2.3: One-dimensional profile of Laplacian of Gaussian Operator

Witkin[25] applied the zero-crossing detector using different values of σ and created a scale-space. Such a scale-space for the one-dimensional function *f(x)* is shown in Figure 2.4,[25] where the horizontal axis represents *x*, the vertical axis represents σ, and the lines are contours of zero-crossings. Fewer edges are detected at larger values of σ, as can be seen by the decrease in the number of zero-crossings. The presence of edges at a particular resolution is found at the appropriate scale, and the exact location of edges is obtained by following the corresponding contours through lower values of σ. No new zero-crossings are created as σ is increased; this is a desirable property of the Gaussian smoothing function.

Figure 2.4: Typical contours of zero-crossings in scale-space (from Witkin).[25]

Using detection, localization, and uniqueness (single response to one edge) criteria, Canny[26] designed an edge detector that is optimum at any scale. He showed that the first derivative of the Gaussian is a good approximation to his optimal edge detector for detecting step edges. He presented a thresholding technique to eliminate streaking of edge contours and proposed a method for obtaining directional masks with oblong support.

The size of the operators causes a major problem in using multiresolution operators. The response of an operator at a point is the sum of the weighted responses at all points in its support. In an ideal situation, the response at a point is the result of the contributions from points on the two sides of the same edge point. Thus, it is desirable that only one edge pass through the support of the operator. However, in a complex image, as the size of the support increases, the number of edges in the support also increases, resulting in mutual influence of edges. This mutual influence may result in unpredictable behavior of detected edges: the edges may be dislocated, false edges may be generated, and real edges may be missing at different scale sizes.[27]

Surface fitting. Most ideas for analyzing images are relative to a continuous domain, while a digital image is actually a sampling of a continuous function of two-dimensional spatial variables. Desired properties must then be computed using discrete approximations of their continuous domain counterparts. If the continuous spatial function can be approximately reconstructed, then image properties can be more precisely computed. Computation of localization to subpixel accuracy may be possible. This concept leads to an interesting approach.

The intensity values in the neighborhood of a point can be used to obtain the underlying continuous intensity surface, which

then can be used as a best approximation from which properties in the neighborhood of the point can be computed. An image function can be described by

$$z = f(x,y) \tag{2.11}$$

That intensity values at a point satisfy some analytical characteristics is assumed. Clearly, if the image represents one surface, the preceding equation will correctly characterize the image. However, an image generally contains several surfaces, in which case this equation would be satisfied only at local surface points. In other words, the intensity values in the neighborhood of a point usually belong to one surface; hence, they should show the structure of the surface. This will not be true at surface discontinuities. The Haralick's[28] facet model of an image is the result of this idea. In the facet model, the neighborhood of a point is approximated by the cubic surface patch that best fits the area. Thus, the intensity values in the neighborhood of the point are approximated by

$$f(r,c) = k_1 + k_2 r + k_3 c + k_4 r^2 + k_5 rc + k_6 c^2 + k_7 r^3 + k_8 r^2 c + k_9 rc^2 + k_{10} c^3 \tag{2.12}$$

where r and c are coordinates, along the x- and y-directions, respectively, relative to the point in the image whose neighborhood is being approximated. The k_i coefficients in the preceding equation can be computed using the least squares method.

Edge points are relative extrema in the first directional derivative of the function that approximates the surface in the neighborhood of a pixel. This fact can be used to detect edge points. Relative extrema in the first derivative will result in a zero-crossing in the second derivative, occurring in the direction of the first derivative. If θ is the edge direction, the second directional derivative at a point (r_0,c_0) is given by

$$\begin{aligned} f''_{\theta}(r,c) &= 6(k_7 \sin^3\theta + k_8 \sin^2\theta\cos\theta + k_9 \sin\theta\cos^2\theta + k_{10}\cos^3\theta)\rho + 2(k_4 \sin^2\theta + k_5 \sin\theta\cos\theta + k_6 \cos^2\theta) \\ &= A\rho + B \end{aligned} \tag{2.13}$$

where $r_0 = \rho \sin\theta$ and $c_0 = \rho \cos\theta$. Thus, (r_0,c_0) is an edge point if, for some ρ, $|\rho| < \rho_0$ (where ρ_0 is the length of the side of a pixel)

$$\begin{aligned} &f''_{\theta}(r_0,c_0;\rho) = 0 \\ &\textit{and} \\ &f_{\theta}(r_0,c_0;\rho) \neq 0 \end{aligned} \tag{2.14}$$

Nalwa and Binford[29] used one-dimensional surfaces to detect edgels—short, linear edge elements. (A typical one-dimensional surface is shown in Figure 2.5.[29]) In Nalwa and Binford's method, the direction of the edgel is estimated initially by fitting a planar patch in a least squares sense; this estimate is refined using a one-dimensional cubic surface. Finally, a one-dimensional *tanh* surface is fitted, and the edgel characteristics are calculated.

Grimson and Pavlidis[30] used statistical properties of the differences between intensity values in a region and the planar approximation to detect discontinuities. If the differences are randomly distributed, then the underlying surface is assumed to be smooth; on the other hand, if the differences are distributed systematically, discontinuities are implied. Detecting discontinuities in an early stage permits more accurate surface fitting in a subsequent stage, since oscillations due to Gibb's phenomena are avoided.

Edge linking. Detection of edge pixels is only a part of the segmentation task. The outputs of typical edge detectors must be linked to form the boundaries of objects. Edge pixels seldom form closed boundaries; missed edge points will result in breaks in the boundaries. Thus, edge linking is an extremely important, but difficult, step in image segmentation.[31] Several methods have been suggested to improve the performance of edge detectors. In the relaxation-based edge-linking method, information from the neighborhood of an edge point is used to improve edge detector performance.[32,33] The basic idea behind this approach is to use the magnitude and direction of the edge at a point to detect other edges in the neighborhood. If the direction of an edge is compatible with its neighboring edges, the edges are linked; incompatible edges are removed. Looking at larger neighborhoods allows missing edges to be filled in. In a sequential edge detector — also called an "edge tracker" — the image is scanned to find a so-called "strong" edge point, which is considered to be a starting point for the boundary of an object. Starting at this point, all possible boundaries are grown by considering edge strength and directions. The tracking operation is computationally simplified by extending the boundary segment in a direction that depends on the current direction of the edge. Even very weak edge segments can be detected using this approach.

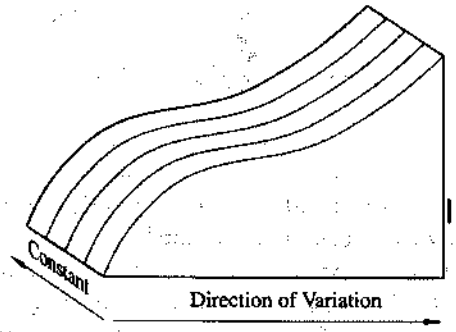


Figure 2.5: A one-dimensional surface (from Nalwa and Binford).[29]

Region-based segmentation

In region-based segmentation, points that have similar characteristics are identified and grouped into regions. Such points usually belong to a single object. A set of connected points that belong to the same object is called a "region." Several techniques are available for segmenting an image using this region-based approach.

Region formation. The segmentation process usually begins with a simple region formation step. In this step, intrinsic characteristics of the image are used to form initial regions. Thresholds obtained from a histogram of the image intensity are commonly used to perform this grouping. In general, an image will have several regions, each of which may have different intrinsic characteristics. In such cases, the intensity histogram of the image will show several peaks; each peak may correspond to one or more regions. Several thresholds are selected using these peaks. After thresholding, a connected-component algorithm can be used to find initial regions. This thresholding approach usually produces too many regions. Improvement can be obtained by performing some simple operations on the image to obtain better behaved histograms. One such operation uses the average of the differences between each pixel $f(x,y)$ and its neighbors. Let $g(x,y)$ represent this average. Then a new image is formed by multiplying h(x,y) and f(x,y), where

$$\begin{aligned} h(x,y) &= 1, \quad if\ g(x,y) < H \\ &= 0, \quad otherwise. \end{aligned} \tag{2.15}$$

where H is a threshold. Peak selection in the histogram of this new image is simplified because the contribution of border points of regions is removed from the histogram. Border points generally have intermediate values and tend to fill in the "valley" between two peaks in the histogram that correspond to the gray levels of two separate regions. Elimination of the border points results in a better behaved histogram, which produces better thresholds for determining the initial regions.

These thresholding techniques may still produce too many regions. Since they were formed based only on first-order characteristics, the regions obtained are usually simplistic and do not correspond to complete objects. The regions obtained via thresholding may be considered to be only the first stage in segmentation. After the initial histogram-based segmentation, more sophisticated techniques are used to refine the segmentation.

Functional approximation. The intensity values in an image may be treated as representing a three-dimensional surface. The aim of segmentation is to find or approximate regular surfaces. The many approaches to partitioning an image using this approach assume that a region Y satisfies a similarity predicate if, at every point in Y,

$$\left[f(x,y) - \sum_{k=1}^{M} a_k \cdot b_k(x,y) \right] \leq h \tag{2.16}$$

where h is a threshold, b_k represents a finite family of linearly independent functions and a_k represents parameters chosen to minimize the maximum pointwise error. Instead of this uniform approximation, the following least integral square error approximation can be used:

$$\sum_{Y}\left[f(x,y)-\sum_{k=1}^{M}a_k \cdot b_k(x,y)\right]^2 \leq \tau \tag{2.17}$$

where τ is a threshold. In the above model, if we select $M = 1$ and $b_1(x,y) = 1$, then the image is approximated using horizontal planes. This scheme then becomes the same as thresholding. In the last few years, this approach to image modeling, where the intensity value is considered to be a height, has received increasing attention. (The facet model for edge detection, discussed earlier, is based on this approach.) The approach is good for modeling objects whose surfaces do not have a strong texture component, while modeling of textured surfaces is somewhat tedious using this approach.

Besl and Jain[34] described an iterative algorithm for fitting variable-order bivariate surfaces to image data. Every pixel in the image is initially labeled as belonging to one of eight possible surface types, depending on the surface curvature sign computed using an approximating surface that best fits the image data surrounding that point. Pixels with similar labels are aggregated using a connected-component procedure to obtain initial regions. Pixels that are interior to these regions are chosen as seed regions for iterative surface fitting and a region-growing algorithm. The order of the function fitting the region is incremented from planar to biquartic, as necessary, until convergence criteria are satisfied. This algorithm has been applied to both range and intensity images.

Split and merge. The major problem with the results from a simple intensity-based segmentation is that there are usually too many regions. Even in images where most human observers will report very clear, uniform-intensity regions, the output of a thresholding algorithm will contain many spurious regions. The main reasons for this problem are high-frequency noise and smooth transition between uniform regions. In most applications, after initial intensity-based region formation, a method for refining or reforming the regions is needed. Several approaches have been proposed for processing such regions. Some of these approaches use domain-dependent knowledge, while others use knowledge about imaging. The refinement may be done interactively — by a person — or automatically — by a computer. In an automatic system, the segmentation will have to be refined based on object characteristics and general knowledge about the images.

Automatic refinement is done using a combination of split and merge operations. Split and merge operations eliminate false boundaries and spurious regions by either splitting a region that contains pieces from more than one object or merging adjacent regions that actually belong to the same object. Some possible approaches for this refinement are to

- Merge similar "adjacent" regions;
- Remove questionable edges;
- Use topological properties of the regions;
- Use shape information about objects in the scene; and
- Use semantic information about the scene.

The last two approaches require domain-dependent information, while the first three approaches use only intensity values and other domain-independent characteristics of regions.

Split. If some "property" of a region is not uniform, the region should be split. Segmentation based on the split approach starts with large regions. In many cases, the whole image may be used as the starting region. Several decisions must be made before a region is split. One is to decide when a property is nonuniform over a region; another is how to split a region so that the property for each of the resulting components is uniform. These decisions usually are not easy to make. In some applications, the variance of the intensity values is used as a measure of uniformity; in other applications, the error in the best functional approximation is used. More difficult than determining property uniformity is deciding where to split a region. Splitting regions based on property values is very difficult. One approach used when trying to determine the best boundaries with which to divide a region is to consider edgeness values within the region. The easiest schemes for splitting regions are those that divide the region into a fixed number of equal regions; such methods are called "regular decomposition methods." For example, in the quadtree approach, the region is split into four quadrants in each step.

Clearly, because of the numerous ways in which a region may be split, splitting regions is generally more difficult than merging them.

Merge. Many approaches have been proposed to judge similarity of regions. Broadly, the approaches to judge region similarity

are based on either the characteristics of regions or the weakness of edges between them. Two approaches to judging the similarity of adjacent regions are to

(1) Compare their mean intensities. In this approach, if the mean intensities do not differ by more than some predetermined value, the regions are considered to be similar. The regions are then candidates for merging. A modified form of this approach uses surface fitting to determine if the regions can be approximated by one surface.

(2) Assume that the intensity values are drawn from a probability distribution. In this approach, the decision of whether or not to merge adjacent regions is based on considering the probability that they will have the same statistical distribution of intensity values. This approach uses hypothesis testing to judge the similarity of adjacent regions.

The first approach is easy to implement. The second approach is more powerful than the first. A description of the second approach follows.

Suppose that two adjacent regions R_1 and R_2 contain points m_1 and m_2, respectively. The two possible hypotheses are listed below.

- H_0: Both regions belong to the same object ≡ the intensities are all drawn from a single Gaussian distribution with parameters (μ_0,σ_0).
- H_1: The regions belong to different objects ≡ the intensities of each region are drawn from separate Gaussian distributions with parameters (μ_1,σ_1), (μ_2,σ_2).

Assuming that the intensities of different points are independent, it can be shown that the likelihood ratio is given by

$$L = \frac{Probability\ that\ there\ are\ two\ regions}{Probability\ there\ is\ one\ region} = \frac{P_1 \cdot P_2}{P} = \frac{\left(\sigma_0^{\,2}\right)^{m_1+m_2}}{\left(\sigma_1^{\,2}\right)^{m_1} \cdot \left(\sigma_2^{\,2}\right)^{m_2}} \tag{2.18}$$

A likelihood ratio L that is below a threshold value indicates strongly that the existence of only one region is more likely than that of two regions. These two regions may be merged. This approach can be used for edge detection also. Since the likelihood ratio indicates when two regions should be considered to be separate, it also indicates when a boundary should be between two regions. For edge detection, the likelihood ratio between fixed neighborhoods on either side of a point can be used to detect the presence of edges. Other possible modifications to this ratio exist; these can play an important role in many applications.

Another approach to merging is to combine two regions if the boundary between them is weak. This approach attempts to remove weak edges between adjacent regions by considering not only the intensity characteristics, but also the length of the common boundary. The common boundary is dissolved if the boundary is "weak" and the resulting boundary (of the merged region) does not grow too fast. (A weak boundary is one for which the intensities on either side differ by less than an amount T.) Other criteria, such as edgeness values, can be used to determine strength of an edge point that is on the boundary separating two regions.

Split and merge operations may be used together. After a presegmentation based on thresholding, a succession of splits and merges may be applied as dictated by the properties of the regions. Such schemes have been proposed for segmentation of complex scenes. Domain knowledge with which the split and merge operations can be controlled may be introduced.

Other segmentation methods

In the preceding discussion, the primary concern was with intensity images. Segmentation techniques that are based on color,[35-37] texture,[38-42] and motion (described in more detail in Chapter 6) have also been developed. Segmentation based on spectral pattern classification techniques is extensively used in remote-sensing applications.

Research trends

After more than two decades of research efforts, segmentation still remains a problem. Many approaches to computer vision do not use any domain-specific information in the early stages.[43] On the other hand, some psychophysicists believe that every aspect of perception uses domain information extensively.[44] Computer vision research has been influenced by both of these views. Haralick and Shapiro[45] provide an excellent review of early segmentation techniques. The authors classify and define several segmentation techniques and illustrate them with examples.

Nine papers were selected for inclusion in this chapter: five are in this book and the remaining four in its companion book, *Computer Vision: Advances and Applications*. Edge detection still remains one of the most active areas of research in computer vision. Scale-space has been attracting increasing attention in the last few years.[25,27,46-48] Although most scale-space research has been concerned with possible reconstruction of signals, approaches are being developed for combining information at different scales.[27,49] The idea of using procedural reasoning, like that proposed by Georgeff and Lansky,[50] is likely to become increasingly important in computer vision; this will be the obvious next step after researchers have tried knowledge-based approaches for segmentation.[51,52] The first paper presented in *Principles*, "Theory of Edge Detection," by Marr and Hildreth, describes the Laplacian of Gaussian (LOG) operator for detecting edge segments. This paper is followed by "Scale-Space Filtering," by Witkin, and "A Computational Approach to Edge Detection," by Canny. Continuing in *Principles*, "Discontinuity Detection for Visual Surface Reconstruction," by Grimson and Pavlidis introduces a technique in which the statistical distribution of error in approximation of intensity values by a planar surface is used to detect discontinuities. Nalwa and Binford[29] discuss how groups of edge pixels — or edgels — are detected by fitting a one-dimensional surface to data. In *Advances*, Eichel et al. describe in their paper entitled "A Method for a Fully Automatic Definition of Coronary Arterial Edges From Cineangiograms," an algorithm for edge linking and its performance on angiogram images.

Though many edge detectors have been developed, no well-defined metric helps researchers select the appropriate edge detector for a given application. This lack of a performance measure makes judicious selection of an edge detector for a given application a difficult problem.

In the early days of computer vision, region-growing and other region-based approaches received the attention of researchers.[45] In the last few years, region-oriented segmentation approaches have once again become increasingly popular because of the availability of range images.[34,53-55] These approaches are usually more robust than edge-detection based segmentation techniques. In *Principles*, in the paper entitled "Segmentation Through Variable-Order Surface Fitting," Besl and Jain describe a functional approximation method in which image data are represented by piecewise smooth surfaces. It appears obvious that features of edge detection and region growing should be combined to yield good early segmentation; however, only little attention has been given in this direction.[56,57]

In many region-based image analysis methods, texture plays an important role. Van Gool et al.[39] present a review of methods used for texture analysis. Many techniques have been developed to model texture using statistical,[58,59] structural,[60,61] and spectral[62] methods. With the use of the models described in "Markov Random Field Texture Models," by Cross and Jain in *Advances*, blurry, sharp, linelike, and bloblike textures can be generated. This paper gives methods for estimating the parameters of the models to approximate various natural textures and a comparison of the generated synthetic texture and natural textures. The paper by Malik and Perona, "Preattentive Texture Discrimination With Early Vision Mechanisms" in *Advances*, is the latest, and the most convincing, model of human preattentive texture perception. A recent book, *A Taxonomy of Texture Description and Identification*, by Rao,[63] gives a good overview of approaches for texture classification and their applications.

We conclude *Advances* with "Using Color for Geometry-Insensitive Segmentation," by Healey which describes representative segmentation techniques that use color information.

Explicit application of knowledge for segmentation has been addressed in many systems.[50,51,64,65] Many of these systems showed promise, but had limited success because of poor performance of operators in domain-independent early processing. Model-based reasoning can help in this direction.[66] The literature has clearly started showing a strong trend toward using explicit, mostly geometric, models that help operators in early processing. Model-based reasoning will likely emerge as a powerful theme in the near future.

References Cited
Chapter 2

22. Sobel, I., "Camera Models and Machine Perception," *Stanford AI Memo 121*, Department of Computer Science, Stanford University, Stanford, Calif., May 1970.
23. R. Kirsch, "Computer Determination of the Constituent Structure of Biological Images," *Computers and Biomedical Research*, Vol. 4, No. 3, 1971, pp. 315-328.
24. D. Marr and E. Hildreth, "Theory of Edge Detection," *Proc. R. Soc. Lond. B*, Vol. 207, 1980, pp. 187-217.
25. A.P. Witkin, "Scale-Space Filtering," *Int'l Joint Conf. Artificial Intelligence*, Morgan Kaufmann Publishers, Inc., San Mateo, Calif., 1983, pp. 1019-1022.
26. J.F. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, 1986, pp. 679-698.
27. Y. Lu and R.C. Jain, "Behavior of Edges in Scale Space," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 3, 1989, pp. 337-356.
28. R.M. Haralick, "Digital Step Edges from Zero Crossing f Second Directional Derivatives," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 6, No. 1, 1984, pp. 58-68.
29. V.S. Nalwa and T.O. Binford, "On Detecting Edges," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, 1986, pp. 699-714.
30. W.E.L. Grimson and T. Pavlidis, "Discontinuity Detection for Visual Surface Reconstruction," *Computer Vision, Graphics, and Image Processing*, Vol. 30, 1985, pp. 316-330.
31. P.H. Eichel et al, "A Method for a Fully Automatic Definition of Coronary Arterial Edges from Cineangiograms," *IEEE Trans. Medical Imaging*, Vol. 7, No. 4, Dec. 1988, pp. 313-320.
32. S.W. Zucker, R.A. Hummel, and A. Rosenfeld, "An Application of Relaxation Labeling to Line and Curve Enhancement" *IEEE Trans. Computers*, Vol. 26, 1977, pp. 394-403.
33. J.M. Prager, "Extracting and Labeling Boundary Segments in Natural Scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 2, No. 1, 1980, pp. 16-27.
34. P. Besl and R. Jain, "Segmentation through Variable-Order Surface Fitting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 10, No. 2, 1988, pp. 167-192.
35. R. Ohlander, K. Price, and D.R. Reddy, "Picture Segmentation Using a Recursive Splitting Method," *Comp. Graph. and Image Processing*, Vol. 8, 1978, pp. 313-333.
36. Y. Ohta, T. Kanade, and T. Sakai, "Color Information for Region Segmentation," *Comp. Graph. and Image Processing*, Vol. 13, 1980, pp. 222-241.
37. G. Healey, "Using Color for Geometry-Insensitive Segmentation," *J. Opt. Soc. Am. A*, Vol. 6, No. 6, 1989, pp. 920-937.
38. K.S. Fu, and J.K. Mui, "A Survey on Image Segmentation," *Pattern Recognition*, Vol. 13, 1981, pp. 3-16.
39. L. Van Gool, P. Dewaele, and O. Oosterlinck, "Texture Analysis Anno 1983," *Computer Vision, Graphics, and Image Processing*, Vol. 29, 1985, pp. 336-357.
40. H. Voorhees and T. Poggio, "Computing Texture Boundaries from Images," *Nature*, Vol. 333, 1988, pp. 364-367.
41. R.L. Kashyap and K.B. Eom, "Texture Boundary Detection Based on the Long Correlation Model," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 1, 1989, pp. 58-67.
42. J. Malik and P. Perona, "Preattentive Texture Discrimination with Early Vision Mechanisms," *J. Opt. Soc. Am. A*, Vol. 7, No. 5, May 1990, pp. 923-932.
43. D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, W.H. Freeman & Co., San Francisco, Calif., 1982.
44. I. Rock, *The Logic of Perception*, MIT Press, Cambridge, Mass., 1983.
45. R.M. Haralick and L.G. Shapiro, "Image Segmentation Techniques," *Computer Vision, Graphics, and Image Processing*, Vol. 29, 1985, pp. 100-133.
46. V. Torre and T.A. Poggio, "On Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 8, No. 2, 1986, pp. 147-163.
47. A.L. Yuille and T. Poggio, "Scaling Theorems for Zero Crossings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 8, No. 1, 1986, pp. 15-25.
48. L. Wu and Z. Xie, "Scaling Theorems for Zero Crossings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, 1990, pp. 46-54.
49. F. Bergholm, "Edge Focusing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 9, No. 6, 1987, pp. 726-741.
50. M.P. Georgeff and A.L. Lansky, "Procedural Knowledge," *Proc. IEEE*, Vol. 74, IEEE Press, New York, N.Y., 1986, pp. 1383-1398.
51. A.M. Nazif and M.D. Levine, "Low Level Image Segmentation: An Expert System," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 6, No. 5, 1984, pp. 555-577.
52. J.R. Beveridge et al, "Segmenting Images Using Localized Histograms and Region Merging," *Int'l J. Computer Vision*, Vol. 2, No. 3, 1989, pp. 311-347.
53. F. Solina and R. Bajcsy, "Recovery of Parametric Models from Range Images: The Case for Superquadrics with Global Deformations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 1, 1990, pp. 131-148.
54. R. Hoffman and A.K. Jain "Segmentation and Classification of Range Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 9, No. 5, 1987, pp. 608-620.

55. T.-J. Fan, G. Medioni, and R. Nevatia, "Segmented Descriptions of 3-D Surfaces," *IEEE Trans. Robotics and Automation*, Vol. 3, No. 6, 1987, pp. 527-538.
56. T. Pavlidis and Y.-T. Liow, "Integrated Region Growing and Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, No. 3, 1990, pp. 225-233.
57. S.P. Liou and R.C. Jain, "A Parallel Technique for Three-Dimensional Scene Segmentation," *Proc. Tenth Int'l Conf. Pattern Recognition*, IEEE CS Press, Los Alamitos, Calif., 1990, pp. 201-203.
58. R.M. Haralick, "Statistical and Structural Approaches to Texture," *Proc. IEEE*, IEEE Press, Vol. 67, No. 5, New York, N.Y., 1979, pp. 786-804.
59. G.R. Cross and A.K. Jain, "Markov Random Field Texture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 5, No. 1, 1983, pp. 25-39.
60. S.Y. Lu and K.S. Fu, "A Syntactic Approach to Texture Analysis," *Computer Graphics and Image Processing*, Vol. 7, 1978, pp. 303-330.
61. F. Tomita, Y. Shirai, and S. Tsuji, "Description of Texture by Structural Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 4, No. 2, 1982, pp. 183-191.
62. R. Bajcsy and L. Lieberman, "Texture Gradient as a Depth Cue," *Computer Graphics and Image Processing*, Vol. 5, 1977, pp. 52-67.
63. A. R. Rao, *Taxonomy for Texture Description and Identification*, Springer-Verlag, New York, N.Y., 1990.
64. T. Matsuyama, "Expert Systems for Image Processing: Knowledge-Based Composition of Image Analysis Processes," *Computer Vision, Graphics, and Image Processing*, Vol. 48, 1989, pp. 22-49.
65. V.S.S. Hwang, L.S. Davis, and T. Matsuyama, "Hypothesis Integration in Image Understanding Systems," *Computer Vision, Graphics, and Image Processing*, Vol. 36, 1986, pp. 321-371.
66. R.A. Brooks, "Model-Based Three-Dimensional Interpretations of Two-Dimensional Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 5, No. 2, 1983, pp. 140-150.

Proc. R. Soc. Lond. B **207**, 187–217 (1980)
Printed in Great Britain

Theory of edge detection

BY D. MARR AND E. HILDRETH

M.I.T. Psychology Department and Artificial Intelligence Laboratory, 79 Amherst Street, Cambridge, Massachusetts 02139, *U.S.A.*

(*Communicated by S. Brenner, F.R.S. – Received 22 February* 1979)

A theory of edge detection is presented. The analysis proceeds in two parts. (1) Intensity changes, which occur in a natural image over a wide range of scales, are detected separately at different scales. An appropriate filter for this purpose at a given scale is found to be the second derivative of a Gaussian, and it is shown that, provided some simple conditions are satisfied, these primary filters need not be orientation-dependent. Thus, intensity changes at a given scale are best detected by finding the zero values of $\nabla^2 G(x, y) * I(x, y)$ for image I, where $G(x, y)$ is a two-dimensional Gaussian distribution and ∇^2 is the Laplacian. The intensity changes thus discovered in each of the channels are then represented by oriented primitives called zero-crossing segments, and evidence is given that this representation is complete. (2) Intensity changes in images arise from surface discontinuities or from reflectance or illumination boundaries, and these all have the property that they are spatially localized. Because of this, the zero-crossing segments from the different channels are not independent, and rules are deduced for combining them into a description of the image. This description is called the raw primal sketch. The theory explains several basic psychophysical findings, and the operation of forming oriented zero-crossing segments from the output of centre–surround $\nabla^2 G$ filters acting on the image forms the basis for a physiological model of simple cells (see Marr & Ullman 1979).

INTRODUCTION

The experiments of Hubel & Wiesel (1962) and of Campbell & Robson (1968) introduced two rather distinct notions of the function of early information processing in higher visual systems. Hubel & Wiesel's description of simple cells as linear with bar- or edge-shaped receptive fields led to a view of the cortex as containing a population of feature detectors (Barlow 1969, p. 881) tuned to edges and bars of various widths and orientations. Campbell & Robson's experiments, showing that visual information is processed in parallel by a number of independent orientation and spatial-frequency-tuned channels, suggested a rather different view, which, in its extreme form, would describe the visual cortex as a kind of spatial Fourier analyser (Pollen *et al.* 1971; Maffei & Fiorentini 1977).

Protagonists of each of these views are able to make substantial criticisms of the other. The main points against a Fourier interpretation are: (1) The bandwidth of the channels is not narrow (1.6 octaves, Wilson & Bergen 1979). The corresponding receptive fields have a definite spatial localization. (2) As Campbell & Robson found, early visual information processing is not linear (e.g. probability summation (Graham 1977; Wilson & Giese 1977), and failure of superposition (Maffei & Fiorentini 1972*a*)). (3) Only rudimentary phase information is apparently encoded (Atkinson & Campbell 1974).

The main point against the linear feature-detector idea is that if a simple cell truly signals either the positive or the negative part of the linear convolution of its bar-shaped receptive field with the image intensity, it can hardly be thought of as making some symbolic assertion about the presence of a bar in the image (Marr 1976*a*, p. 648). Such a cell would necessarily respond to many stimuli other than a bar, more vigorously, for example, to a bright edge than to a dim bar, and thus would not be specific enough in its response to warrant being called a feature detector.

Perhaps the greatest difficulty faced by both camps is that neither approach can give direct information about the goals of the early analysis of an image. This motivated a new approach to vision, which enquired directly about the information processing problems inherent in the task of vision itself (Marr 1976*a*, *b*; and see Marr 1978 for the overall scheme). According to this scheme, the purpose of early visual processing is to construct a primitive but rich description of the image that is to be used to determine the reflectance and illumination of the visible surfaces, and their orientation and distance relative to the viewer. The first primitive description of the image was called the primal sketch (*Marr* 1976*b*) and it is formed in two parts. First, a description is constructed of the intensity changes in an image, using a primitive language of edge-segments, bars, blobs and terminations. This description was called the raw primal sketch (Marr 1976*b*, p. 497). Secondly, geometrical relations are made explicit (using virtual lines), and larger, more abstract tokens are constructed by selecting, grouping and summarizing the raw primitives in various ways. The resulting hierarchy of descriptions covers a range of scales, and is called the full primal sketch of an image.

Although the primal sketch was inspired by findings about mammalian visual systems, we were until recently unable to make it the basis of a detailed theory of human early vision. Three developments have made this possible now: (*a*) the emergence of quantitative information about the channels present in early human vision (Cowan 1977; Graham 1977; Wilson & Giese 1977; Wilson & Bergen 1979); (*b*) Marr & Poggio's (1979) theory of human vision (especially the framework within which it was written); and (*c*) the related observations of Marr *et al.* (1979) about the relevance of a result like Logan's (1977) theorem to early vision.

These advances have made possible the formulation of a satisfactory computational theory. This article deals with the first part, the derivation of the raw primal sketch. The theory itself is given in two sections, the first dealing with the

analysis within each channel, and the second, with combining information from different channels. Each computational section discusses algorithms for implementing the theory, and gives examples.

The second half of the article examines the implications for biology. The behaviour of the algorithms is shown to account for a range of basic psychophysical findings, and a specific neural implementation is presented. Our model is not intended as a complete proposal for a physiological mechanism, because it ignores the attribute of directional selectivity that so pervades cortical simple cells. The model does, however, make explicit certain nonlinear features that we regard as critical, and it forms the starting point for the more complete proposal of Marr & Ullman (1979), which incorporates directional selectivity.

Detecting and representing intensity changes in an image

A major difficulty with natural images is that changes can and do occur over a wide range of scales (Marr 1976*a*, *b*). No single filter can be optimal simultaneously at all scales, so it follows that one should seek a way of dealing separately with the changes occurring at different scales. This requirement, together with the findings of Campbell & Robson (1968), leads to the basic idea, illustrated in figure 1, in which one first takes local averages of the image at various resolutions and then detects the changes in intensity that occur at each one. To realize this idea, we need to determine (*a*) the nature of the optimal smoothing filter, and (*b*) how to detect intensity changes at a given scale.

The optimal smoothing filter

There are two physical considerations that combine to determine the appropriate smoothing filter. The first is that the motivation for filtering the image is to reduce the range of scales over which intensity changes take place. The filter's spectrum should therefore be smooth and roughly band-limited in the frequency domain. We may express this condition by requiring that its variance there, $\Delta\omega$, should be small.

The second consideration is best expressed as a constraint in the spatial domain, and we call it the constraint of spatial localization. The things in the world that give rise to intensity changes in the image are: (1) illumination changes, which include shadows, visible light sources and illumination gradients; (2) changes in the orientation or distance from the viewer of the visible surfaces; and (3) changes in surface reflectance. The critical observation here is that, at their own scale, these things can all be thought of as spatially localized. Apart from the occasional diffraction pattern, the visual world is not constructed of ripply, wave-like primitives that extend and add together over an area (c.f. Marr 1970, p. 169), but of contours, creases, scratches, marks, shadows and shading.

The consequence for us of this constraint is that the contributions to each

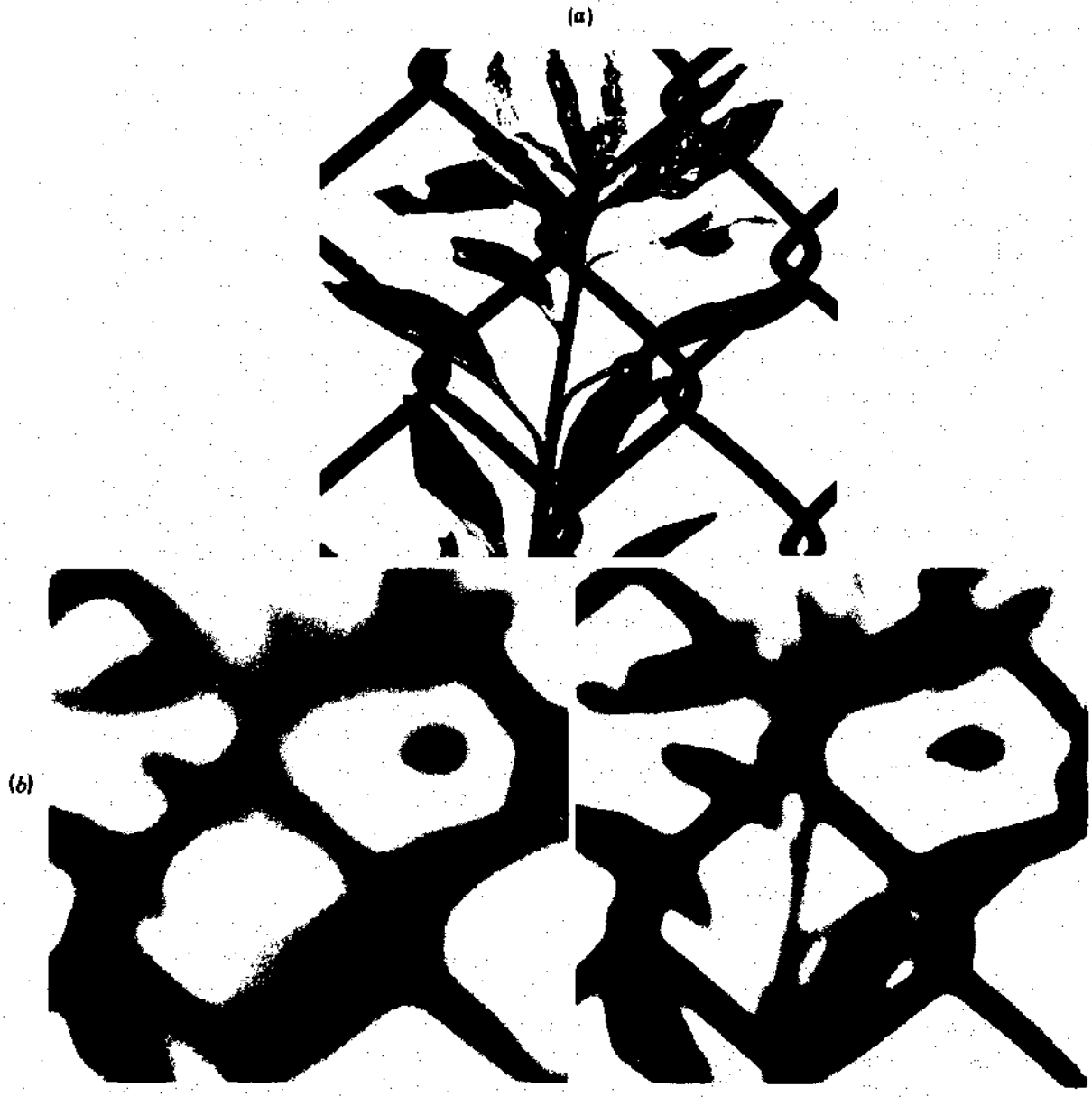


FIGURE 1. A local-average filtered image. In the original image (*a*), intensity changes can take place over a wide range of scales and no single operator will be very efficient at detecting all of them. The problem is much simplified in a Gaussian band-limited filtered image because there is effectively an upper limit to the rate at which changes can take place. The first part of our scheme can be thought of as decomposing the original image into a set of copies, each filtered like this, and detecting the intensity changes separately in each. In (*b*) the image is filtered with a Gaussian having $\sigma = 8$ picture elements, and, in (*c*), $\sigma = 4$. The image is 320×320 picture elements.

point in the filtered image should arise from a smooth average of nearby points, rather than any kind of average of widely scattered points. Hence the filter that we seek should also be smooth and localized in the spatial domain, and in particular its spatial variance, Δx, should also be small.

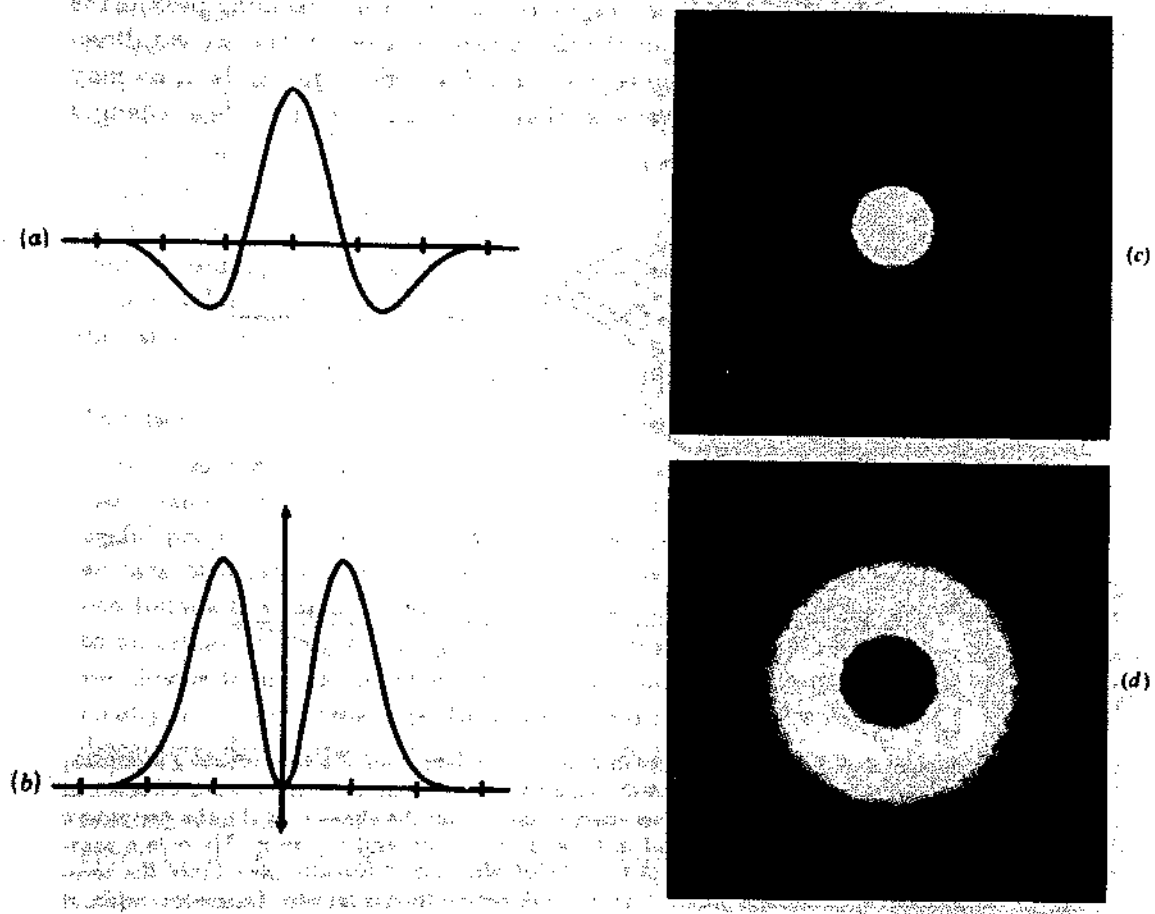


FIGURE 2. The operators G'' (equation 5) and $\nabla^2 G$: (*a*) shows G'', the second derivative of the one-dimensional Gaussian distribution; (*c*) shows $\nabla^2 G$, its rotationally symmetric two-dimensional counterpart; (*b*) and (*d*) exhibit their Fourier transforms.

Unfortunately, these two localization requirements, the one in the spatial and the other in the frequency domain, are conflicting. They are, in fact, related by the uncertainty principle, which states that $\Delta x\, \Delta\omega \geqslant \frac{1}{4}\pi$ (see, for example, Bracewell 1965, pp. 160–163). There is, moreover, only one distribution that optimizes this relation (Leipnik 1960), namely the Gaussian

$$G(x) = [1/\sigma(2\pi)^{\frac{1}{2}}] \exp(-x^2/2\sigma^2), \text{ with Fourier transform} \tag{1}$$

$$\tilde{G}(\omega) = \exp(-\tfrac{1}{2}\sigma^2\omega^2). \tag{2}$$

In two dimensions, $G(r) = (\frac{1}{2}\pi\sigma^2) \exp(-r^2/2\sigma^2)$.

The filter G thus provides the optimal trade-off between our conflicting requirements.

Detecting intensity changes

Wherever an intensity change occurs, there will be a corresponding peak in the first directional derivative, or equivalently, a zero-crossing in the second directional derivative of intensity (Marr 1976*b*; Marr & Poggio 1979). In fact, we may define an intensity change in this way, so that the task of detecting these changes

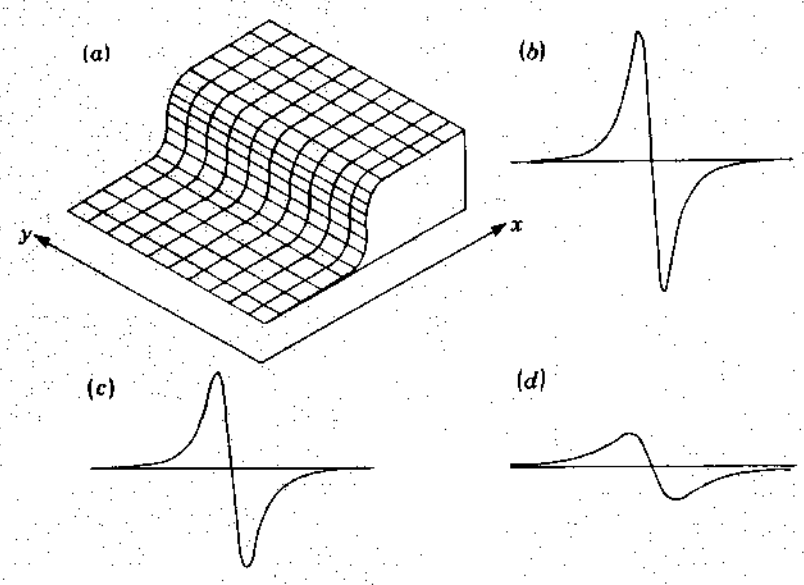


FIGURE 3. Spatial and directional factors interact in the definition of a zero-crossing segment; (*a*) shows an intensity change, and (*b*), (*c*) and (*d*) show values of the second directional derivative near the origin at various orientations across the change. In (*b*), the derivative is taken parallel to the x-axis, and in (*c*) and (*d*), at 30° and 60° to it. There is a zero-crossing at every orientation except for $\partial^2 I/\partial y^2$, which is identically zero. Since the zero-crossings line up along the y-axis, this is the direction that is chosen. In this example, it is also the direction that maximizes the slope of the second derivative.

can be reduced to that of finding the zero-crossings of the second derivative D^2 of intensity, in the appropriate direction. That is to say, we seek the zero-crossings in

$$f(x, y) = D^2[G(r)*I(x, y)], \tag{3}$$

where $I(x, y)$ is the image, and $*$ is the convolution operator. By the derivative rule for convolutions,

$$f(x, y) = D^2G * I(x, y). \tag{4}$$

We can write the operator D^2G as G'', and in one dimension

$$G''(x) = [-1/\sigma^3(2\pi)^{\frac{1}{2}}]\,(1 - x^2/\sigma^2)\exp(-x^2/2\sigma^2). \tag{5}$$

$G''(x)$ looks like a Mexican hat operator (see figure 2), it closely resembles Wilson & Giese's (1977) difference of two Gaussians (DOG), and it is, in fact, the limit of the DOG function as the sizes of the two Gaussians tend to one another (see figure 11 and appendix B). It is an approximately bandpass operator, with a half-power bandwidth of about 1.2 octaves, and so it can be thought of as looking at the information contained in one particular part of the spectrum of the image.

These arguments establish that intensity changes at one scale may, in principle, be detected by convolving the image with the operator D^2G and looking for zero-crossings in its output. Only one issue is still unresolved, and it concerns the orientation associated with D^2. It is not enough to choose zero-crossings of the second derivative in *any* direction. To understand this, imagine a uniform intensity change running down the y-axis, as shown in figure 3. At the origin, the second directional derivative is zero in every direction, but it is non-zero nearby in every direction except along the y-axis.

In which direction should the derivative be taken?

To choose which directional derivative to use, we observe that the underlying motivation for detecting changes in intensity is that they will correspond to useful properties of the physical world, like changes in reflectance, illumination, surface orientation, or distance from the viewer. Such properties are spatially continuous and can almost everywhere be associated with a direction that projects to an orientation in the image. The orientation of the directional derivative that we choose to use is therefore that which coincides with the orientation formed locally by its zero-crossings. In figure 3, this orientation is the y-axis, and so the directional derivative we would choose there is $\partial^2 I/\partial x^2$.

Under what conditions does this direction coincide with that in which the zero-crossing has maximum slope? The answer to this is given by theorem 1 (see appendix A), and we call it the *condition of linear variation*:

> the intensity variation near and parallel to the line of zero-crossings should locally be linear.

This condition will be approximately true in smoothed images, and in the rest of this article we shall assume that the condition of linear variation holds.

This direction can be found by means of the Laplacian

There are three main steps in the detection of zero-crossings. They are: (1) a convolution with D^2G, where D^2 stands for a second directional derivative operator; (2) the localization of zero-crossings; and (3) checking of the alignment and orientation of a local segment of zero-crossings. Although it is possible to implement this scheme directly (Marr 1976*b*, p. 494), one immediate question that can be asked is, are directional derivatives of critical importance here? Convolutions are relatively expensive, and it would much lessen the computational burden if

their number could be reduced, for example, by using just one orientation-independent operator.

The only orientation-independent second-order differential operator is the Laplacian ∇^2, and theorem 2 (see appendix A) makes explicit the conditions under which it can be used. They are weaker than the condition of linear variation, which we met in theorem 1, and they state that provided the intensity variation in $(G * I)$ is linear along but not necessarily near to a line of zero-crossings, then the zero-crossings will be detected and accurately located by the zero values of the Laplacian. Again, because in our application the condition of linear variation is approximately satisfied, so will be this condition. It follows that the detection of intensity changes can be based on the filter $\nabla^2 G$, illustrated in figure 2. It is, however, worth remembering that in principle, if intensity varies along a segment in a very non-linear way, the Laplacian, and hence the operator $\nabla^2 G$ will see the zero-crossing displaced to one side.

Summary of the argument

The main steps in the argument so far are, therefore, these.

(1) To limit the rate at which intensities can change, we first convolve the image I with a two-dimensional Gaussian operator G.

(2) Intensity changes in $G * I$ are then characterized by the zero-crossings in the second directional derivative $\mathrm{D}^2(G * I)$. This operator is roughly bandpass, and so it examines only a portion of the spectrum of the image.

(3) The orientation of the directional derivative should be chosen to coincide with the local orientation of the underlying line of zero-crossings.

(4) Provided that the condition of linear variation holds, this orientation is also the one at which the zero-crossing has maximum slope (measured perpendicular to the orientation of the zero-crossing).

(5) By theorem 1 of appendix A, if the condition of linear variation holds, the lines of zero-crossings defined by (3) are precisely the zero-crossings of the orientation-independent differential operator, the Laplacian ∇^2.

(6) The loci of zero-crossings defined by (3) may therefore be detected economically in the image at each given scale by searching for the zero values of the convolution $\nabla^2 G * I$. In two dimensions,

$$\nabla^2 G(r) = -1/\pi\sigma^4[1 - r^2/2\sigma^2] \exp(-r^2/2\sigma^2).$$

We turn now to the question of how to represent the intensity changes thus detected.

Representing the intensity changes

In a band-limited image, changes take place smoothly, so it is always possible to divide a line of zero-crossings into small segments, each of which approximately obeys the condition of linear variation. This fact allows us to make the following definitions.

The raw primal sketch

(1) *A zero-crossing segment* in a Gaussian filtered image consists of a linear segment l of zero-crossings in the second directional derivative operator whose direction lies perpendicular to l.

(2) We can also define an *amplitude* ν associated with a zero-crossing segment, as the slope of the directional derivative taken perpendicular to the segment. To see why this is an appropriate measure, observe that a narrow bandpass channel near a zero-crossing at the origin can be described approximately by $\nu \sin \omega x$, which has slope $\nu\omega$ at the origin. Hence, if s is the measured slope of the zero-crossing, $\nu = s/\omega$. The factor $1/\omega$ is a space constant, and scales linearly with the sampling interval required.

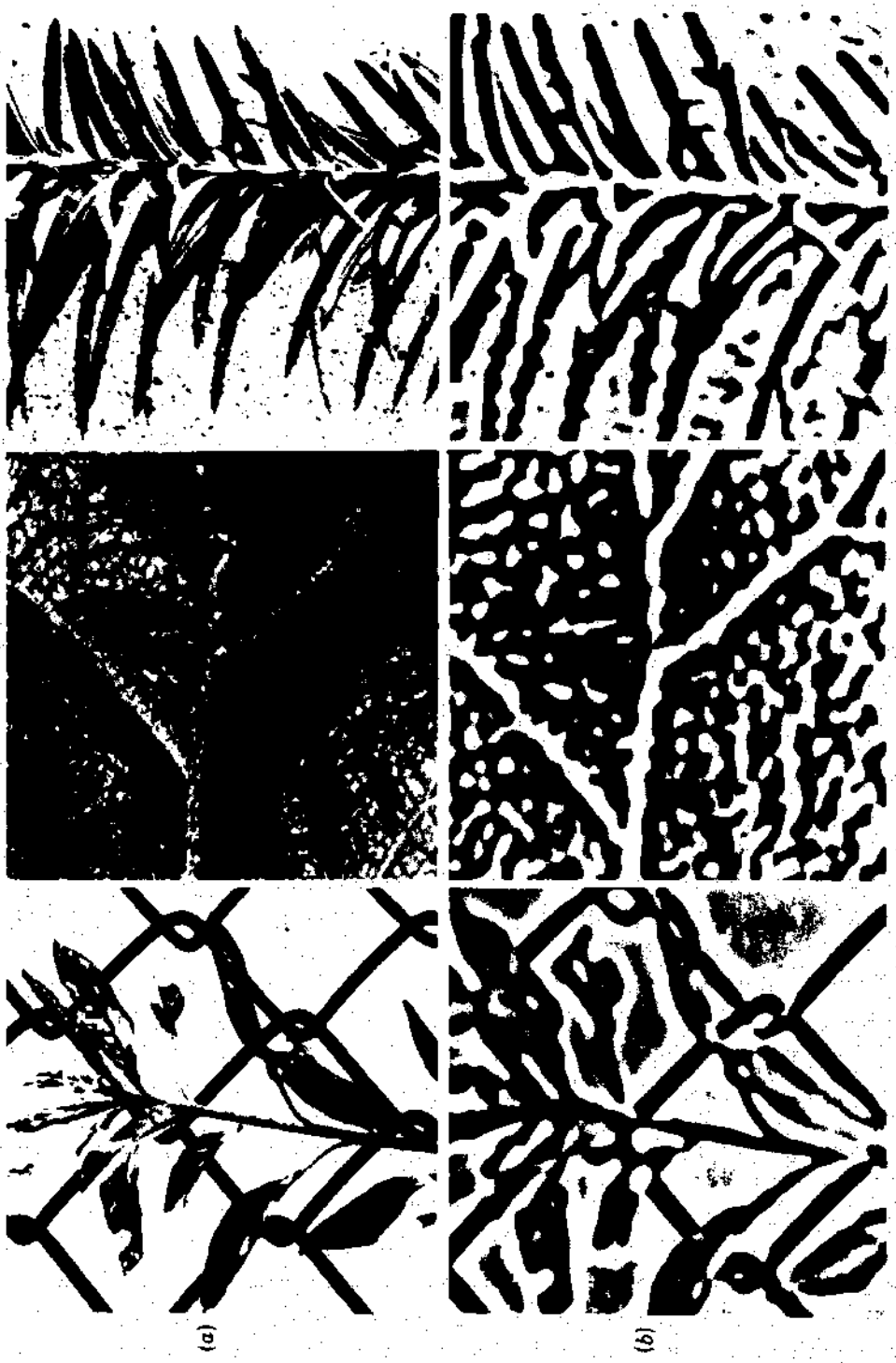
The set of zero-crossing segments together with their amplitudes, constitutes a primitive symbolic representation of the changes taking place within one region of the spectrum of an image. Full coverage of the spectrum can now be had simply by applying the analysis over a sufficient number of channels simultaneously.

Finally, there are grounds for believing that this representation of the image is complete. Marr *et al.* (1979) noted that Logan's (1977) recent theorem, about the zero-crossings of one-octave bandpass signals, shows that the set of such zero-crossing segments is extremely rich in information. If the filters had bandwidth of an octave or less, they would in fact contain complete information about the filtered image. In practice, the $\nabla^2 G$ filter has a half-sensitivity bandwidth of about 1.75 octaves, which puts it outside the range in which Logan's theorem applies. On the other hand, if we add information about the slopes of the zero-crossings, the situation may be more congenial. In the standard sampling theorem, if the first derivative, as well as the value, is given, the sampling density can be halved (see, for example, Bracewell 1978, pp. 198–200). It seems likely than an analogous extension holds for Logan's (1977) theorem. If this were true, the zero-crossing segments, whose underlying motivation is physical, would in fact provide a sufficient basis for the recovery of arbitrary intensity profiles.

In summary, then, we have shown how intensity changes at one scale may be detected by means of the $\nabla^2 G$ operator and that they may be represented, probably completely, by oriented zero-crossing segments and their amplitudes. To detect changes at all scales, it is necessary only to add other channels, like the one described above, and to carry out the same computation in each. These representations are precursors of the descriptive primitives in the raw primal sketch, and mark the transition from the 'analytic' to the 'symbolic' analysis of an image. The remaining step is to combine the zero-crossings from the different channels into primitive 'edge' elements, and this task is addressed later in the article.

Examples and comments

Figure 4 shows some examples of zero-crossings. The top row shows images and the second shows their convolutions with the operator $\nabla^2 G$, exhibited in figure 2. Zero is represented here by an intermediate grey, so that very positive values



(a) (b)

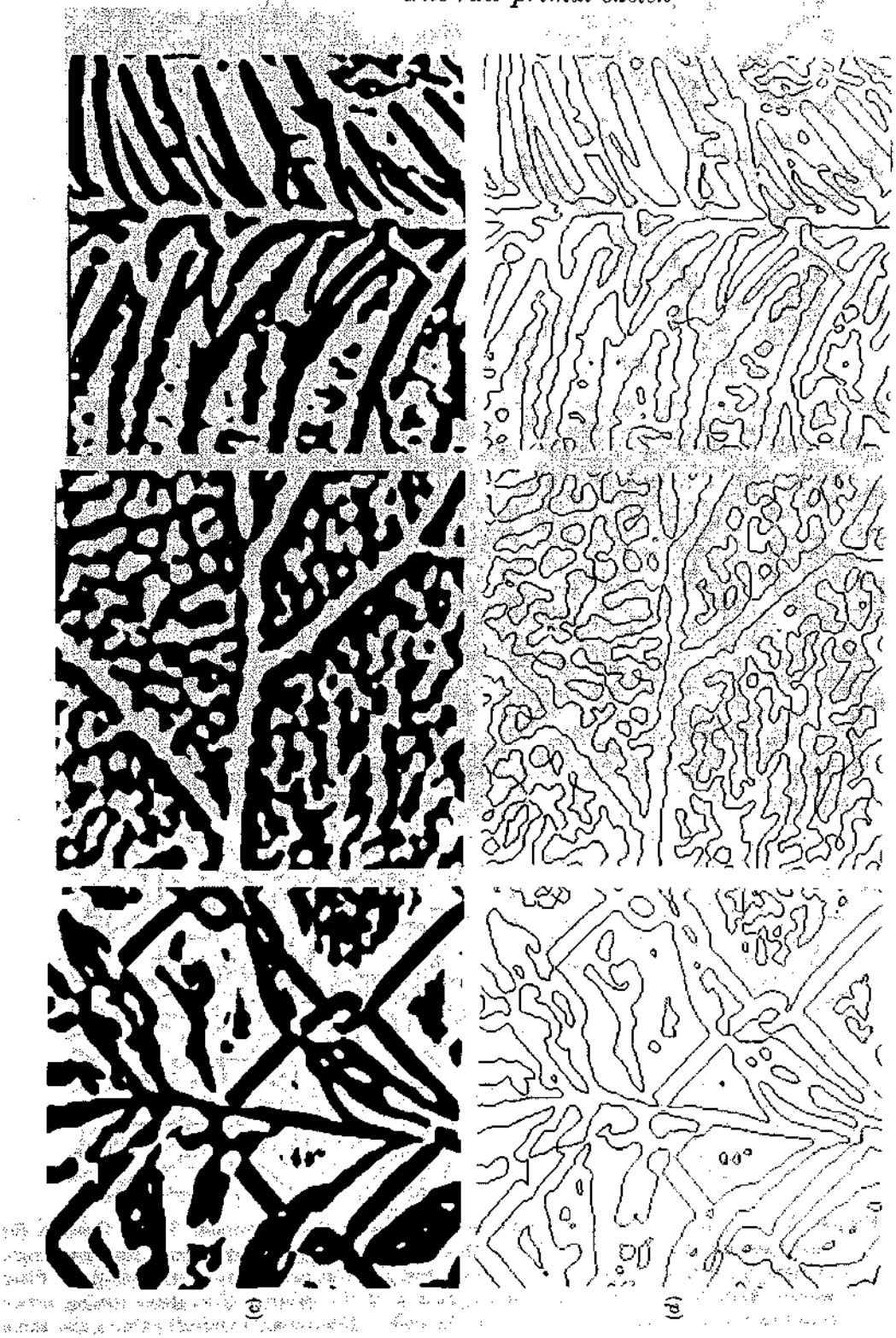


FIGURE 4. Examples of zero-crossing detection by means of $\nabla^2 G$. Row (*a*) shows three images and row (*b*) shows their convolutions with the $\nabla^2 G$ filter of figure 2 ($w = 2\sigma = 8$), zero being represented by an intermediate grey. In row (*c*), positive values are shown white, and negative, black; and in row (*d*) only the zero-crossings appear.

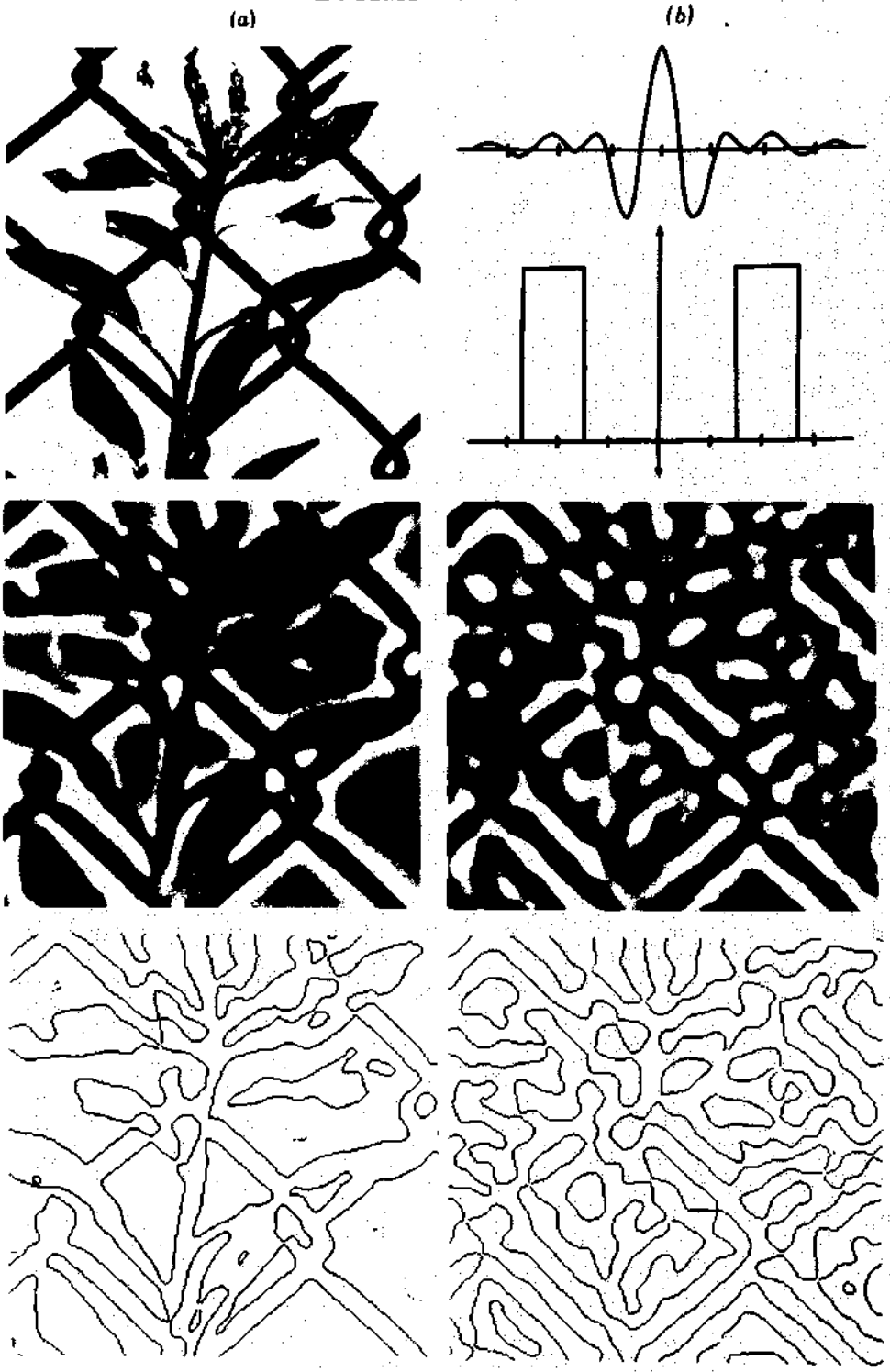
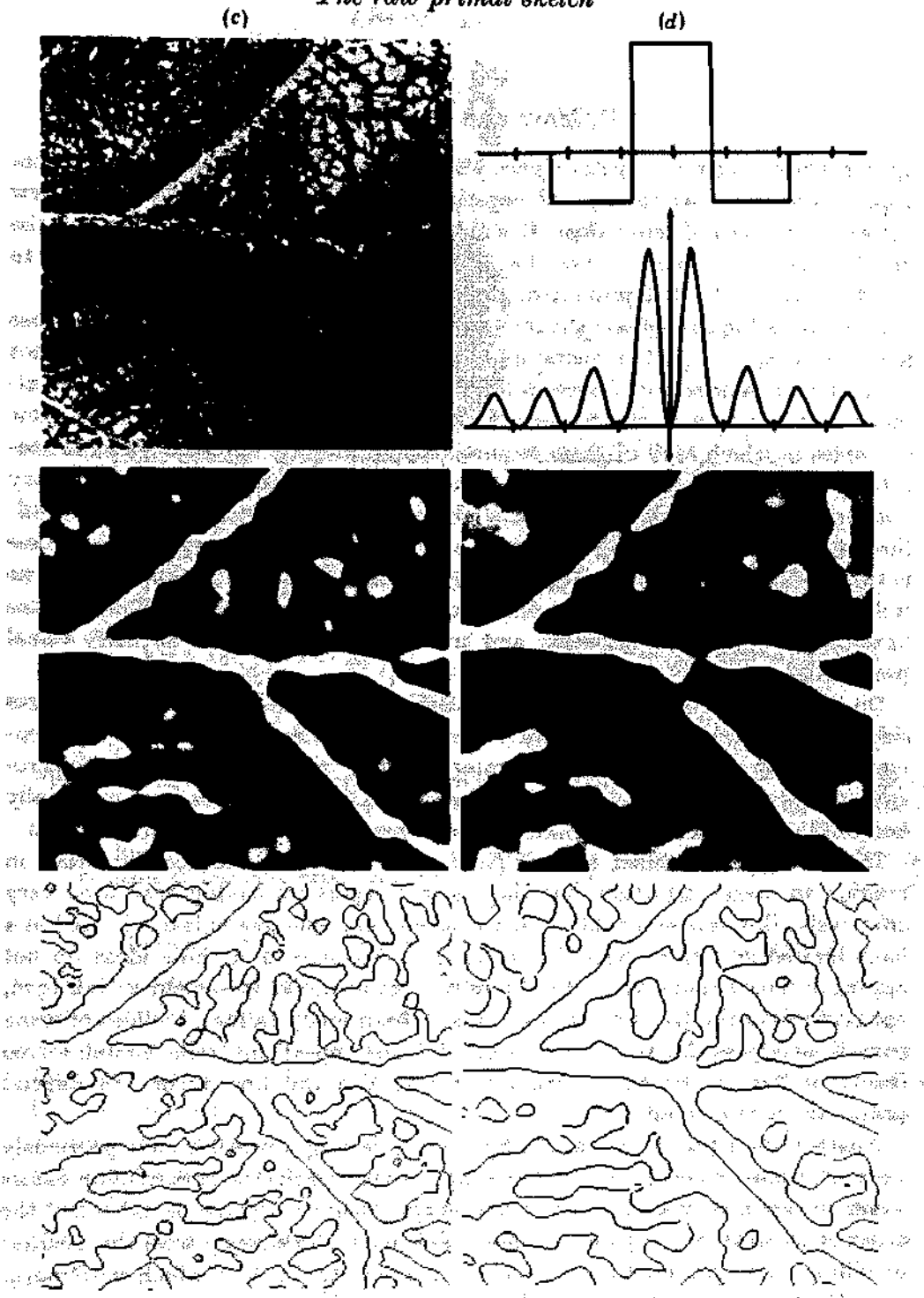


FIGURE 5. Comparison of the performance of $\nabla^2 G$ with that of similar filters. Column (*a*) shows an image, its convolution with $\nabla^2 G$ and the resulting signed zero-crossings. Column (*b*) contains the same sequence, but for the pure one-octave bandpass filter shown, with its Fourier transform, at the top of the column. The zero-crossing array contains echoes of the strong edges in the image. Columns (c) and (*d*) exhibit the same



analysis of another image, except that here, $\nabla^2 G$ is compared with a square-wave approximation to the second derivative. The widths of the central excitatory regions of the filters are the same for each comparison pair, being 12 for (*a*) and (*b*), and 18 for (*c*) and (*d*). The square-wave filter sees relatively few zero-crossings.

appear white, and very negative ones, black. In the third row, all positive values appear completely white, and all negative ones are black, and the fourth row shows just the loci of zero values. It will be observed that these delineate well the visible edges in the images. (See the legend for more details.) It remains only to break the zero value loci into oriented line segments.

It is interesting to compare the zero-crossings found by means of $\nabla^2 G$ with those found by means of similar operators that, according to our arguments, are not optimal. Our choice of the Gaussian filter was based on the requirements of simultaneous localization in the frequency and spatial domains. We therefore show examples in which each of these requirements is severely violated. An ideal one-octave bandpass filter satisfies the localization requirement in the frequency domain, but violates it in the spatial domain. The reason is that strict band-limiting gives rise to sidelobes in the spatial filter, and the consequence of these is that, in the zero-crossing image, strong intensity changes give rise to echoes as well as to the directly corresponding zero-crossings (see figure 5). These echoes have no direct physical correlate, and are therefore undesirable for early visual processing.

On the other hand, if one cuts off the filter in the spatial domain, one acquires sidelobes in the frequency domain. Figure 5 also shows a square-wave approximation to the second derivative operator, together with an example of the zero-crossings to which it gives rise. This operator sees fewer zero-crossings, essentially because it is averaging out the changes that occur over a wider range of scales.

Interestingly, Rosenfeld & Kak (1976, pp. 281–4) discuss the Laplacian in relation to 'edge' detection, but they do not report its having been used very effectively. One reason for this is that it is not very effective unless it is used in a band-limited situation and one uses its zero-crossings, and these ideas do not appear in the computer vision literature (see, for example, Rosenfeld & Kak 1976, fig. 10, for how the Laplacian has previously been used). In fact, the idea of using narrow bandpass differential operators did not appear until the human stereo theory of Marr & Poggio (1979), which was also the first theory to depend primarily on zero-crossings.

Another, more practical, reason why 'edge-detecting' operators have previously been less than optimally successful in computer vision is that most current operators examine only a very small part of the image, their 'receptive fields' are of the order of 10 to 20 image points at most. This contrasts sharply with the smallest of Wilson's four psychophysical channels, the receptive field of which must cover over 500 foveal cones (see figure 4).

Finally, notice that G'', and hence $\nabla^2 G$, is approximately a second derivative operator, because its Fourier transform is $-4\pi^2\omega^2 \exp(-\sigma^2\omega^2)$, which behaves like $-\omega^2$ near the origin.

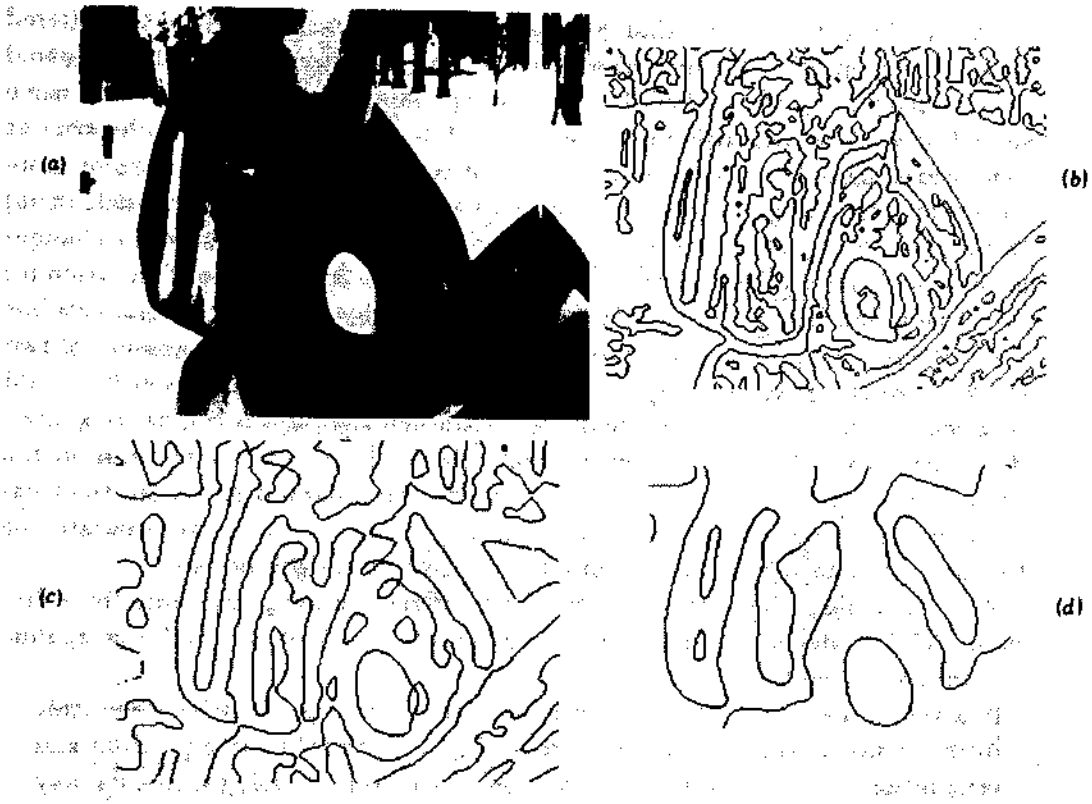


FIGURE 6. The image (*a*) has been convolved with $\nabla^2 G$ having $w = 2\sigma = 6$, 12 and 24 pixels. These filters span approximately the range of filters that operate in the human fovea. In (*b*), (*c*) and (*d*) are shown the zero-crossings thus obtained. Notice the fine detail picked up by the smallest. This set of figures neatly poses our next problem: how does one combine all this information into a single description?

COMBINING INFORMATION FROM DIFFERENT CHANNELS

The signals transmitted through channels that do not overlap in the Fourier domain will be generally unrelated unless the underlying signal is constrained. The critical question for us here is, therefore (and we are indebted to T. Poggio for conversations on this point), what additional information needs to be taken into account when we consider how to combine information from the different channels to form a primitive description of the image? In other words, are there any general physical constraints on the structure of the visual world that allow us to place valid restrictions on the way in which information from the different channels may be combined? Figure 6 illustrates the problem that we have to solve.

The spatial coincidence assumption

The additional information that we need here comes from the constraint of spatial localization, which we defined in the previous section. It states that the physical phenomena that give rise to intensity changes in the image are spatially

localized. Since it is these changes that produce zero-crossings in the filtered images, it follows that if a discernible zero-crossing is present in a channel centred on wavelength λ_0, there should be a corresponding zero-crossing at the same spatial location in channels for wavelengths $\lambda > \lambda_0$. If this ceases to be true at some wavelength $\lambda_1 > \lambda_0$, it will be for one of two reasons: either (*a*) two or more local intensity changes are being averaged together in the larger channel; or (*b*) two independent physical phenomena are operating to produce intensity changes in the same region of the image but at different scales. An example of situation (*a*) would be a thin bar, whose edges will be accurately located by small channels but not by large ones. Situations of this kind can be recognized by the presence of two nearby zero-crossings in the smaller channels. An example of situation (*b*) would be a shadow superimposed on a sharp reflectance change, and it can be recognized if the zero-crossings in the larger channels are displaced relative to those in the smaller. If the shadow has exactly the correct position and orientation, the locations of the zero-crossings may not contain enough information to separate the two physical phenomena, but, in practice, this situation will be rare.

We can therefore base the parsing of sets of zero-crossing segments from different $\nabla^2 G$ channels on the following assumption, which we call the *spatial coincidence assumption*:

> If a zero-crossing segment is present in a set of independent $\nabla^2 G$ channels over a contiguous range of sizes and the segment has the same position and orientation in each channel, then the set of such zero-crossing segments may may be taken to indicate the presence of an intensity change in the image that is due to a single physical phenomenon (a change in reflectance, illumination, depth or surface orientation).

In other words, provided that the zero-crossings from independent channels of adjacent sizes coincide, they can be taken together. If they do not, they probably arise from distinct surfaces or physical phenomena. It follows that the minimum number of channels required is two, and that provided the two channels are reasonably separated in the frequency domain, and their zero-crossings agree, the combined zero-crossings can be taken to indicate the presence of an edge in the image.

The parsing of sets of zero-crossing segments

Figure 6 shows the zero-crossings obtained from two channels whose dimensions are approximately the same as the two sustained channels present at the fovea in the human visual system (Wilson & Bergen 1979). We now derive the parsing rules needed for combining zero-crossings from the different channels.

Case (1): *isolated edges*

For an isolated, linearly disposed intensity change, there is a single zero-crossing present at the same orientation in all channels above some size that depends upon the channel sensitivity and the spatial extent of the edge. This set of zero-

crossings may, therefore, be combined into a symbol that we shall call an edge-segment, with the attributes of edge-amplitude and width, which we may obtain as follows.

Calculation of edge-amplitude. Because the assumptions that we have made mean that the type of intensity change involved is a simple one, we can, in fact, use what Marr (1976 figure 1) called the selection criterion, according to which one

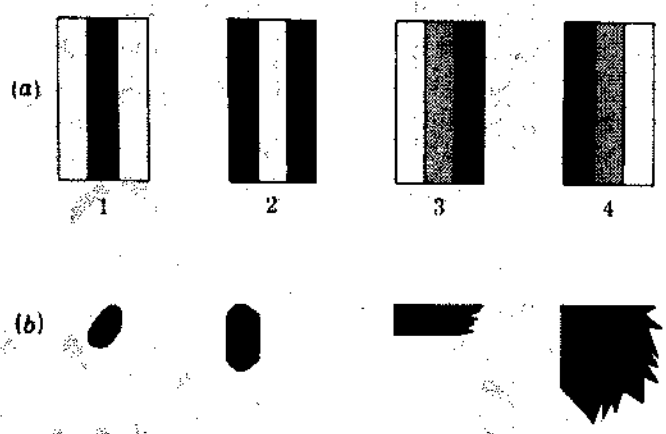


FIGURE 7. Parsing of sets of zero-crossing segments. (*a*) If zero-crossing segments lie close and roughly parallel (as in profile (*a*) of column 3 above), larger masks cannot be used, only the smaller masks. There are four possible configurations, shown in (1)–(4), and the figure represents the way in which the contrast changes across the edge. Each of these cases needs to be detected separately. (*b*) If the bar- or edge-segments are terminated, special descriptors are required. Doubly terminated bars, with $l \leqslant 3w$, are called *blobs* and the other assertions are labelled *terminations*. These are illustrated here for one contrast sign. Termination assertions may mark only a discontinuity in edge orientation, but it is often useful later on to have such positions explicitly available.

selects the smallest channel to which the intensity change is essentially indistinguishable from a step function, and uses that channel alone to estimate the contrast by means of the amplitude v derived above. If one has just two independent channels with amplitudes v_1 and v_2, an approximation to the edge amplitude is $\sqrt{(v_1^2 + v_2^2)}$.

Calculation of width. The *width* of the edge in this case can also be estimated from the channel selected according to the selection criterion. For a narrow channel with central wavelength λ, the physical notion of width corresponds to the distance over which intensity increases. This distance is $\frac{1}{2}\lambda$, which is approximately w, the width of the central excitatory region of the receptive field associated with the most excited channel (in fact, $\lambda = \pi w$).

Case (2): *bars*

If two parallel edges with opposite contrast lie only a small distance d apart in the image, zero-crossings from channels with associated wavelength that exceeds about $2d$ cannot be relied upon to provide accurate information about the positions

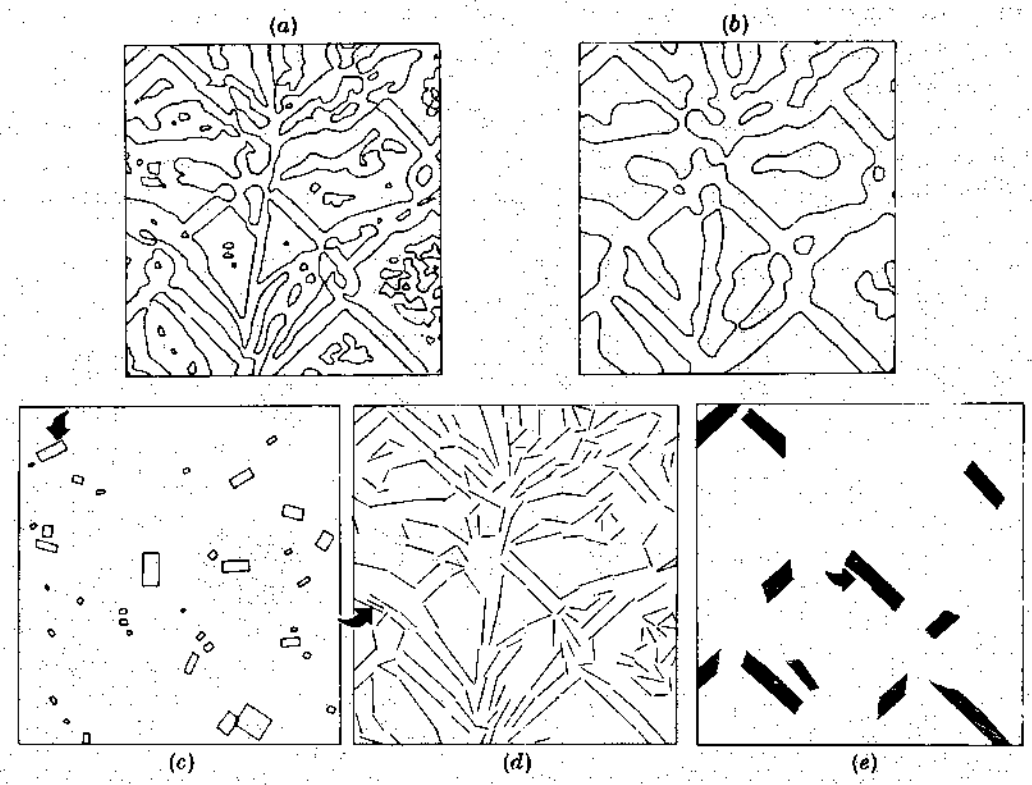


FIGURE 8. Combination of information from two channels. In (*a*) and (*b*) are shown the zero-crossings obtained from one of the images of figure 4, by means of masks with $w = 9$ and 18. Because there are no zero-crossings in the larger channel that do not correspond to zero-crossings in the smaller channel, the locations of the edges in the combined description also correspond to (*a*). In (*c*), (*d*) and (*e*) are shown symbolic representations of the descriptors attached to the locations marked in (*a*): (*c*) shows the blobs; (*d*), the local orientations assigned to the edge segments; and (*e*), the bars. These diagrams show only the spatial information contained in the descriptors. Typical examples of the full descriptors are as follows.

```
(BLOB (POSITION 146 21)      (EDGE (POSITION 104 23)      (BAR (POSITION 118 134)
      (ORIENTATION 105)            (ORIENTATION 120)           (ORIENTATION 120)
      (CONTRAST 76)                (CONTRAST -25)              (CONTRAST -25)
      (LENGTH 16)                  (LENGTH 25)                 (LENGTH 25)
      (WIDTH 6))                   (WIDTH 4))                  (WIDTH 4))
```

The descriptors to which these correspond are marked with arrows. The resolution of this analysis of the image of figure 4 roughly corresponds to what a human would see when viewing it from a distance of about 6 ft (1.83 m).

or contrasts of the edges. In these circumstances, the larger channels must be ignored, and the description formed solely from small channels of which the zero-crossing segments do superimpose. An edge can have either positive or negative contrast, and so two together give us the four situations shown in figure 7*a*. There is, of course, no reason why the two edges should have the same contrast, and the contrast of each edge must be obtained individually from the smallest channels

($w < d$). Two other parameters are useful; one is the average orientation of the two zero-crossing segments, and the other is their average separation.

Our case (2) applies only to situations in which neither zero-crossing segment terminates and they both remain approximately parallel (w or less apart). When the two edges are closer together than w for the smallest available channel, the zero-crossings associated with even the smallest channel will not accurately reflect the positions of the two edges, they will over estimate the distance between them. If the two edges have opposite contrasts that are not too different in absolute magnitude, the position of the centre of the 'line segment' so formed in the image will be the midpoint of the two corresponding zero-crossings. In these circumstances, the parameters associated with the line segment will be more reliable than those associated with each individual edge.

Case (3): *blobs and terminations*

It frequently happens that the zero-crossing segments do not continue very far across the image. Two parallel segments can merge, or be joined by a third segment, and in textured images they often form small closed curves (see figure 6), which are quite small compared to the underlying field size. Both situations can give rise to anomalous effects at larger channel sizes, and so are best made explicit early on. Following Marr (1976*b*), the closed contours we call BLOBS, and assign to them a length, width, orientation and (average) contrast; and the terminations are assigned a position and orientation (see figure 7*c*).

Remarks

Two interesting practical details have emerged from our implementation. First, the intensity changes at each edge of a bar are, in practice, rarely the same, so it is perhaps more proper to think of the BAR descriptor as a primitive grouping predicate that combines two edges the contrasts of which are specified precisely by the smallest channel. Brightness within the area of the bar will, of course, be constant. Secondly, it is often the case that the zero-crossings from the small and from the large masks roughly coincide, but those from the small mask weave around much more, partly because of the image structure and partly because of noise and the image tesselation. Local orientation has little meaning over distances shorter than the width w of the central excitatory region of the $\nabla^2 G$ filter, so if the zero-crossings from the smaller filter are changing direction rapidly locally, the orientation derived from the larger mask can provide a more stable and more reliable measure.

IMPLICATIONS FOR BIOLOGY

We have presented specific algorithms for the construction of the raw primal sketch, and we now ask whether the human visual system implements these algorithms or something close to them. There are two empirically accessible

characteristics of our scheme. The first concerns the underlying convolutions and zero-crossing segments, and the second, whether zero-crossing segments from the different channels are combined in the way that we have described.

Detection of zero-crossing segments

According to our theory, the most economical way of detecting zero-crossing segments requires that the image first be filtered through at least two independent $\nabla^2 G$ channels, and that the zero-crossings then be found in the filtered outputs. These zero-crossings may be divided into short, oriented zero-crossing segments.

The empirical data

Recent psychophysical work by Wilson & Giese (1977), Wilson & Bergen (1979) (see also Macleod & Rosenfeld 1974), has led to a precise quantitative model of the orientation-dependent spatial-frequency-tuned channels discovered by Campbell & Robson (1968). At each point in the visual field, there are four such channels spanning about three octaves, and their peak sensitivity wavelength increases linearly with retinal eccentricity. The larger two channels at each point are transient and the smaller two are sustained. These channels can be realized by linear units with bar-shaped receptive fields made of the difference of two Gaussian distributions, with excitatory to inhibitory space constants in the ratio of 1:1.75 for the sustained, and 1:3.0 for the transient, channels (Wilson & Bergen 1979). The largest receptive field at each point is about four times the smallest.

This state of affairs is consistent with the neurophysiology since Hubel & Wiesel (1962) originally defined simple cells by the linearity of their response, and they reported many bar-shaped receptive fields. In addition, simple cell receptive field sizes increase linearly with eccentricity (Hubel & Wiesel 1974, fig. 6*a*), and the scatter in size at each location seems to be about 4:1 (Hubel & Wiesel 1974, fig. 7). It is therefore tempting to identify at least some of the simple cells with the psychophysical channels. If so, the first obvious way of making the identification is to propose that the simple cells measure the second directional derivatives, thus perhaps providing the convolution values from which zero-crossing segments are subsequently detected.

There are, however, various reasons why this proposal can probably be excluded. They are:

(1) If the simple cells are essentially performing a linear convolution that approximates the second directional derivative, why are they so orientation sensitive? Three measurements, in principle, suffice to characterize the second derivative completely and, in practice, the directional derivatives measured along four orientations are apparently enough for this stage (see Marr 1976*b*; Hildreth, in preparation), and yet simple cells divide the domain into about 12 orientations.

(2) Schiller *et al.* (1976*b*, pp. 1324–5) found that the orientation sensitivity of simple cells is relatively independent of the strength of flanking inhibition, and

of the separation and lengths of the positive and negative subfields of the receptive field of the cell. In addition, tripartite receptive fields did not appear to be more orientation sensitive than bipartite ones. These points provide good evidence that simple cells are not linear devices.

(3) If the simple cells perform the convolution, what elements find the zero-crossings and implement the spatial part of the computation, lining the zero-crossings up with the convolution orientations, for example?

Wilson's channel data is consistent with $\nabla^2 G$

Wilson's DOG functions are very similar to $\nabla^2 G$, and probably indistinguishable by means of his experimental technique, which yields about 10% accuracy (H. G. Wilson, personal communication). In appendix B, we show: (*a*) that $\nabla^2 G$ is the limit of the DOG function as σ_1/σ_e, the ratio of the inhibitory to excitatory space constants, tends to unity; and (*b*) that if an approximation to $\nabla^2 G$ is to be constructed out of the difference of two Gaussian distributions, one excitatory and the other inhibitory, the optimal choice on engineering grounds for σ_1/σ_e is about 1.6.

A specific proposal: lateral geniculate X*-cells carry* $\nabla^2 G * I$*, and some simple cells detect and represent zero-crossing segments*

It is known that retinal ganglion X-cells have receptive fields that are accurately described by the difference of two Gaussian distributions (Rodieck & Stone 1965; Ratliff 1965; Enroth-Cugell & Robson 1966). The positive and negative parts are not quite balanced (there is a response to diffuse illumination and it increases with intensity), and since the ganglion cells have a spontaneous resting discharge, they signal somewhat more than just the positive or just the negative part of such a convolution. Interestingly, there is little scatter in receptive field sizes of X-cells at a given location in the retina (Peichl & Wässle 1979).

There is some controversy about the way in which lateral geniculate receptive fields are constructed (cf. Maffei & Fiorentini 1972*b*), but it seems most likely that the on-centre geniculate X-cell fields are formed by combining a small number of on-centre retinal ganglion X-cell fields of which the centres approximately coincide (Cleland *et al.* 1971). It seems likely that the scatter in receptive field size arises in this way, since the amount of scatter required to account for the psychophysical findings is only a factor of two in both the X and the Y channels. Finally, lateral geniculate cells give a smaller response to diffuse illumination than do retinal ganglion cells, sometimes giving no response at all (Hubel & Wiesel 1961).

These facts lead us to a particularly attractive scheme, which, for simplicity, we present in idealized form.

(1) *Measurement of* $\nabla^2 G$. The sustained, or X-cell, geniculate fibres can be thought of as carrying either the positive or the negative part of $\nabla^2 G * I$, where the filter $\nabla^2 G$ of figure 2 is, in practice, approximated by a difference

of Gaussian convolution operator with centre-to-surround space constants in the ratio 1 : 1.75. (One should probably think of this as being a convolution on linear intensity values, rather than on their logarithms. The reason for this is that although the nerve signal in the retina is an adaptation term multiplied by $I/(I+K)$, where I is the incident illumination and $K = 800$ quanta per receptor per second (Alpern *et al.* 1970), in any given image the ratio of the darkest to the brightest portion rarely exceeds 25 (a local ratio of around

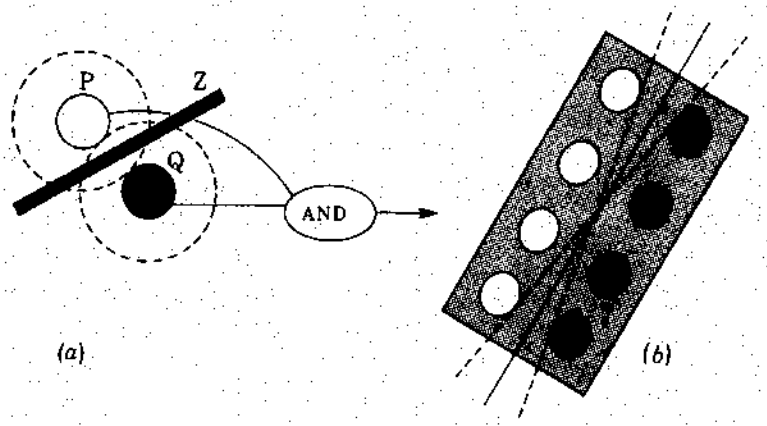


FIGURE 9. Proposed mechanism whereby some simple cells detect zero-crossing segments. In (*a*), if P represents an on-centre geniculate X-cell receptive field, and Q, an off-centre one, then if both are active, a zero-crossing Z in the Laplacian passes between them. If they are connected to a logical AND gate, as shown, then the gate will 'detect' the presence of the zero-crossing. If several are arranged in tandem, as in (*b*), and also connected by logical ANDs, the resulting operation detects an oriented zero-crossing segment within the orientation bounds given roughly by the dotted lines. This gives our most primitive model for simple cells. Ideally, one would like gates such that there is a response only if all (P, Q) inputs are active, and the magnitude of the response then varies with their sum. Marr & Ullman (1979) extend this model to include directional selectivity.

30 is seen as a light source (Ullman 1976)), and over such ranges this function does not depart far from linearity.) At each point in the visual field, there are two sizes of filter (the minimum required for combining zero-crossings between channels), and these correspond to Wilson & Bergen's (1979) N and S channels. The one-dimensional projection of the widths w of the central excitatory regions of these two channels scales linearly with eccentricity from 3.1′ and 6.2′ at the central fovea.

The basic idea behind our model for the detection of zero-crossings rests on the following observations: if an on-centre geniculate cell is active at location P and an off-centre cell is active at nearby location Q, then the value of $\nabla^2 G * I$ passes through zero between P and Q (see figure 9*a*). Hence, by combining the signals from P and Q through a logical AND operation, one can construct an operator for detecting when a zero-crossing segment (at some unknown orientation) passes

between P and Q (figure 9*a*). By adding nonlinear AND operations in the longitudinal direction, one can, in a similar way, construct an operator that detects oriented zero-crossing segments. It is easy to see that the pure logical operator of figure 9*b* will respond only to zero-crossing segments whose orientations lie within its sensitivity range (shown roughly dotted). We therefore propose:

(2) *Detection and representation of zero-crossing segments.* Part of the function of one subclass of simple cells is to detect zero-crossing segments. Their receptive fields include the construction shown in figure 9*b*, with the proviso that the non-linearities may be weaker than the pure logical ANDs shown there. It is, however, a critical feature of this model that the (P AND Q) interaction (figure 9*a*) across the zero-crossing segment should contain a strong nonlinear component and that the longitudinal interaction (e.g. between the ends in figure 9*b*) contains at least a weak nonlinear component. Marr & Ullman's (1979) full model for simple cells contains this organization, but includes additional machinery for detecting the direction of movement of the zero-crossing segment, and it is this that provides a role for the two larger transient channels.

(3) *Signalling amplitude.* Ideally, the output of the cell should be gated by the logical AND function of (2), but its value should be the average local amplitude ν associated with the zero-crossings along the segment. As we saw earlier, this may be found by measuring the average local value of the slope of the zero-crossings, which (in suitable units) is equal to the sum of the inputs to the cell.

(4) *Sampling density.* Finally, for this scheme to be successful, the sampling density of the function $\nabla^2 G * I$ must be great enough to ensure that the zero-crossings may subsequently be localized accurately enough to account for the findings about hyperacuity (see, for example, Westheimer & McKee 1977), which means roughly to within 5′. This implies an extremely high precision of representation, but in layer IV of the monkey's striate cortex, there apparently exists a myriad of small, centre–surround, non-oriented cells (Hubel & Wiesel 1968). Barlow (1979) and Crick *et al.* (1980) have suggested that these cells may be involved in the reconstruction of the $\nabla^2 G$ function to an adequate precision for hyperacuity.

The empirical consequences of this overall scheme are set out by Marr & Ullman (1979).

Combination of zero-crossings

Empirical predictions for psychophysics

There are several aspects of our algorithm, for combining zero-crossings from different channels, that are accessible to psychophysical experiment. They are: (*a*) the phase relations; (*b*) combination of zero-crossings from different channels, and (*c*) the special cases that arise when zero-crossings lie close to one another.

(1) *Phase relations.* Our theory predicts that descriptors need exist only for sets

of zero-crossings, from different channels, that coincide spatially (i.e. have a phase relation of 0 or π). Interestingly, Atkinson & Campbell (1974) superimposed 1 and 3 cycles/deg sinusoidal gratings of the same orientation, and found that the number of perceptual fluctuations per minute (which they called rate of monocular rivalry) was low near the in-phase, 0, and out-of-phase, π, positions, but reached a high plateau for intermediate phase positions. They concluded (p. 161) that the visual system contains a device that 'seems to be designed to respond only to 0 and π phase relation. When ...[it] .... is active, it gives rise to a stable percept that is the sum of the two spatial frequency selective channels' (cf. also Maffei & Fiorentini 1972*a*). Our theory would predict these results, if the additional assumption were made that units exist that represent explicitly the edge segment descriptor formed by combining appropriately arranged zero-crossing segments.

(2) *The parsing process.* The main point here is that the description of an edge (its width, amplitude and orientation) can be obtained from the (smallest) channel whose zero-crossing there has maximum slope. As Marr (1976*b*, pp. 496–497) observed, this is consistent with Harmon & Julesz's (1973) finding that noise bands spectrally adjacent to the spectrum of a picture are most effective at suppressing recognition, since these have their greatest effect on mask response amplitudes near the important mask sizes. It also explains why removal of the middle spatial frequencies from such an image leaves a recognizable image of Lincoln behind a visible graticule (see Harmon & Julesz 1973). The reason is that the zero-crossings from different mask sizes fail to coincide, and the gap in the spectrum means that the small bar descriptors fail to account for this discrepancy. Hence, the assumption of spatial coincidence cannot be used, and the outputs from the different mask sizes are assumed to be due to different physical phenomena. Accordingly, they give rise to independent descriptions.

There is another possible but weaker consequence. If one makes the extra assumption, that the selection criterion is implemented by inhibitory connections between zero-crossing segment detectors that are spatially coincident and lying adjacent in the frequency domain, then one would expect to find an inhibitory interaction between channels at the cortical, orientation-dependent level. There is, in fact, evidence that this occurs (see, for example, Tolhurst 1972; de Valois 1977*a*).

(3) *Bar-detectors.* Case (2) of our parsing algorithm requires the specific detection of close, parallel, zero-crossing segments. This requires the existence of units sensitive, at each orientation, to one of the four cases (black bar, white bar, two dark edges, two light edges) and sensitive to their width (i.e. the distance separating the edges) rather than to spatial frequency characteristics of the whole pattern. Adaptation studies that lead to these conclusions for white bars and for black bars have recently been published (Burton *et al.* 1977; de Valois 1977*b*). If our algorithm is implemented by the human visual system, the analogous result should hold for the remaining two cases (see figure 7*a*).

(4) *Blob-detectors and terminations.* Case (3) of our parsing algorithm requires

the explicit representation of (oriented) blobs and terminations. Units that represent them should be susceptible to psychophysical adaptation, and, in fact, Nakayama & Roberts (1972) and Burton & Ruddock (1978) have found evidence for units that are sensitive to bars whose length does not exceed three times the width.

Consequences for neurophysiology

There are several ways of implementing the parsing process that we have described, but it is probably not worth setting them out in detail until we have good evidence from psychophysics about the parsing algorithm that is actually used and we know whether simple cells, in fact, implement the detection of zero-crossing segments. Without these pieces of information firm predictions cannot be made, but we offer the following suggestions as a possible framework for the neural implementation. (1) The four types of 'bar' detectors could be implemented at the very first, simple cell level (along the lines of figure 9, but being fed by three rows of centre–surround cells instead of two). (2) For relatively isolated edges, there should exist oriented edge-segment–detecting neurons that combine zero-crossing segment detectors (simple cells) from different channels when, and only when, the segments are spatially coincident. (3) Detectors for terminations and blobs (doubly-terminated oriented bars) seem to have been found already (Hubel & Wiesel 1962, 1968). Interestingly, Schiller *et al.* (1976*a*) found that even some simple cells are stopped. Our scheme is consistent with this since it requires such detectors at a very early stage.

Discussion

The concept of an 'edge' has a partly visual and partly physical meaning. One of our main purposes in this article is to make explicit this dual dependence: our definition of an edge rests lightly on the early assumptions of theorem 1 about directional derivatives and heavily on the constraint of spatial localization.

Our theory is based on two main ideas. First, one simplifies the detection of intensity changes by dealing with the image separately at different resolutions. The detection process can then be based on finding zero-crossings in a second derivative operator, which, in practice, can be the (non-oriented) Laplacian. The representation at this point consists of zero-crossing segments and their slopes. This representation is probably complete and is, therefore, in principle, invertible. This had previously been given only an empirical demonstration by Marr and by R. Woodham (see Marr 1978, fig. 7).

The subsequent step, of combining information from different channels into a single description, rests on the second main idea of the theory, which we formulated as the spatial coincidence assumption. Physical edges will produce roughly coincident zero-crossings in channels of nearby sizes. The spatial coincidence assumption asserts that the converse of this is true, that is the coincidence of zero-

crossings is sufficient evidence for the existence of a real physical edge. If the zero-crossings in one channel are not consistent with those in the others, they are probably caused by different physical phenomena, so descriptions need to be formed from both sources and kept somewhat separate.

Finally, the basic idea, that some simple cells detect and represent zero-crossing segments and that this is carried out simultaneously at different scales, has some implications for Marr & Poggio's (1979) stereo theory. According to various neurophysiological studies (Barlow *et al.* 1967; Poggio & Fischer 1978; von der Heydt *et al.* 1978), there exist disparity sensitive simple cells. The existence of such cells is consistent with our suggestion that they detect zero-crossing segments, but not with the idea that they perform a linear convolution equivalent to a directional derivative, since it is the primitive symbolic descriptions provided by zero-crossing segments that need to be matched between images, not the raw convolution values.

We thank K. Nishihara, T. Poggio and S. Ullman for their illuminating and helpful comments. This work was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defence and monitored by the Office of Naval Research, under contract number N00014-75-C-0643. D. M. was also supported by N.S.F. contract number 77-07569-MCS.

References

Alpern, M., Rushton, W. A. H. & Torii, S. 1970 The size of rod signals. *J. Physiol., Lond.* **206**, 193–208.

Atkinson, J. & Campbell, F. W. 1974 The effect of phase on the perception of compound gratings. *Vision Res.* **14**, 159–162.

Barlow, H. B. 1969 Pattern recognition and the responses of sensory neurons. *Ann. N.Y. Acad. Sci.* **156**, 872–881.

Barlow, H. B. 1979 Reconstructing the visual image in space and time. *Nature, Lond.* **279**, 189–190.

Barlow, H. B., Blakemore, C. & Pettigrew, J. D. 1967 The neural mechanism of binocular depth discrimination. *J. Physiol., Lond.* **193**, 327–342.

Bracewell, R. 1965 *The Fourier transform and its applications*. New York: MacGraw-Hill.

Burton, G. J., Nagshineh, S. & Ruddock, K. H. 1977 Processing by the human visual system of the light and dark contrast components of the retinal image. *Biol. Cybernetics* **28**, 1–9.

Burton, G. J. & Ruddock, K. H. 1978 Visual adaptation to patterns containing two-dimensional spatial structure. *Vision Res.* **18**, 93–99.

Campbell, F. W. & Robson, J. G. 1968 Applications of Fourier analysis to the visibility of gratings. *J. Physiol., Lond.* **197**, 551–556.

Cleland, B. G., Dubin, M. W. & Levick, W. R. 1971 Sustained and transient neurones in the cat's retina and lateral geniculate nucleus. *J. Physiol., Lond.* **217**, 473–496.

Cowan, J. D. 1977 Some remarks on channel bandwidths for visual contrast detection. *Neurosci. Res. Prog. Bull.* **15**, 492–517.

Crick, F. H. C., Marr, D. & Poggio, T. 1980 An information processing approach to understanding the visual cortex. To appear in the N.R.P. symposium *The cerebral cortex* (ed. F. O. Schmidt & F. G. Worden).

De Valois, K. K. 1977*a* Spatial frequency adaptation can enhance contrast sensitivity. *Vision Res.* **17**, 1057–1065.

De Valois, K. K. 1977*b* Independence of black and white: phase-specific adaptation. *Vision Res.* **17**, 209–215.

Enroth-Cugell, C. & Robson, J. G. 1966 The contrast sensitivity of retinal ganglion cells of the cat. *J. Physiol., Lond.* **187**, 517–552.

Graham, N. 1977 Visual detection of aperiodic spatial stimuli by probability summation among narrowband channels. *Vision Res.* **17**, 637–652.

Harmon, L. D. & Julesz, B. 1973 Masking in visual recognition: effects of two-dimensional filtered noise. *Science N.Y.* **180**, 1194–1197.

von der Heydt, R., Adorjani, Cs., Hanny, P. & Baumgartner, G. 1978 Disparity sensitivity and receptive field incongruity of units in the cat striate cortex. *Exp. Brain Res.* **31**, 523–545.

Hubel, D. H. & Wiesel, T. N. 1961 Integrative action in the cat's lateral geniculate body. *J. Physiol., Lond.* **155**, 385–398.

Hubel, D. H. & Wiesel, T. N. 1962 Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol., Lond.* **160**, 106–154.

Hubel, D. H. & Wiesel, T. N. 1968 Receptive fields and functional architecture of monkey striate cortex. *J. Physiol., Lond.* **195**, 215–243.

Hubel, D. H. & Wiesel, T. N. 1974 Uniformity of monkey striate cortex: a parallel relationship between field size, scatter, and magnification factor. *J. comp. Neurol.* **158**, 295–306.

Kulikowski, J. J. & King-Smith, P. E. 1973 Spatial arrangement of line, edge, and grating detectors revealed by subthreshold summation. *Vision Res.* **13**, 1455–1478.

Leipnik, R. 1960 The extended entropy uncertainty principle. *Inf. Control* **3**, 18–25.

Logan, B. F. Jr 1977 Information in the zero-crossings of bandpass signals. *Bell Syst. tech. J.* **56**, 487–510.

Macleod, I. D. G. & Rosenfeld, A. 1974 The visibility of gratings: spatial frequency channels or bar-detecting units? *Vision Res.* **14**, 909–915.

Maffei, L. & Fiorentini, A. 1972*a* Process of synthesis in visual perception. *Nature, Lond.* **240**, 479–481.

Maffei, L. & Fiorentini, A. 1972*b* Retinogeniculate convergence and analysis of contrast. *J. Neurophysiol.* **35**, 65–72.

Maffei, L. & Fiorentini, A. 1977 Spatial frequency rows in the striate visual cortex. *Vision Res.* **17**, 257–264.

Marr, D. 1970 A theory for cerebral neocortex. *Proc. R. Soc. Lond.* B **176**, 161–234.

Marr, D. 1976*a* Analyzing natural images: a computational theory of texture vision. *Cold Spring Harbor Symp. quant. Biol.* **40**, 647–662.

Marr, D. 1976*b* Early processing of visual information. *Phil. Trans. R. Soc. Lond.* B **275**, 483–524.

Marr, D. 1978 Representing visual information. A.A.A.S. 143rd Annual Meeting, Symposium on: Some mathematical questions in biology, February 1977. Published in *Lectures on mathematics in the life sciences* **10**, 101–180. Also available as *M.I.T A.I. Lab. Memo* 415.

Marr, D. & Poggio, T. 1979 A computational theory of human stereo vision. *Proc. R. Soc. Lond.* B **204**, 301–328.

Marr, D., Poggio, T. & Ullman, S. 1979 Bandpass channels, zero-crossings, and early visual information processing. *J. opt. Soc. Am.* **69**, 914–916.

Marr, D. & Ullman, S. 1979 Directional selectivity and its use in early visual processing. (In preparation.)

Mayhew, J. E. W. & Frisby, J. P. 1978 Suprathreshold contrast perception and complex random textures. *Vision Res.* **18**, 895–897.

Nakayama, K. & Roberts, D. J. 1972 Line-length detectors in the human visual system: evidence from selective adaptation. *Vision Res.* **12**, 1709–1713.

Peichl, L. & Wässle, H. 1979 Size, scatter and coverage of ganglion cell receptive field centres in the cat retina. *J. Physiol., Lond.* **291**, 117–141.

Poggio, G. F. & Fischer, B. 1978 Binocular interaction and depth sensitivity of striate and prestriate neurons of the behaving rhesus monkey. *J. Neurophysiol.* **40**, 1392–1405.

Pollen, D. A., Lee, J. R. & Taylor, J. H. 1971 How does the striate cortex begin the reconstruction of the visual world? *Science N.Y.* **173**, 74–77.

Ratliff, F. 1965 *Mach bands: quantitative studies on neural networks in the retina.* San Francisco: Holden-Day.

Rodieck, R. W. & Stone, J. 1965 Analysis of receptive fields of cat retinal ganglion cells. *J. Neurophysiol.* **28**, 833–849.

Rosenfeld, A. & Kak, A. C. 1976 *Digital picture processing.* New York: Academic Press.

Sachs, M. B., Nachmias, J. & Robson, J. G. 1971 Spatial-frequency channels in human vision. *J. opt. Soc. Am.* **61**, 1176–1186.

Shapley, R. M. & Tolhurst, D. J. 1973 Edge detectors in human vision. *J. Physiol., Lond.* **229**, 165–183.

Schiller, P. H., Finlay, B. L. & Volman, S. F. 1976*a* Quantitative studies of single-cell properties in monkey striate cortex. I. Spatiotemporal organization of receptive fields. *J. Neurophysiol.* **39**, 1288–1319.

Schiller, P. H., Finlay, B. L. & Volman, S. F. 1976*b* Quantitative studies of single-cell properties in monkey striate cortex. II. Orientation specificity and ocular dominance. *J. Neurophysiol.* **39**, 1320–1333.

Ullman, S. 1976 On visual detection of light sources. *Biol. Cybernetics* **21**, 205–212.

Westheimer, G. & McKee, S. P. 1977 Spatial configurations for visual hyperacuity. *Vision Res.* **17**, 941–947.

Wilson, H. R. & Bergen, J. R. 1979 *A four mechanism model for spatial vision. Vision Res.* **19**, 19–32.

Wilson, H. R. & Giese, S. C. 1977 Threshold visibility of frequency gradient patterns. *Vision Res.* **17**. 1177–1190.

Appendix A

Theorem 1

Let l be an open line segment of the y-axis, containing the origin O. Suppose that $f(x, y)$ is twice continuously differentiable and that $N(l)$ is an open two-dimensional neighbourhood of l. Assume that $\partial^2 f/\partial x^2 = 0$ on l. Then, if $\partial f/\partial y$ is constant in $N(l)$, the slope of the second directional derivative taken perpendicular to l (i.e., the slope of $\partial^2 f/\partial x^2$) is greater than the slope of the zero-crossing along any other line through O.

Proof

Consider the line segment $\Omega = (r\cos\theta,\ r\sin\theta)$ for fixed θ and values of r sufficiently small that Ω lies entirely within $N(l)$ (see figure 10). Now writing f_{xx} for $\partial^2 f/\partial x^2$ etc., we have

$$\begin{aligned}(\partial^2 f/\partial\Omega^2)_{r,\theta} &= (f_{xx}\cos^2\theta + f_{xy}\,2\sin\theta\cos\theta + f_{yy}\sin^2\theta)_{r,\theta}\\ &= (f_{xx}\cos^2\theta)_{r,\theta},\end{aligned}$$

since the condition of the theorem that f_y be constant implies that f_{xy} and f_{yy} are both zero. As required, therefore, the above quantity is zero at $r = 0$ and has maximum slope when $\theta = 0$.

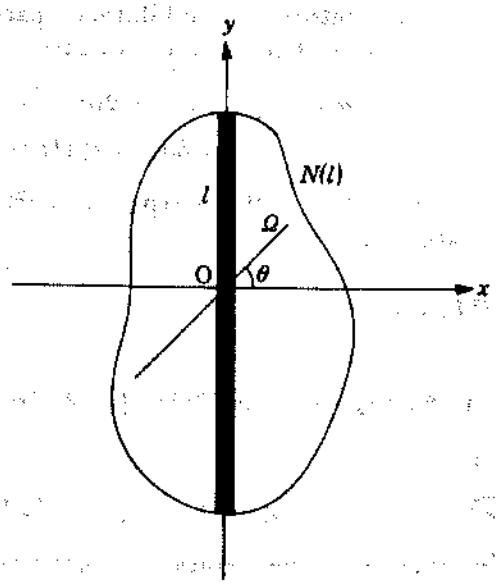


FIGURE 10. Diagram for theorems 1 and 2: l is a segment of the y-axis, containing the origin; $N(l)$ is a neighbourhood of it. Provided that $\partial f/\partial y$ is constant in $N(l)$, theorem 1 states that the orientation of the line of zero-crossings is perpendicular to the orientation at which the zero-crossings have maximum slope.

THEOREM 2

Let $f(x, y)$ be a real-valued, twice continuously differentiable function on the plane. Let l be an open line segment along the axis $x = 0$. Then the two conditions

$$\text{(i)}\ \nabla^2 f = 0 \text{ on } l$$

$$\text{and (ii)}\ \partial^2 f/\partial x^2 = 0 \text{ on } l$$

are equivalent if and only if $f(0, y)$ is constant or linear on l.

Proof

If $f(0, y)$ is linear on l, $\partial^2 f/\partial y^2 = 0$ on l. Hence, $\nabla^2 f = 0$ there implies that $\partial^2 f/\partial x^2 = 0$ on l too.

Conversely, if $\partial^2 f/\partial x^2 = \nabla^2 f = 0$ on l, then $\partial^2 f/\partial y^2 = 0$ on l, and so $f(0, y)$ varies at most linearly on l.

APPENDIX B

DOGS *and* $\nabla^2 G$

$\nabla^2 G$ is the limit of a DOG

Wilson's *DOG* function may be written

$$DOG(\sigma_e, \sigma_1) = [1/(2\pi)^{\frac{1}{2}}\sigma_e] \exp(-x^2/2\sigma_e^2) - [1/(2\pi)^{\frac{1}{2}}\sigma_1] \exp(-x^2/2\sigma_1^2), \quad (3)$$

where σ_e and σ_i are the excitatory and inhibitory space constants. Writing $\sigma_e = \sigma$, and $\sigma_i = \sigma + \delta\sigma$, the right hand side varies with

$$(1/\sigma)\exp(-x^2/2\sigma^2) - [1/(\sigma+\delta\sigma)]\exp[-x^2/2(\sigma+\delta\sigma)^2] = \delta\sigma\,(\partial/\partial\sigma)\,(1/\sigma\exp[-x^2/2\sigma^2]). \quad (4)$$

This derivative is equal to $-(1/\sigma^2 - x^2/\sigma^4)\exp(-x^2/2\sigma^2)$, which equals G'' up to a constant (text equation 5).)

Approximation of $\nabla^2 G$ by a DOG

The function

$$\text{DOG}(\sigma_e, \sigma_i) = [1/(2\pi)^{\frac{1}{2}}\sigma_e]\exp(-x^2/2\sigma_e^2) - [1/(2\pi)^{\frac{1}{2}}\sigma_i]\exp(-x^2/2\sigma_i^2) \quad (5)$$

has Fourier transform

$$\widetilde{\text{DOG}}(\omega) = \exp(-\sigma_e^2\omega^2/2) - \exp(-\sigma_i^2\omega^2/2) \quad (6)$$

Notice that $\widetilde{\text{DOG}}(\omega)$ behaves like ω^2 for values of ω that are small compared with σ_e and σ_i, so that these filters, in common with $\nabla^2 G$, approximate a second derivative operator.

The problem with using a DOG to approximate $\nabla^2 G$ is to find a space constant that keeps the bandwidth of the filter small and yet allows the filter adequate sensitivity: for, clearly, as the space constants approach one another, the contributions of the excitatory and inhibitory components become identical and the sensitivity of the filter is reduced.

The bandwidths at half sensitivity and at half power and the peak sensitivity all depend together on the value of σ_i/σ_e in a way that is shown in figure 11. From this we see that: (i) the bandwidth at half sensitivity increases very slowly up to about $\sigma_i/\sigma_e = 1.6$, increases faster from there to $\sigma_i/\sigma_e = 3.0$, and is thereafter approximately constant; (ii) the peak sensitivity of the filter is desultory for small σ_i/σ_e, reaching about 33% at $\sigma_i/\sigma_e = 1.6$. Since our aim is to create a narrow bandpass differential operator, we should choose σ_i/σ_e to minimize the bandwidth. Since the bandwidth is approximately constant for $\sigma_i/\sigma_e < 1.6$, and since sensitivity is low there, the minimal value one would in practice choose for σ_i/σ_e is around 1.6, giving a half-sensitivity bandwidth of 1.8 octaves and a half power bandwidth of 1.3 octaves.

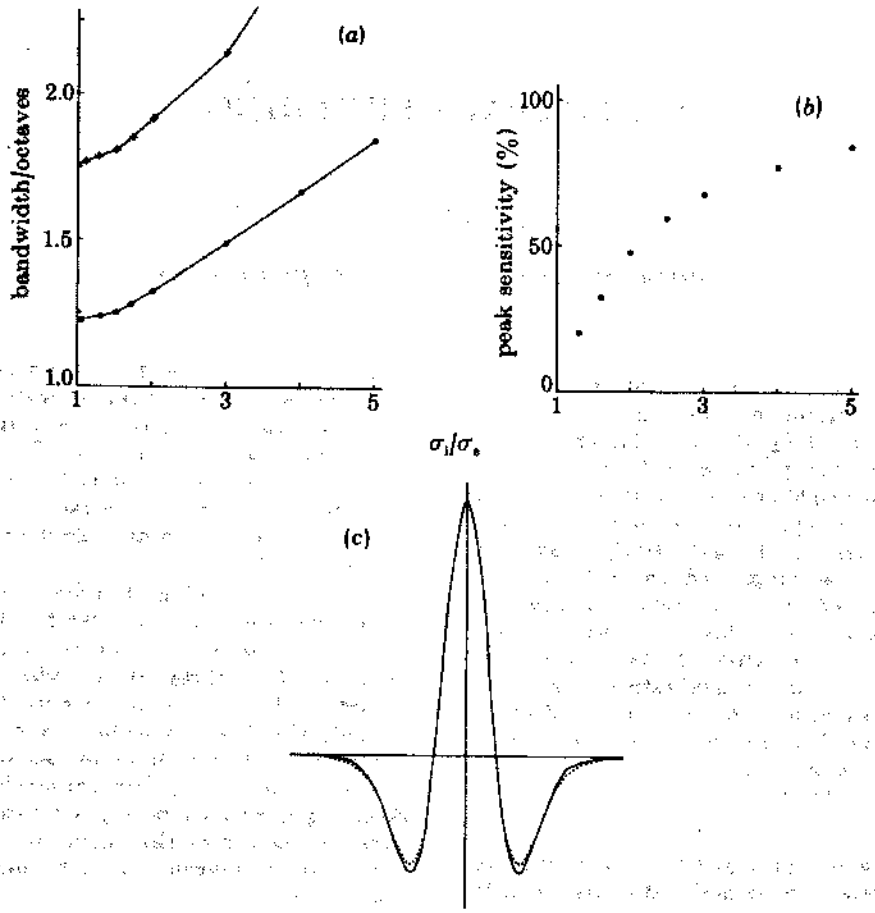


FIGURE 11. The values of certain parameters associated with difference-of-Gaussian (DOG) masks, with excitatory and inhibitory space constants σ_e and σ_i. (*a*) For various values of σ_e/σ_i, we show the half-sensitivity bandwidth (+) and the half-power bandwidth (●) of the filter. In (*b*) is shown its peak sensitivity in the Fourier plane. (The peak sensitivity of the excitatory component alone equals 100% on this scale.) (*c*) The arguments in the appendix show that the best engineering approximation to $\nabla^2 G$ using a DOG occurs with σ_i/σ_e around 1.6. In figure (c), this particular DOG is shown dotted against the operator $\nabla^2 G$ with the appropriate σ. The two profiles are very similar.

"Scale-space Filtering" by A.P. Witkin from *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, August 1983, pages 1019-1022, Volume 2.

SCALE-SPACE FILTERING

Andrew P. Witkin

Fairchild Laboratory for Artificial Intelligence Research

ABSTRACT—The extrema in a signal and its first few derivatives provide a useful general-purpose qualitative description for many kinds of signals. A fundamental problem in computing such descriptions is scale: a derivative must be taken over some neighborhood, but there is seldom a principled basis for choosing its size. Scale-space filtering is a method that describes signals qualitatively, managing the ambiguity of scale in an organized and natural way. The signal is first expanded by convolution with gaussian masks over a continuum of sizes. This "scale-space" image is then collapsed, using its qualitative structure, into a tree providing a concise but complete qualitative description covering all scales of observation. The description is further refined by applying a stability criterion, to identify events that persist of large changes in scale.

1. Introduction

Hardly any sophisticated signal understanding task can be performed using the raw numerical signal values directly; some description of the signal must first be obtained. An initial description ought to be as compact as possible, and its elements should correspond as closely as possible to meaningful objects or events in the signal-forming process. Frequently, local extrema in the signal and its derivatives—and intervals bounded by extrema—are particularly appropriate descriptive primitives: although local and closely tied to the signal data, these events often have direct semantic interpretations, e.g. as edges in images. A description that characterizes a signal by its extrema and those of its first few derivatives is a *qualitative* description of exactly the kind we were taught to use in elementary calculus to "sketch" a function.

A great deal of effort has been expended to obtain this kind of primitive qualitative description (for overviews of this literature, see [1,2,3].) and the problem has proved extremely difficult. The problem of *scale* has emerged consistently as a fundamental source of difficulty, because the events we perceive and find meaningful vary enormously in size and extent. The problem is not so much to eliminate fine-scale noise, as to separate events at different scales arising from distinct physical processes.[4] It is possible to introduce a *parameter of scale* by smoothing the signal with a mask of variable size, but with the introduction of scale-dependence comes ambiguity: every setting of the scale parameter yields a different description; new extremal points may appear, and existing ones may move or disappear. How can we decide which if any of this continuum of descriptions is "right"?

There is rarely a sound basis for setting the scale parameter. In fact, it has become apparent that for many tasks no one scale of description is categorically correct: the physical processes that generate signals such as images act at a variety of scales, none intrinsically more interesting or important than another. Thus the ambiguity introduced by scale is inherent and inescapable, so the goal of scale-dependent description cannot be to eliminate this ambiguity, but rather to manage it effectively, and reduce it where possible.

This line of thinking has led to considerable interest in multi-scale descriptions [5,2,6,7]. However, merely computing descriptions at multiple scales does not solve the problem; if anything, it exacerbates it by increasing the volume of data. Some means must be found to organize or simplify the description, by relating one scale to another. Some work has been done in this area aimed at obtaining "edge pyramids" (e.g. [8]), but no clear-cut criteria for constructing them have been put forward. Marr [4] suggested that zero-crossings that coincide over several scales are "physically significant," but this idea was neither justified nor tested.

How, then, can descriptions at different scales be related to each other in an organized, natural, and compact way? Our solution, which we call *scale-space filtering*, begins by continuously varying the scale parameter, sweeping out a surface that we call the *scale-space image*. In this representation, it is possible to track extrema as they move continuously with scale changes, and to identify the singular points at which new extrema appear. The scale-space image is then collapsed into a tree, providing a concise but complete qualitative description of the signal over all scales of observation.[1]

2. The Scale-Space Image

Descriptions that depend on scale can be computed in many ways. As a primitive scale-parameterization, the gaussian convolution is attractive for a number of its properties, amounting to "well-behavedness": the gaussian is symmetric and strictly decreasing about the mean, and therefore the weighting assigned to signal values decreases smoothly with distance. The gaussian convolution behaves well near the limits of the scale parameter, σ, approaching the un-smoothed signal for small σ, and approaching the signal's mean for large σ. The gaussian is also readily differentiated and integrated.

The gaussian is not the only convolution kernel that meets these criteria. However, a more specific motivation for our choice is a property of the gaussian convolution's

[1] A complementary approach to the "natural" scale problem has been developed by Hoffman [9].

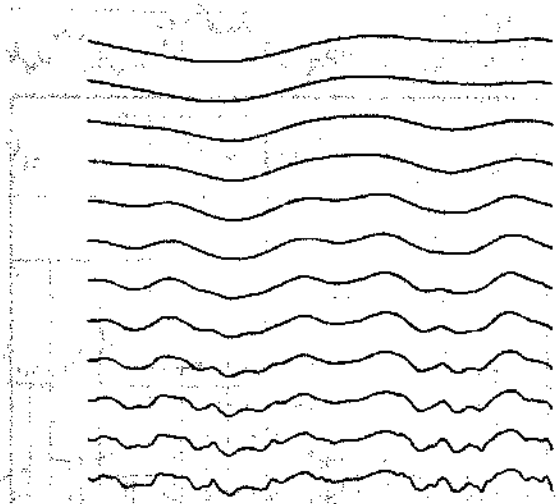


Figure 1. A sequence of gaussian smoothings of a waveform, with σ decreasing from top to bottom. Each graph is a constant-σ profile from the scale-space image.

zero-crossings (and those of its derivatives): as σ decreases, additional zeroes may appear, but existing ones cannot in general disappear; moreover, of convolution kernels satisfying "well behavedness" criteria (roughly those enumerated above,) the gaussian is the *only* one guaranteed to satisfy this condition [12]. The usefulness of this property will be explained in the following sections.

The gaussian convolution of a signal $f(x)$ depends both on x, the signal's independent variable, and on σ, the gaussian's standard deviation. The convolution is given by

$$F(x,\sigma) = f(x) * g(x,\sigma) = \int_{-\infty}^{\infty} f(u)\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-u)^2}{2\sigma^2}}\,\mathrm{d}\,u, \tag{1}$$

where "$*$" denotes convolution with respect to x. This function defines a surface on the (x,σ)-plane, where each profile of constant σ is a gaussian-smoothed version of $f(x)$, the amount of smoothing increasing with σ. We will call the (x,σ)-plane *scale space*, and the function, F, defined in (1), the *scale-space image* of f.[2] Fig. 1 graphs a sequence of gaussian smoothings with increasing σ. These are constant-σ profiles from the scale-space image.

At any value of σ, the extrema in the nth derivative of the smoothed signal are given by the zero-crossings in the $(n+1)$th derivative, computed using the relation

$$\frac{\partial^n F}{\partial x^n} = f * \frac{\partial^n g}{\partial x^n},$$

where the derivatives of the gaussian are readily obtained. Although the methods presented here apply to zeros in any derivative, we will restrict our attention to those in the second. These are extrema of slope, i.e. inflection points. In terms of the scale-space image, the inflections at *all* values of σ are the points that satisfy

$$F_{xx} = 0, F_{xxx} \neq 0, \tag{2}$$

[2]It is actually convenient to treat $\log\sigma$ as the scale parameter, uniform expansion or contraction of the signal in the x-direction will cause a translation of the scale-space image along the $\log\sigma$ axis.

Figure 2. Contours of $F_{xx} = 0$ in a scale-space image. The x-axis is horizontal; the coarsest scale is on top. To simulate the effect of a continuous scale-change on the qualitative description, hold a straight-edge (or better still, a slit) horizontally. The intersections of the edge with the zero-contours are the extremal points at some single value of σ. Moving the edge up or down increases or decreases σ.

using subscript notation to indicate partial differentiation.[3]

3. Coarse-to-fine Tracking

The contours of $F_{xx} = 0$ mark the appearance and motion of inflection points in the smoothed signal, and provide the raw material for a qualitative description over all scales, in terms of inflection points. Next, we will apply two simplifying assumptions to these contours: (1) the *identity* assumption, that extrema observed at different scales, but lying on a common zero-contour in scale space, arise from a single underlying event, and (2) the *localization* assumption, that the true location of an event giving rise to a zero-contour is the contour's x location as $\sigma \to 0$.

Referring to fig. 2, notice that the zero contours form arches, closed above, but open below. The restriction that zero-crossings may never disappear with with decreasing σ (see section 2) means that the contours may *never* be closed below. Note that at the apexes of the arches, $F_{xxx} = 0$, so by eq. (2), these points do not belong to the contour. Each arch consists of a pair of contours, crossing zero with opposite sign.

The *localization assumption* is motivated by the observation that linear smoothing has two effects: qualitative simplification—the removal of fine-scale features—and spatial distortion—dislocation, broadening and flattening of the features that survive. The latter undesirable effect may be overcome, by tracking coarse extrema to their fine-scale locations. Thus, a coarse scale may be used to *identify* extrema, and a fine scale, to *localize* them. Each zero-contour therefore reduces to an (x,σ) pair, specifying its fine-scale location on the x-axis, and the coarsest scale at which the contour appears.

A coarse-to-fine tracking description is compared to the

[3]Note that the second condition in (2) excludes zero-crossings that are parallel to the x-axis, because these are not zero-crossings in the convolved signal.

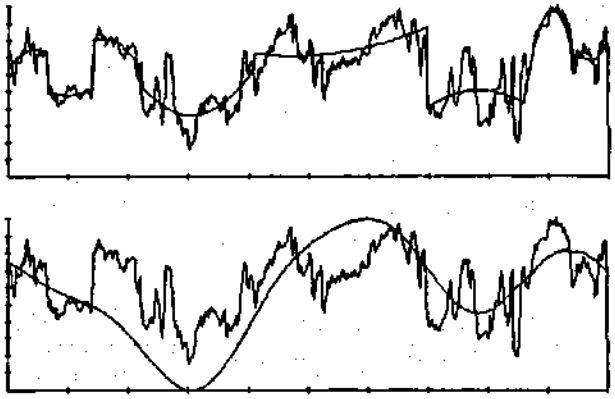


Figure 3. Above is shown a signal with a coarse-to-fine tracking approximation superimposed. The approximation was produced by independent parabolic fits between the localized inflections. Below is shown the corresponding (qualitatively isomorphic) gaussian smoothing.

corresponding linear smoothing in Fig. 3.[4]

4. The Interval Tree

While coarse-to-fine tracking solves the problem of localizing large-scale events, it does not solve the multi-scale integration problem, because the description still depends on the choice of the continuous global scale parameter, σ, just as simple linear filtering does. In this section, we reduce the scale-space image to a simple tree, concisely but completely describing the qualitative structure of the signal over all scales of observation.

This simplification rests on a basic property of the scale-space image: as σ is varied, extremal points in the smoothed signal appear and disappear at singular points (the tops of the arches in fig. 2.) Passing through such a point with decreasing σ, a pair of extrema of opposite sign appear in the smoothed signal. At these points, and only these points, the *un*distinguished interval (i.e. an interval bounded by extremal points but containing none) in which the singularity occurs splits into three subintervals. In general, each undistinguished interval, observed in scale space, is bounded on each side by the zero contours that define it, bounded above by the singular point at which it merges into an enclosing interval, and bounded below by the singular point at which it divides into sub-intervals.

Consequently, to each interval, I, corresponds a node in a (generally ternary-branching) tree, whose parent node denotes the larger interval from which I emerged, and whose offspring represent the smaller intervals into which I subdivides. Each interval also defines a rectangle in scale-space, denoting its location and extent on the signal (as defined by coarse-to-fine tracking) and its location and extent on the scale dimension. Collectively, these rectangles

[4]In this and all illustrations, approximations were drawn by fitting parabolic arcs independently to the signal data on each interval marked by the description. This procedure is crude, particularly because continuity is not enforced across inflections. Bear in mind that this procedure has been used only to display the qualitative description.

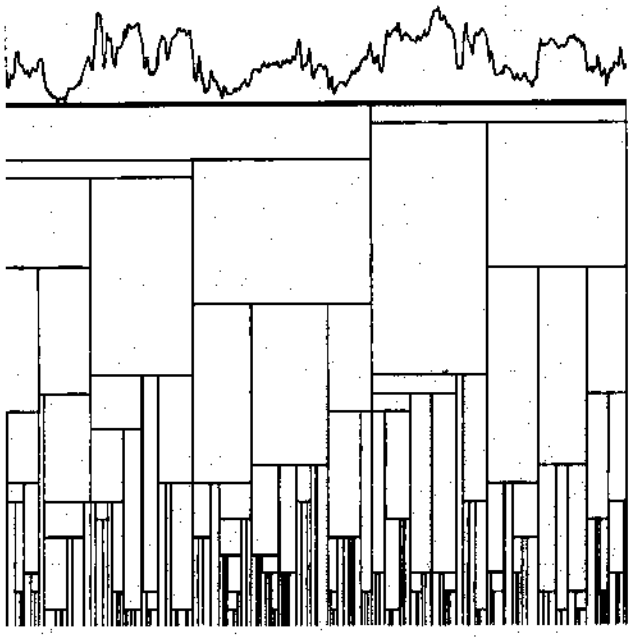


Figure 4. A signal with its interval tree, represented as a rectangular tesselation of scale-space. Each rectangle is a node, indicating an interval on the signal, and the scale interval over which the signal interval exists.

tesselate the (x, σ)-plane. See fig. 4 for an illustration of the tree.

This *interval tree* may be viewed in two ways: as describing the signal simultaneously at all scales, or as generating a family of single-scale descriptions, each defined by a subset of nodes in the tree that cover the x-axis. On the second interpretation, one may move through the family of descriptions in orderly, local, discrete steps, either by choosing to subdivide an interval into its offspring, or to merge a triple of intervals into their parent.[5]

We found that it is in general possible, by moving interactively through the tree and observing the resulting "sketch" of the signal, to closely match observers' spontaneously perceived descriptions. Thus the interval tree, though tightly constrained, seems flexible enough to capture human perceptual intuitions. Somewhat surprisingly, we found that the tree, rather than being too constraining, is not constrained enough. That is, the perceptually salient descriptions can in general be duplicated within the tree's constraints, but the tree also generates many descriptions that plainly have no perceptual counterpart. This observation led us to develop a *stability* criterion for further pruning or ordering the states of the tree, which is described in the next section.

5. Stability

Recall that to each interval in the tree corresponds a rectangle in scale space. The x boundaries locate the interval on the signal. The σ boundaries define the scale range over which the interval exists, its *stability* over scale changes. We have observed empirically a marked correspondence between the stability of an interval and its perceptual salience: those intervals that survive over a broad range of scales

[5]For previous uses of hierarchic signal descriptions see e.g. [10,11,2].

Figure 5. Several signals, with their maximum-stability descriptions. These are "top-level" descriptions, generated automatically and without thresholds. You should compare the descriptions to your own first-glance "top-level" percepts. (the noisy sine and square waves are synthetic signals.)

tend to leap out at the eye, while the most ephemeral are not perceived at all. To capture this relation, we have devised several versions of a stability criterion, one of which picks a "top-level" description by descending the tree until a local maximum in stability is found. Another iteratively removes nodes from the tree, splicing out nodes that are less stable than any of their parents and offspring. Both of these radically improve correspondence between the interval tree's descriptions and perceptual features (see fig. 5.)

6. Summary

Scale-space filtering is a method that describes signals qualitatively, in terms of extrema in the signal or its derivatives, in a manner that deals effectively with the problem of scale—precisely localizing large-scale events, and effectively managing the ambiguity of descriptions at multiple scales, without introducing arbitrary thresholds or free parameters. The one-dimensional signal is first expanded into a two-dimensional *scale-space image*, by convolution with gaussians over a continuum of sizes. This continuous surface is then collapsed into a discrete structure, using the connectivity of extremal points tracked through scale-space, and the singular points at which new extrema appear. The resulting tree representation is a a concise but complete qualitative description of the signal over all scales of observation. The tree is further constrained using a maximum-stability criterion to favor events that persist over large changes in scale.

We are currently developing applications of scale-space filtering to several signal matching and interpretation problems, and investigating its ability to explain perceptual grouping phenomena. The method is also being extended to apply to two-dimensional images: the scale-space image of a 2-D signal occupies a volume, containing zero-crossing surfaces.[6]

REFERENCES

[1] A. Rosenfeld and A. C. Kak Digital Picture Processing. *Academic Press, New York, New York,* 1976.

[2] D. H. Ballard and C. M. Brown Computer Vision. *Prentice Hall, Englewood Cliffs, New Jersey,* 1982.

[3] Pavlidis, T. *Structural Pattern Recognition.* Springer, 1977

[4] D. Marr Vision, *W. H. Freeman, San Fransisco,* 1982.

[5] Rosenfeld, A. and Thurston, M. "Edge and curve detection for visual scene analysis." *IEEE Transactions on computers, Vol C-20 pp. 562–569.* (May 1971).

[6] Marr, D and Poggio, T. "A computational theory of human stereo vision." *Proc. R. Soc. Lond.*, B. 204 (1979) pp. 301–328

[7] D. Marr and E. C. Hildreth Theory of Edge Detection. *M.I.T. Artificial Intelligence Memo Number 518, Cambridge, Massachusetts,* April 1979.

[8] Hong, T. H., Shneier, M. and Rosenfeld, A. Border Extraction using linked edge pyramids. *TR-1080, Computer Vision Laboratory, U. Maryland,* July 1981

[9] Hoffman, D. Representing Shapes For Visual Recognition *Ph.D. Thesis, MIT, forthcoming.*

[10] Erich, R. and Foith, J., "Representation of random waveforms by relational trees," *IEE Trans. Computers, Vol C-26, pp. 725-736,* (July 1976).

[11] Blumenthal A., Davis, L., and Rosenfeld, R., "Detecting natural 'plateaus' in one-dimensional patterns." *IEEE Transactions on computers,* (Feb. 1977)

[12] J. Babaud, R. Duda, and A. Witkin, *in preparation.*

[6] **Acknowledgments**—I thank my colleagues at FLAIR, particularly Richard Duda and Peter Hart, as well as J. Babaud of Schlumberger, ltd., for their help and encouragement.

Reprinted from *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume PAMI-8, Number 6, November 1986, pages 679-698.

A Computational Approach to Edge Detection

JOHN CANNY, MEMBER, IEEE

Abstract—**This paper describes a computational approach to edge detection. The success of the approach depends on the definition of a *comprehensive* set of goals for the computation of edge points. These goals must be precise enough to delimit the desired behavior of the detector while making minimal assumptions about the form of the solution. We define detection and localization criteria for a class of edges, and present mathematical forms for these criteria as functionals on the operator impulse response. A third criterion is then added to ensure that the detector has only one response to a single edge. We use the criteria in numerical optimization to derive detectors for several common image features, including step edges. On specializing the analysis to step edges, we find that there is a natural uncertainty principle between detection and localization performance, which are the two main goals. With this principle we derive a single operator shape which is optimal at any scale. The optimal detector has a simple approximate implementation in which edges are marked at maxima in gradient magnitude of a Gaussian-smoothed image. We extend this simple detector using operators of several widths to cope with different signal-to-noise ratios in the image. We present a general method, called feature synthesis, for the fine-to-coarse integration of information from operators at different scales. Finally we show that step edge detector performance improves considerably as the operator point spread function is extended along the edge. This detection scheme uses several elongated operators at each point, and the directional operator outputs are integrated with the gradient maximum detector.**

Index Terms—**Edge detection, feature extraction, image processing, machine vision, multiscale image analysis.**

I. Introduction

EDGE detectors of some kind, particularly step edge detectors, have been an essential part of many computer vision systems. The edge detection process serves to simplify the analysis of images by drastically reducing the amount of data to be processed, while at the same time preserving useful structural information about object boundaries. There is certainly a great deal of diversity in the applications of edge detection, but it is felt that many applications share a common set of requirements. These requirements yield an abstract edge detection problem, the solution of which can be applied in any of the original problem domains.

We should mention some specific applications here. The Binford–Horn line finder [14] used the output of an edge detector as input to a program which could isolate simple geometric solids. More recently the model-based vision system ACRONYM [3] used an edge detector as the front end to a sophisticated recognition program. Shape from motion [29], [13] can be used to infer the structure of three-dimensional objects from the motion of edge contours or edge points in the image plane. Several modern theories of stereopsis assume that images are preprocessed by an edge detector before matching is done [19], [20]. Beattie [1] describes an edge-based labeling scheme for low-level image understanding. Finally, some novel methods have been suggested for the extraction of three-dimensional information from image contours, namely shape from contour [27] and shape from texture [31].

In all of these examples there are common criteria relevant to edge detector performance. The first and most obvious is low error rate. It is important that edges that occur in the image should not be missed and that there be no spurious responses. In all the above cases, system performance will be hampered by edge detector errors. The second criterion is that the edge points be well localized. That is, the distance between the points marked by the detector and the "center" of the true edge should be minimized. This is particularly true of stereo and shape from motion, where small disparities are measured between left and right images or between images produced at slightly different times.

In this paper we will develop a mathematical form for these two criteria which can be used to design detectors for arbitrary edges. We will also discover that the first two criteria are not "tight" enough, and that it is necessary to add a third criterion to circumvent the possibility of multiple responses to a single edge. Using numerical optimization, we derive optimal operators for ridge and roof edges. We will then specialize the criteria for step edges and give a parametric closed form for the solution. In the process we will discover that there is an uncertainty principle relating detection and localization of noisy step edges, and that there is a direct tradeoff between the two. One consequence of this relationship is that there is a single unique "shape" of impulse response for an optimal step edge detector, and that the tradeoff between detection and localization can be varied by changing the spatial width of the detector. Several examples of the detector performance on real images will be given.

II. One-Dimensional Formulation

To facilitate the analysis we first consider one-dimensional edge profiles. That is, we will assume that two-

Manuscript received December 10, 1984; revised November 27, 1985. Recommended for acceptance by S. L. Tanimoto. This work was supported in part by the System Development Foundation, in part by the Office of Naval Research under Contract N00014-81-K-0494, and in part by the Advanced Research Projects Agency under Office of Naval Research Contracts N00014-80-C-0505 and N00014-82-K-0334.

The author is with the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139.

IEEE Log Number 8610412.

0-8186-9102-6/91 $1.00 © 1986 IEEE

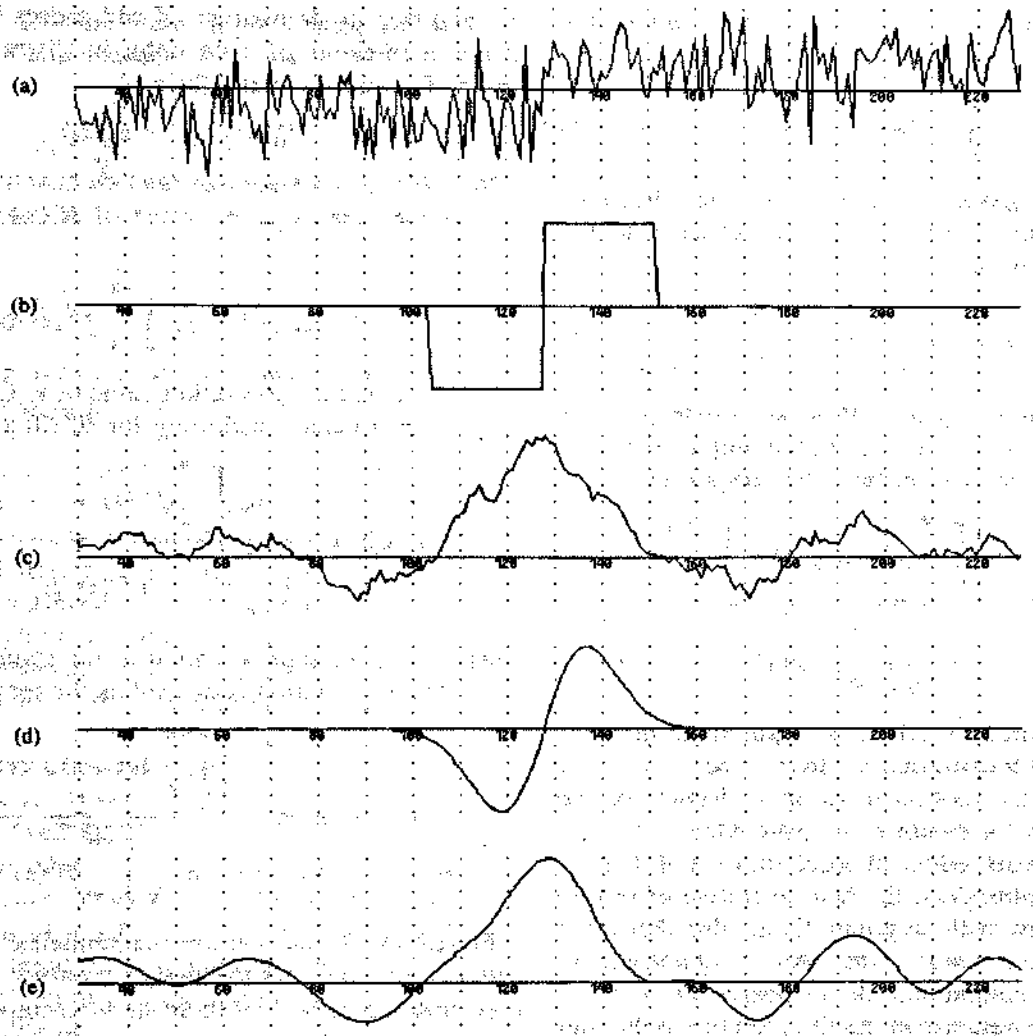


Fig. 1. (a) A noisy step edge. (b) Difference of boxes operator. (c) Difference of boxes operator applied to the edge. (d) First derivative of Gaussian operator. (e) First derivative of Gaussian applied to the edge.

dimensional edges locally have a constant cross-section in some direction. This would be true for example, of smooth edge contours or of ridges, but not true of corners. We will assume that the image consists of the edge and additive white Gaussian noise.

The detection problem is formulated as follows: We begin with an edge of known cross-section bathed in white Gaussian noise as in Fig. 1(a), which shows a step edge. We convolve this with a filter whose impulse response could be illustrated by either Fig. 1(b) or (d). The outputs of the convolutions are shown, respectively, in Fig. 1(c) and (e). We will mark the center of an edge at a local maximum in the output of the convolution. The design problem then becomes one of finding the filter which gives the best performance with respect to the criteria given below. For example, the filter in Fig. 1(d) performs much better than Fig. 1(b) on this example, because the response of the latter exhibits several local maxima in the region of the edge.

In summary, the three performance criteria are as follows:

1) Good detection. There should be a low probability of failing to mark real edge points, and low probability of falsely marking nonedge points. Since both these probabilities are monotonically decreasing functions of the output signal-to-noise ratio, this criterion corresponds to maximizing signal-to-noise ratio.

2) Good localization. The points marked as edge points by the operator should be as close as possible to the center of the true edge.

3) Only one response to a single edge. This is implicitly captured in the first criterion since when there are two responses to the same edge, one of them must be considered false. However, the mathematical form of the first criterion did not capture the multiple response requirement and it had to be made explicit.

A. Detection and Localization Criteria

A crucial step in our method is to capture the intuitive criteria given above in a mathematical form which is readily solvable. We deal first with signal-to-noise ratio and localization. Let the impulse response of the filter be $f(x)$, and denote the edge itself by $G(x)$. We will assume that the edge is centered at $x = 0$. Then the response of the

filter to this edge at its center H_G is given by a convolution integral:

$$H_G = \int_{-W}^{+W} G(-x)\, f(x)\, dx \tag{1}$$

assuming the filter has a finite impulse response bounded by $[-W, W]$. The root-mean-squared response to the noise $n(x)$ only, will be

$$H_n = n_0 \left[\int_{-W}^{+W} f^2(x)\, dx \right]^{1/2} \tag{2}$$

where n_0^2 is the mean-squared noise amplitude per unit length. We define our first criterion, the output signal-to-noise ratio, as the quotient of these two responses.

$$\text{SNR} = \frac{\left| \int_{-W}^{+W} G(-x)\, f(x)\, dx \right|}{n_0 \sqrt{\int_{-W}^{+W} f^2(x)\, dx}} \tag{3}$$

For the localization criterion, we want some measure which increases as localization improves, and we will use the reciprocal of the root-mean-squared distance of the marked edge from the center of the true edge. Since we have decided to mark edges at local maxima in the response of the operator $f(x)$, the first derivative of the response will be zero at these points. Note also that since edges are centered at $x = 0$, *in the absence of noise* there should be a local maximum in the response at $x = 0$.

Let $H_n(x)$ be the response of the filter to noise only, and $H_G(x)$ be its response to the edge, and suppose there is a local maximum in the total response at the point $x = x_0$. Then we have

$$H_n'(x_0) + H_G'(x_0) = 0. \tag{4}$$

The Taylor expansion of $H_G'(x_0)$ about the origin gives

$$H_G'(x_0) = H_G'(0) + H_G''(0)x_0 + O(x_0^2). \tag{5}$$

By assumption $H_G'(0) = 0$, i.e., the response of the filter in the absence of noise has a local maximum at the origin, so the first term in the expansion can be ignored. The displacement x_0 of the actual maximum is assumed to be small so we will ignore quadratic and higher terms. In fact by a simple argument we can show that if the edge $G(x)$ is either symmetric or antisymmetric, all even terms in x_0 vanish. Suppose $G(x)$ is antisymmetric, and express $f(x)$ as a sum of a symmetric component and an antisymmetric component. The convolution of the symmetric component with $G(x)$ contributes nothing to the numerator of the SNR, but it does contribute to the noise component in the denominator. Therefore, if $f(x)$ has any symmetric component, its SNR will be worse than a purely antisymmetric filter. A dual argument holds for symmetric edges, so that if the edge $G(x)$ is symmetric or antisymmetric, the filter $f(x)$ will follow suit. The net result of this is that the response $H_G(x)$ is always symmetric, and that its derivatives of odd orders [which appear in the coefficients of even order in (5)] are zero at the origin. Equations (4) and (5) give

$$H_G''(0)x_0 \approx -H_n'(x_0). \tag{6}$$

Now $H_n'(x_0)$ is a Gaussian random quantity whose variance is the mean-squared value of $H_n'(x_0)$, and is given by

$$E[H_n'(x_0)^2] = n_0^2 \int_{-W}^{+W} f'^2(x)\, dx \tag{7}$$

where $E[y]$ is the expectation value of y. Combining this result with (6) and substituting for $H_G''(0)$ gives

$$E[x_0^2] \approx \frac{n_0^2 \int_{-W}^{+W} f'^2(x)\, dx}{\left[\int_{-W}^{+W} G'(-x)\, f'(x)\, dx \right]^2} = \delta x_0^2 \tag{8}$$

where δx_0 is an approximation to the standard deviation of x_0. The localization is defined as the reciprocal of δx_0.

$$\text{Localization} = \frac{\left| \int_{-W}^{+W} G'(-x)\, f'(x)\, dx \right|}{n_0 \sqrt{\int_{-W}^{+W} f'^2(x)\, dx}}. \tag{9}$$

Equations (3) and (9) are mathematical forms for the first two criteria, and the design problem reduces to the maximization of both of these simultaneously. In order to do this, we maximize the product of (3) and (9). We could conceivably have combined (3) and (9) using any function that is monotonic in two arguments, but the use of the product simplifies the analysis for step edges, as should become clear in Section III. For the present we will make use of the product of the criteria for arbitrary edges, i.e., we seek to maximize

$$\frac{\left| \int_{-W}^{+W} G(-x)\, f(x)\, dx \right| \left| \int_{-W}^{+W} G'(-x)\, f'(x)\, dx \right|}{n_0 \sqrt{\int_{-W}^{+W} f^2(x)\, dx}\; n_0 \sqrt{\int_{-W}^{+W} f'^2(x)\, dx}}. \tag{10}$$

There may be some additional constraints on the solution, such as the multiple response constraint (12) described next.

B. Eliminating Multiple Responses

In our specification of the edge detection problem, we decided that edges would be marked at local maxima in the response of a linear filter applied to the image. The detection criterion given in the last section measures the effectiveness of the filter in discriminating between signal and noise at the center of an edge. It does not take into account the behavior of the filter *nearby* the edge center. The first two criteria can be trivially maximized as fol-

lows. From the Schwarz inequality for integrals we can show that SNR (3) is bounded above by

$$n_0^{-1}\sqrt{\int_{-W}^{+W} G^2(x)\,dx}$$

and localization (9) by

$$n_0^{-1}\sqrt{\int_{-W}^{+W} G'^2(x)\,dx}.$$

Both bounds are attained, and the product of SNR and localization is maximized when $f(x) = G(-x)$ in $[-W, W]$.

Thus, according to the first two criteria, the optimal detector for step edges is a truncated step, or difference of boxes operator. The difference of boxes was used by Rosenfeld and Thurston [25], and in conjunction with lateral inhibition by Herskovits and Binford [11]. However it has a very high bandwidth and tends to exhibit many maxima in its response to noisy step edges, which is a serious problem when the imaging system adds noise or when the image itself contains textured regions. These extra edges should be considered erroneous according to the first of our criteria. However, the analytic form of this criterion was derived from the response at a single point (the center of the edge) and did not consider the interaction of the responses at several nearby points. If we examine the output of a difference of boxes edge detector we find that the response to a noisy step is a roughly triangular peak with numerous sharp maxima in the vicinity of the edge (see Fig. 1).

These maxima are so close together that it is not possible to select one as the response to the step while identifying the others as noise. We need to add to our criteria the requirement that the function f will not have "too many" responses to a single step edge in the vicinity of the step. We need to limit the number of peaks in the response so that there will be a low probability of declaring more than one edge. Ideally, we would like to make the distance between peaks in the noise response approximate the width of the response of the operator to a single step. This width will be some fraction of the operator width W.

In order to express this as a functional constraint on f, we need to obtain an expression for the distance between adjacent noise peaks. We first note that the mean distance between adjacent maxima in the output is twice the distance between adjacent zero-crossings in the derivative of the operator output. Then we make use of a result due to Rice [24] that the average distance between zero-crossings of the response of a function g to Gaussian noise is

$$x_{\text{ave}} = \pi\left(\frac{-R(0)}{R''(0)}\right)^{1/2} \tag{11}$$

where $R(\tau)$ is the autocorrelation function of g. In our case we are looking for the mean zero-crossing spacing for the function f'. Now since

$$R(0) = \int_{-\infty}^{+\infty} g^2(x)\,dx \quad \text{and} \quad R''(0) = -\int_{-\infty}^{+\infty} g'^2(x)\,dx$$

the mean distance between zero-crossings of f' will be

$$x_{zc}(f) = \pi\left(\frac{\int_{-\infty}^{+\infty} f'^2(x)\,dx}{\int_{-\infty}^{+\infty} f''^2(x)\,dx}\right)^{1/2} \tag{12}$$

The distance between adjacent maxima in the noise response of f, denoted $x_{\max}$, will be twice x_{zc}. We set this distance to be some fraction k of the operator width.

$$x_{\max}(f) = 2x_{zc}(f) = kW. \tag{13}$$

This is a natural form for the constraint because the response of the filter will be concentrated in a region of width $2W$, and the expected number of noise maxima in this region is N_n where

$$N_n = \frac{2W}{x_{\max}} = \frac{2}{k}. \tag{14}$$

Fixing k fixes the number of noise maxima that could lead to a false response.

We remark here that the intermaximum spacing (12) scales with the operator width. That is, we first define an operator f_w which is the result of stretching f by a factor of w, $f_w(x) = f(x/w)$. Then after substituting into (12) we find that the intermaximum spacing for f_w is $x_{zc}(f_w) = wx_{zc}(f)$. Therefore, if a function f satisfies the multiple response constraint (13) for fixed k, then the function f_w will also satisfy it, assuming W scales with w. For any fixed k, the multiple response criterion is invariant with respect to spatial scaling of f.

III. Finding Optimal Detectors by Numerical Optimization

In general it will be difficult (or impossible) to find a closed form for the function f which maximizes (10) subject to the multiple response constraint. Even when G has a particularly simple form (e.g., it is a step edge), the form of f may be complicated. However, if we are given a candidate function f, evaluation of (10) and (12) is straightforward. In particular, if the function f is represented by a discrete time sequence, evaluation of (10) requires only the computation of four inner products between sequences. This suggests that numerical optimization can be done directly on the sampled operator impulse response.

The output will not be an analytic form for the operator, but an implementation of a detector for the edge of interest will require discrete point-spread functions anyway. It is also possible to include additional constraints by using a *penalty method* [15]. In this scheme, the constrained optimization is reduced to one, or possibly several, unconstrained optimizations. For each constraint we define a penalty function which has a nonzero value when one

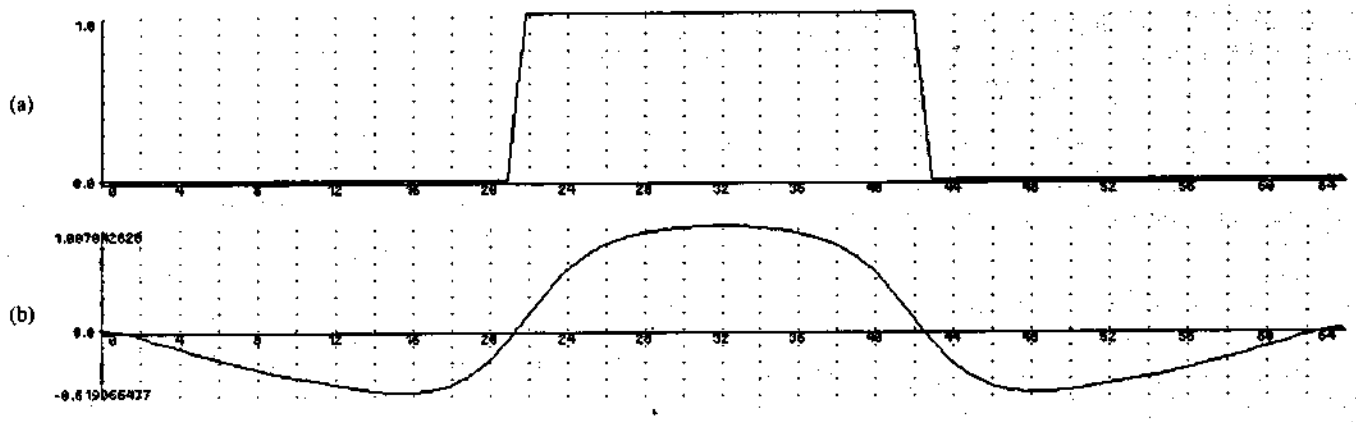


Fig. 2. A ridge profile and the optimal operator for it.

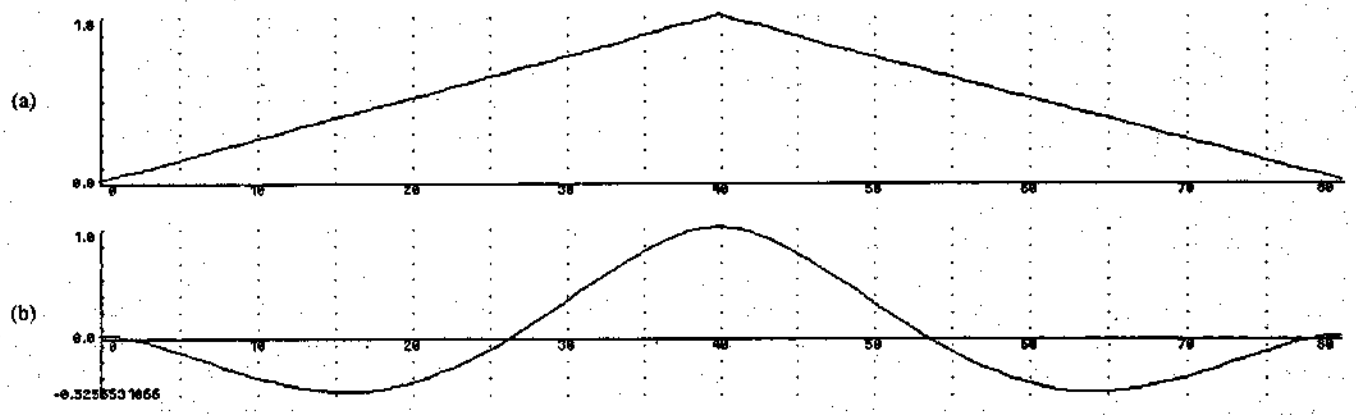


Fig. 3. A roof profile and an optimal operator for roofs.

of the constraints is violated. We then find the f which maximizes

$$\text{SNR}(f) * \text{Localization}\ (f) - \sum \mu_i P_i(f) \qquad (15)$$

where P_i is a function which has a positive value only when a constraint is violated. The larger the value of μ_i the more nearly the constraints will be satisfied, but at the same time the greater the likelihood that the problem will be ill-conditioned. A sequence of values of μ_i may need to be used, with the final form of f from each optimization used as the starting form for the next. The μ_i are increased at each iteration so that the value of $P_i(f)$ will be reduced, until the constraints are "almost" satisfied.

An example of the method applied to the problem of detecting "ridge" profiles is shown in Fig. 2. For a ridge, the function G is defined to be a flat plateau of width w, with step transitions to zero at the ends. The auxiliary constraints are

- The multiple response constraint. This constraint is taken directly from (12), and does not depend on the form of the edge.
- The operator should have zero dc component. That is it should have zero output to constant input.

Since the width of the operator is dependent on the width of the ridge, there is a suggestion that several widths of operators should be used. This has not been done in the present implementation however. A wide ridge can be considered to be two closely spaced edges, and the implementation already includes detectors for these. The only reason for using a ridge detector is that there are ridges in images that are too small to be dealt with effectively by the narrowest edge operator. These occur frequently because there are many edges (e.g., scratches and cracks or printed matter) which lie at or beyond the resolution of the camera and result in contours only one or two pixels wide.

A similar procedure was used to find an optimal operator for roof edges. These edges typically occur at the concave junctions of two planar faces in polyhedral objects. The results are shown in Fig. 3. Again there are two subsidiary constraints, one for multiple responses and one for zero response to constant input.

A roof edge detector has not been incorporated into the implementation of the edge detector because it was found that ideal roof edges were relatively rare. In any case the ridge detector is an approximation to the ideal roof detector, and is adequate to cope with roofs. The situation may be different in the case of an edge detector designed explicitly to deal with images of polyhedra, like the Binford–Horn line-finder [14].

The method just described has been used to find optimal operators for both ridge and roof profiles and in addition it successfully finds the optimal step edge operator derived in Section IV. It should be possible to use it to find operators for arbitrary one-dimensional edges, and it should be possible to apply the method in two dimensions to find optimal detectors for various types of corner.

IV. A Detector for Step Edges

We now specialize the results of the last section to the case where the input $G(x)$ is step edge. Specifically we set $G(x) = Au_{-1}(x)$ where $u_n(x)$ is the nth derivative of a delta function, and A is the amplitude of the step. That is,

$$u_{-1}(x) = \begin{cases} 0, & \text{for } x < 0; \\ 1, & \text{for } x \geq 0; \end{cases} \tag{16}$$

and substituting for $G(x)$ in (3) and (9) gives

$$\text{SNR} = \frac{A\left|\int_{-w}^{0} f(x)\,dx\right|}{n_0\sqrt{\int_{-w}^{+w} f^2(x)\,dx}} \tag{17}$$

$$\text{Localization} = \frac{A|f'(0)|}{n_0\sqrt{\int_{-w}^{+w} f'^2(x)\,dx}}. \tag{18}$$

Both of these criteria improve directly with the ratio A/n_0 which might be termed the signal-to-noise ratio of the image. We now remove this dependence on the image and define two performance measures Σ and Λ which depend on the filter only:

$$\text{SNR} = \frac{A}{n_0}\Sigma(f) \qquad \Sigma(f) = \frac{\left|\int_{-w}^{0} f(x)\,dx\right|}{\sqrt{\int_{-w}^{+w} f^2(x)\,dx}} \tag{19}$$

$$\text{Localization} = \frac{A}{n_0}\Lambda(f') \qquad \Lambda(f') = \frac{|f'(0)|}{\sqrt{\int_{-w}^{+w} f'^2(x)\,dx}}. \tag{20}$$

Suppose now that we form a spatially scaled filter f_w from f, where $f_w(x) = f(x/w)$. Recall from the end of Section II that the multiple response criterion is unaffected by spatial scaling. When we substitute f_w into (19) and (20) we obtain for the performance of the scaled filter:

$$\Sigma(f_w) = \sqrt{w}\Sigma(f) \quad \text{and} \quad \Lambda(f'_w) = \frac{1}{\sqrt{w}}\Lambda(f'). \tag{21}$$

The first of these equations is quite intuitive, and implies that a filter with a broad impulse response will have better signal-to-noise ratio than a narrow filter when applied to a step edge. The second is less obvious, and it implies that a narrow filter will give better localization than a broad one. What is surprising is that the changes are inversely related, that is, both criteria either increase or decrease by $\sqrt{w}$. There is an uncertainty principle relating the detection and localization performance of the step edge detector. Through spatial scaling of f we can trade off detection performance against localization, but we cannot improve both simultaneously. This suggests that a natural choice for the composite criterion would be the product of (19) and (20), since this product would be invariant under changes in scale.

$$\Sigma(f)\,\Lambda(f') = \frac{\left|\int_{-w}^{0} f(x)\,dx\right|}{\sqrt{\int_{-w}^{+w} f^2(x)\,dx}} \frac{|f'(0)|}{\sqrt{\int_{-w}^{+w} f'^2(x)\,dx}}. \tag{22}$$

The solutions to the maximization of this expression will be a *class of functions* all related by spatial scaling. In fact this result is independent of the method of combination of the criteria. To see this we assume that there is a function f which gives the best localization Λ for a particular Σ. That is, we find f such that

$$\Sigma(f) = c_1 \quad \text{and} \quad \Lambda(f') \text{ is maximized.} \tag{23}$$

Now suppose we seek a second function f_w which gives the best possible localization while its signal-to-noise ratio is fixed to a different value, i.e.,

$$\Sigma(f_w) = c_2 \quad \text{while} \quad \Lambda(f'_w) \text{ is maximized.} \tag{24}$$

If we now define $f_1(x)$ in terms of $f_w(x)$ as $f_1(x) = f_w(xw)$ where

$$w = c_2^2/c_1^2$$

then the constraint on f_w in (24) translates to a constraint on f_1 which is identical to (23), and (24) can be rewritten as

$$\Sigma(f_1) = c_1 \quad \text{and} \quad \frac{1}{\sqrt{w}}\Lambda(f'_1) \text{ is maximized} \tag{25}$$

which has the solution $f_1 = f$. So if we find a single such function f, we can obtain maximal localization for any fixed signal-to-noise ratio by scaling f. The design problem for step edge detection has a *single unique* (up to spatial scaling) solution regardless of the absolute values of signal to noise ratio or localization.

The optimal filter is implicitly defined by (22), but we must transform the problem slightly before we can apply the calculus of variations. Specifically, we transform the maximization of (22) into a constrained minimization that involves only integral functionals. All but one of the integrals in (22) are set to undetermined constant values. We then find the extreme value of the remaining integral (since it will correspond to an extreme in the total expression) as a function of the undetermined constants. The values of the constants are then chosen so as to maximize the original expression, which is now a function only of these constants. Given the constants, we can uniquely specify the function $f(x)$ which gives a maximum of the composite criterion.

A second modification involves the limits of the integrals. The two integrals in the denominator of (22) have

limits at $+W$ and $-W$, while the integral in the numerator has one limit at 0 and the other at $-W$. Since the function f should be antisymmetric, we can use the latter limits for all integrals. The denominator integrals will have half the value over this subrange that they would have over the full range. Also, this enables the value of $f'(0)$ to be set as a boundary condition, rather than expressed as an integral of f''. If the integral to be minimized shares the same limits as the constraint integrals, it is possible to exploit the *isoperimetric constraint* condition (see [6, p. 216]). When this condition is fulfilled, the constrained optimization can be reduced to an unconstrained optimization using Lagrange multipliers for the constraint functionals. The problem of finding the maximum of (22) reduces to the minimization of the integral in the denominator of the SNR term, subject to the constraint that the other integrals remain constant. By the principle of reciprocity, we could have chosen to extremize any of the integrals while keeping the others constant, and the solution should be the same.

We seek some function f chosen from a space of *admissible* functions that minimizes the integral

$$\int_{-W}^{0} f^2(x)\,dx \tag{26}$$

subject to

$$\int_{-W}^{0} f(x)\,dx = c_1 \qquad \int_{-W}^{0} f'^2(x)\,dx = c_2$$

$$\int_{-W}^{0} f''^2(x)\,dx = c_3 \qquad f'(0) = c_4. \tag{27}$$

The space of admissible functions in this case will be the space of all continuous functions that satisfy certain boundary conditions, namely that $f(0) = 0$ and $f(-W) = 0$. These boundary conditions are necessary to ensure that the integrals evaluated over finite limits accurately represent the infinite convolution integrals. That is, if the nth derivative of f appears in some integral, the function must be continuous in its $(n-1)$st derivative over the range $(-\infty, +\infty)$. This implies that the values of f and its first $(n-1)$ derivatives must be zero at the limits of integration, since they are zero outside this range.

The functional to be minimized is of the form $\int_a^b F(x, f, f', f'')$ and we have a series of constraints that can be written in the form $\int_a^b G_i(x, f, f', f'') = c_i$. Since the constraints are isoperimetric, i.e., they share the same limits of integration as the integral being minimized, we can form a composite functional using Lagrange multipliers [6]. The functional is a linear combination of the functionals that appear in the expression to be minimized and in the constraints. Finding a solution to the unconstrained maximization of $\Psi(x, f, f', f'')$ is equivalent to finding the solution to the constrained problem. The composite functional is

$$\Psi(x, f, f', f'') = F(x, f, f', f'') + \lambda_1 G_1(x, f, f', f'') + \lambda_2 G_2(x, f, f', f'') + \cdots$$

Substituting,

$$\Psi(x, f, f', f'') = f^2 + \lambda_1 f'^2 + \lambda_2 f''^2 + \lambda_3 f. \tag{28}$$

It may be seen from the form of this equation that the choice of which integral is extremized and which are constraints is arbitrary, the solution will be the same. This is an example of the *reciprocity* that was mentioned earlier. The choice of an integral from the denominator is simply convenient since the standard form of variational problem is a minimization problem. The Euler equation that corresponds to the functional Ψ is

$$\Psi_f - \frac{d}{dx}\Psi_{f'} + \frac{d^2}{dx^2}\Psi_{f''} = 0 \tag{29}$$

where Ψ_f denotes the partial derivative of Ψ with respect to f, etc. We substitute for Ψ from (28) in the Euler equation giving:

$$2f(x) - 2\lambda_1 f''(x) + 2\lambda_2 f''''(x) + \lambda_3 = 0. \tag{30}$$

The solution of this differential equation is the sum of a constant and a set of four exponentials of the form $e^{\gamma x}$ where γ derives from the solution of the corresponding homogeneous differential equation. Now γ must satisfy

$$2 - 2\lambda_1\gamma^2 + 2\lambda_2\gamma^4 = 0$$

so

$$\gamma^2 = \frac{\lambda_1}{2\lambda_2} \pm \frac{\sqrt{\lambda_1^2 - 4\lambda_2}}{2\lambda_2}. \tag{31}$$

This equation may have roots that are purely imaginary, purely real, or complex depending on the values of λ_1 and λ_2. From the composite functional Ψ we can infer that λ_2 is positive (since the integral of f''^2 is to be minimized) but it is not clear what the sign or magnitude of λ_1 should be. The Euler equation supplies a necessary condition for the existence of a minimum, but it is not a sufficient condition. By formulating such a condition we can resolve the ambiguity in the value of λ_1. To do this we must consider the second variation of the functional. Let

$$J[f] = \int_{x_0}^{x_1} \Psi(x, f, f', f'')\,dx.$$

Then by Taylor's theorem (see also [6, p. 214]),

$$J[f + \epsilon g] = J[f] + \epsilon J_1[f, g] + \tfrac{1}{2}\epsilon^2 J_2[f + \rho g, g]$$

where ρ is some number between 0 and ϵ, and g is chosen from the space of admissible functions, and where

$$J_1[f, g] = \int_{x_0}^{x_1} \Psi_f g + \Psi_{f'} g' + \Psi_{f''} g''\,dx$$

$$J_2[f, g] = \int_{x_0}^{x_1} \Psi_{ff} g^2 + \Psi_{f'f'} g'^2 + \Psi_{f''f''} g''^2 + 2\Psi_{ff'} g g' + 2\Psi_{f'f''} g' g'' + 2\Psi_{ff''} g g''\,dx. \tag{32}$$

Note that J_1 is nothing more than the integral of g times the Euler equation for f (transformed using integration by parts) and will be zero if f satisfies the Euler equation. We can now define the second variation $\delta^2 J$ as

$$\delta^2 J = \frac{\epsilon^2}{2} J_2[f, g].$$

The necessary condition for a minimum is $\delta^2 J \geq 0$. We compute the second partial derivatives of Ψ from (28) and we get

$$J_1[f + g] = \int_{x_0}^{x_1} 2g^2 + 2\lambda_1 g'^2 + 2\lambda_2 g''^2 \, dx \geq 0. \tag{33}$$

Using the fact that g is an admissible function and therefore vanishes at the integration limits, we transform the above using integration by parts to

$$2\int_{x_0}^{x_1} g^2 - \lambda_1 g g'' + \lambda_2 g''^2 \, dx \geq 0$$

which can be written as

$$2\int_{x_0}^{x_1} \left(g^2 - \frac{\lambda_1}{2} g''\right)^2 + \left(\lambda_2 - \frac{\lambda_1^2}{4}\right) g''^2 \, dx \geq 0.$$

The integral is guaranteed to be positive if the expression being integrated is positive for all x, so if

$$4\lambda_2 > \lambda_1^2$$

then the integral will be positive for all x and for arbitrary g, and the extremum will certainly be a minimum. If we refer back to (31) we find that this condition is precisely that which gives complex roots for γ, so we have both guaranteed the existence of a minimum and resolved a possible ambiguity in the form of the solution. We can now proceed with the derivation and assume four complex roots of the form $\gamma = \pm\alpha \pm i\omega$ with α, ω real. Now $\gamma^2 = \alpha^2 - \omega^2 \pm 2i\alpha\omega$ and equating real and imaginary parts with (31) we obtain

$$\alpha^2 - \omega^2 = \frac{\lambda_1}{2\lambda_2} \quad \text{and} \quad 4\alpha^2\omega^2 = \frac{4\lambda_2 - \lambda_1^2}{4\lambda_2^2} \tag{34}$$

The general solution in the range $[-W, 0]$ may now be written

$$f(x) = a_1 e^{\alpha x} \sin \omega x + a_2 e^{\alpha x} \cos \omega x + a_3 e^{-\alpha x} \sin \omega x + a_4 e^{-\alpha x} \cos \omega x + c. \tag{35}$$

This function is subject to the boundary conditions

$$f(0) = 0 \quad f(-W) = 0 \quad f'(0) = s \quad f'(-W) = 0$$

where s is an unknown constant equal to the slope of the function f at the origin. Since $f(x)$ is asymmetric, we can extend the above definition to the range $[-W, W]$ using $f(-x) = -f(x)$. The four boundary conditions enable us to solve for the quantities a_1 through a_4 in terms of the unknown constants α, ω, c, and s. The boundary conditions may be rewritten

$$\begin{aligned}
a_2 + a_4 + c &= 0 \\
a_1 e^{\alpha} \sin \omega + a_2 e^{\alpha} \cos \omega + a_3 e^{-\alpha} \sin \omega & \\
+ a_4 e^{-\alpha} \cos \omega + c &= 0 \\
a_1 \omega + a_2 \alpha + a_3 \omega - a_4 \alpha &= s \\
a_1 e^{\alpha}(\alpha \sin \omega + \omega \cos \omega) + a_2 e^{\alpha}(\alpha \cos \omega & \\
- \omega \sin \omega) + a_3 e^{-\alpha}(-\alpha \sin \omega + \omega \cos \omega) & \\
+ a_4 e^{-\alpha}(-\alpha \cos \omega - \omega \sin \omega) &= 0.
\end{aligned} \tag{36}$$

These equations are linear in the four unknowns a_1, a_2, a_3, a_4 and when solved they yield

$$\begin{aligned}
a_1 &= c(\alpha(\beta - \alpha) \sin 2\omega - \alpha\omega \cos 2\omega + (-2\omega^2 \sinh \alpha \\
&\quad + 2\alpha^2 e^{-\alpha}) \sin \omega + 2\alpha\omega \sinh \alpha \cos \omega \\
&\quad + \omega e^{-2\alpha}(\alpha + \beta) - \beta\omega)/4(\omega^2 \sinh^2 \alpha - \alpha^2 \sin^2 \omega) \\
a_2 &= c(\alpha(\beta - \alpha) \cos 2\omega + \alpha\omega \sin 2\omega - 2\alpha\omega \cosh \alpha \\
&\quad \cdot \sin \omega - 2\omega^2 \sinh \alpha \cos \omega + 2\omega^2 e^{-\alpha} \sinh \alpha \\
&\quad + \alpha(\alpha - \beta))/4(\omega^2 \sinh^2 \alpha - \alpha^2 \sin^2 \omega) \\
a_3 &= c(-\alpha(\beta + \alpha) \sin 2\omega + \alpha\omega \cos 2\omega + (2\omega^2 \sinh \alpha \\
&\quad + 2\alpha^2 e^{\alpha}) \sin \omega + 2\alpha\omega \sinh \alpha \cos \omega \\
&\quad + \omega e^{2\alpha}(\beta - \alpha) - \beta\omega)/4(\omega^2 \sinh^2 \alpha - \alpha^2 \sin^2 \omega) \\
a_4 &= c(-\alpha(\beta + \alpha) \cos 2\omega - \alpha\omega \sin 2\omega + 2\alpha\omega \cosh \alpha \\
&\quad \cdot \sin \omega + 2\omega^2 \sinh \alpha \cos \omega - 2\omega^2 e^{\alpha} \sinh \alpha \\
&\quad + \alpha(\alpha - \beta))/4(\omega^2 \sinh^2 \alpha - \alpha^2 \sin^2 \omega)
\end{aligned} \tag{37}$$

where β is the slope s at the origin divided by the constant c. On inspection of these expressions we can see that a_3 can be obtained from a_1 by replacing α by $-\alpha$, and similarly for a_4 from a_2.

The function f is now parametrized in terms of the constants α, ω, β, and c. We have still to find the values of these parameters which maximize the quotient of integrals that forms our composite criterion. To do this we first express each of the integrals in terms of the constants. Since these integrals are very long and uninteresting, they are not given here but may be found in [4]. We have reduced the problem of optimizing over an infinite-dimensional space of functions to a nonlinear optimization in three variables α, ω, and β (not surprisingly, the combined criterion does not depend on c). Unfortunately the resulting criterion, which must still satisfy the multiple response constraint, is probably too complex to be solved analytically, and numerical methods must be used to provide the final solution.

The shape of f will depend on the multiple response constraint, i.e., it will depend on how far apart we force the adjacent responses. Fig. 5 shows the operators that result from particular choices of this distance. Recall that there was no single best function for arbitrary ω, but a class of functions which were obtained by scaling a pro-

totype function by ω. We will want to force the responses further apart as the signal-to-noise ratio in the image is lowered, but it is not clear what the value of signal-to-noise ratio will be for a single operator. In the context in which this operator is used, several operator widths are available, and a decision procedure is applied to select the smallest operator that has an output signal-to-noise ratio above a fixed threshold. With this arrangement the operators will spend much of the time operating close to their output Σ thresholds. We try to choose a spacing for which the probability of a multiple response error is comparable to the probability of an error due to thresholding.

A rough estimate for the probability of a spurious maximum in the neighborhood of the true maximum can be formed as follows. If we look at the response of f to an ideal step we find that its second derivative has magnitude $|Af'(0)|$ at $x = 0$. There will be only one maximum near the center of the edge if $|Af'(0)|$ is greater than the second derivative of the response to noise only. This latter quantity, denoted s_n, is a Gaussian random variable with standard deviation

$$n_0\sigma_s = n_0\left(\int_{-W}^{+W} f''^2(x)\,dx\right)^{1/2}.$$

The probability p_m that the noise slope s_n exceeds $Af'(0)$ is given in terms of the normal distribution function Φ

$$p_m = 1 - \Phi\left(\frac{A|f'(0)|}{n_0\sigma_s}\right). \tag{38}$$

We can choose a value for this probability as an acceptable error rate and this will determine the ratio of $f'(0)$ to σ_s. We can relate the probability of a multiple response p_m to the probability of falsely marking an edge p_f which is

$$p_f = 1 - \Phi\left(\frac{A}{n_0}\Sigma\right) \tag{39}$$

by setting $p_m = p_f$. This is a natural choice since it makes a detection error or a multiple response error equally likely. Then from (38) and (39) we have

$$\frac{|f'(0)|}{\sigma_s} = \Sigma. \tag{40}$$

In practice it was impossible to find filters which satisfied this constraint, so instead we search for a filter satisfying

$$\frac{|f'(0)|}{\sigma_s} = r\Sigma \tag{41}$$

where r is as close as possible to 1. The performance indexes and parameter values for several filters are given in Fig. 4. The a_i coefficients for all these filters can be found from (37), by fixing c to, say, $c = 1$. Unfortunately, the largest value of r that could be obtained using the constrained numerical optimization was about 0.576 for filter number 6 in the table. In our implementation, we have approximated this filter using the first derivative of a Gaussian as described in the next section.

Filter Parameters						
n	x_{max}	$\Sigma\Delta$	r	α	ω	β
1	0.15	4.21	0.215	24.59550	0.12250	63.97566
2	0.3	2.87	0.313	12.47120	0.38284	31.26860
3	0.5	2.13	0.417	7.85869	2.62856	18.28800
4	0.8	1.57	0.515	5.06500	2.56770	11.06100
5	1.0	1.33	0.561	3.45580	0.07161	4.80684
6	1.2	1.12	0.576	2.05220	1.56939	2.91540
7	1.4	0.75	0.484	0.00297	3.50350	7.47700

Fig. 4. Filter parameters and performance measures for the filters illustrated in Fig. 5.

The first derivative of Gaussian operator, or even filter 6 itself, should not be taken as the final word in edge detection filters, even with respect to the criteria we have used. If we are willing to tolerate a slight reduction in multiple response performance r, we can obtain significant improvements in the other two criteria. For example, filters 4 and 5 both have significantly better $\Sigma\Delta$ product than filter 6, and only slightly lower r. From Fig. 5 we can see that these filters have steeper slope at the origin, suggesting that the performance gain is mostly in localization, although this has not been verified experimentally. A thorough empirical comparison of these other operators remains to be done, and the theory in this case is unclear on how best to make the tradeoff.

V. An Efficient Approximation

The operator derived in the last section as filter number 6, and illustrated in Fig. 6, can be approximated by the first derivative of a Gaussian $G'(x)$, where

$$G(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right).$$

The reason for doing this is that there are very efficient ways to compute the two-dimensional extension of the filter if it can be represented as some derivative of a Gaussian. This is described in detail elsewhere [4], but for the present we will compare the theoretical performance of a first derivative of a Gaussian filter to the optimal operator. The impulse response of the first derivative filter is

$$f(x) = -\frac{x}{\chi^2}\exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{42}$$

and the terms in the performance criteria have the values

$$|f'(0)| = \frac{1}{\sigma_s}$$

$$\int_{-\infty}^{0} f(x)\,dx = 1 \qquad \int_{-\infty}^{+\infty} f^2(x)\,dx = \frac{\sqrt{\pi}}{2\sigma}$$

$$\int_{-\infty}^{+\infty} f'^2(x)\,dx = \frac{3\sqrt{\pi}}{4\sigma^3} \qquad \int_{-\infty}^{+\infty} f''^2(x)\,dx = \frac{15\sqrt{\pi}}{8\sigma^5}. \tag{43}$$

The overall performance index for this operator is

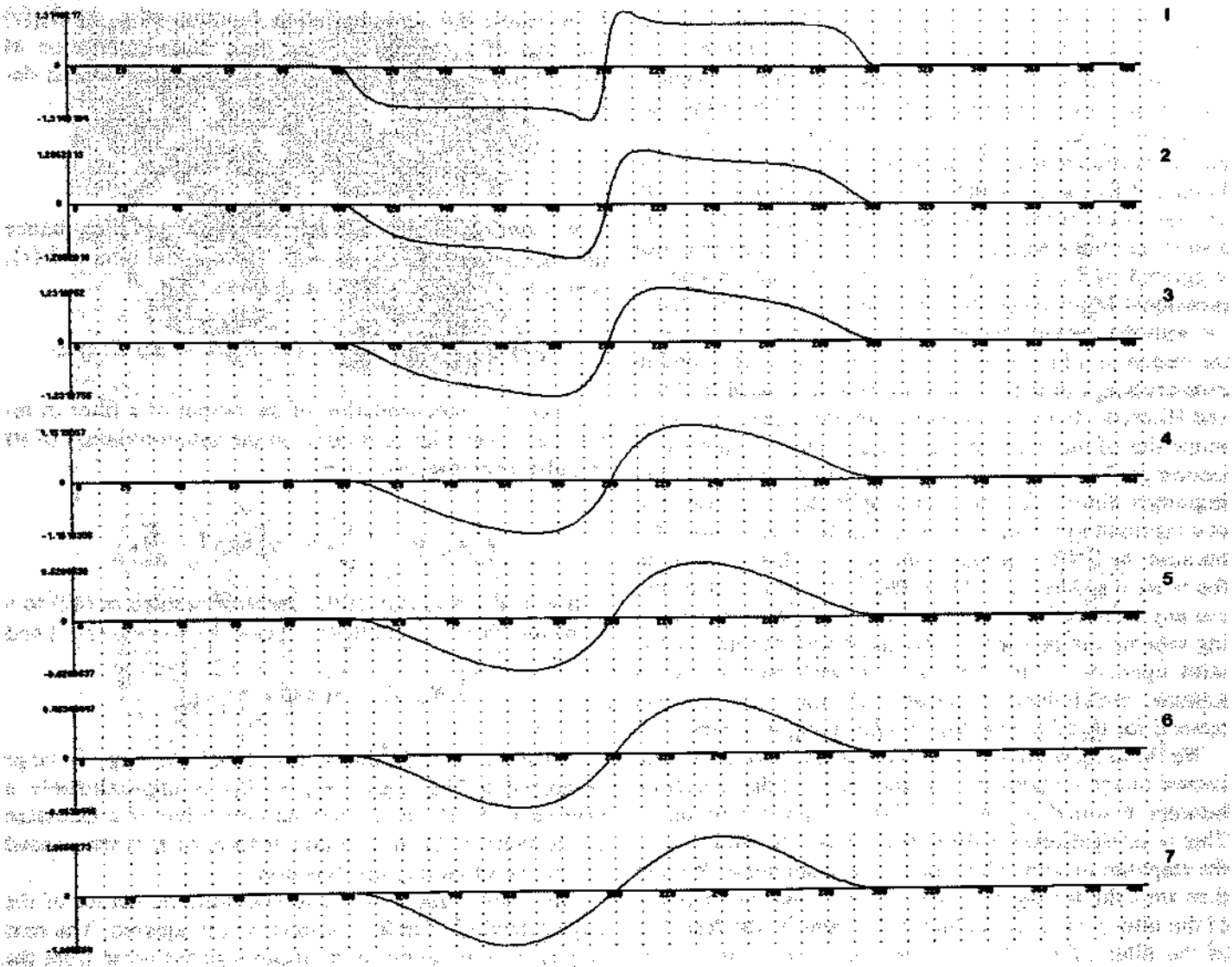


Fig. 5. Optimal step edge operators for various values of x_{max}. From top to bottom, they are $x_{max} = 0.15, 0.3, 0.5, 0.8, 1.0, 1.2, 1.4$.

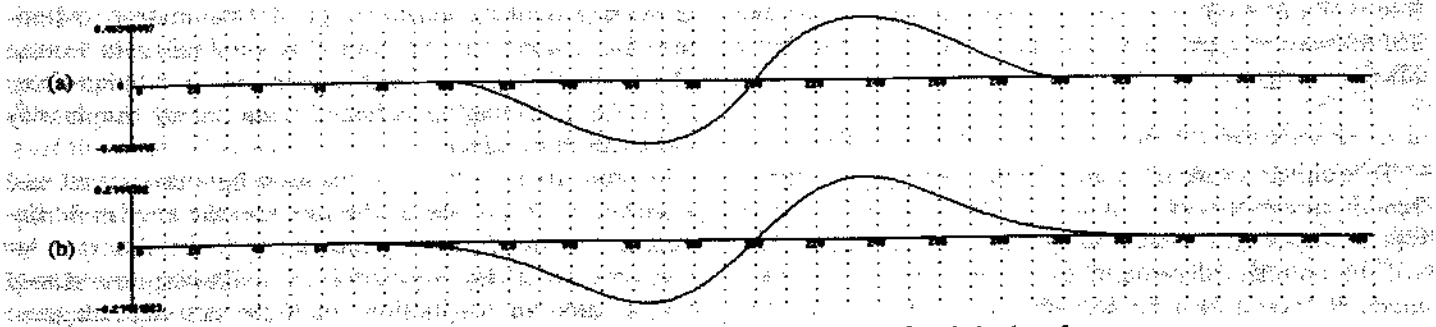


Fig. 6. (a) The optimal step edge operator. (b) The first derivative of a Gaussian.

$$\Sigma\Lambda = \sqrt{\frac{8}{3\pi}} \approx 0.92 \qquad (44)$$

while the r value is, from (41),

$$r = \sqrt{\frac{4}{15}} \approx 0.51.$$

The performance of the first derivative of Gaussian operator above is worse than the optimal operator by about 20 percent and its multiple response measure r, is worse by about 10 percent. It would probably be difficult to detect a difference of this magnitude by looking at the performance of the two operators on real images, and because the first derivative of Gaussian operator can be computed with much less effort in two dimensions, it has

been used exclusively in experiments. The impulse responses of the two operators can be compared in Fig. 6.

A close approximation of the first derivative of Gaussian operator was suggested by Macleod [16] for step edge detection. Macleod's operator is a difference of two displaced two-dimensional Gaussians. It was evaluated in Fram and Deutsch [7] and compared very favorably with several other schemes considered in that paper. There are also strong links with the Laplacian of Gaussian operator suggested by Marr and Hildreth [18]. In fact, a one-dimensional Marr–Hildreth edge detector is almost identical with the operator we have derived because maxima in the output of a first derivative operator will correspond to zero-crossings in the Laplacian operator as used by Marr and Hildreth. In two dimensions however, the directional properties of our detector enhance its detection and localization performance compared to the Laplacian. Another important difference is that the amplitude of the response at a maximum provides a good estimate of edge strength, because the SNR criterion is the ratio of this response to the noise response. The Marr–Hildreth operator does not use any form of thresholding, but an adaptive thresholding scheme can be used to advantage with our first derivative operator. In the next section we describe such a scheme, which includes noise estimation and a novel method for thresholding edge points along contours.

We have derived our optimal operator to deal with known image features in Gaussian noise. Edge detection between textured regions is another important problem. This is straightforward if the texture can be modelled as the response of some filter $t(x)$ to Gaussian noise. We can then treat the texture as a noise signal, and the response of the filter $f(x)$ to the texture is the same as the response of the filter $(f * t)(x)$ to Gaussian noise. Making this replacement in each integral in the performance criteria that computes a noise response gives us the texture edge design problem. The generalization to other types of texture is not as easy, and for good discrimination between known texture types, a better approach would involve a Markov image model as in [5].

VI. Noise Estimation and Thresholding

To estimate noise from an operator output, we need to be able to separate its response to noise from the response due to step edges. Since the performance of the system will be critically dependent on the accuracy of this estimate, it should also be formulated as an optimization. Wiener filtering is a method for optimally estimating one component of a two-component signal, and can be used to advantage in this application. It requires knowledge of the autocorrelation functions of the two components and of the combined signal. Once the noise component has been optimally separated, we form a global histogram of noise amplitude, and estimate the noise strength from some fixed percentile of the noise signal.

Let $g_1(x)$ be the signal we are trying to detect (in this case the noise output), and $g_2(x)$ be some disturbance (paradoxically this will be the edge response of our filter), then denote the autocorrelation function of g_1 as $R_{11}(\tau)$ and that of g_2 as $R_{22}(\tau)$, and their cross-correlation as $R_{12}(\tau)$, where the correlation of two real functions is defined as follows:

$$R_{ij}(\tau) = \int_{-\infty}^{+\infty} g_i(x)\, g_j(x + \tau)\, dx.$$

We assume in this case that the signal and disturbance are uncorrelated, so $R_{12}(\tau) = 0$. The optimal filter is $K(x)$, which is implicitly defined as follows [30]:

$$R_{11}(\tau) = \int_{-\infty}^{+\infty} (R_{11}(\tau - x) + R_{22}(\tau - x))\, K(x)\, dx.$$

Since the autocorrelation of the output of a filter in response to white noise is equal to the autocorrelation of its impulse response, we have

$$R_{11}(x) = k_3\left(\frac{x^2}{2\sigma^2} - 1\right) \exp\left(-\frac{x^2}{4\sigma^2}\right)$$

If g_2 is the response of the operator derived in (42) to a step edge then we will have $g_2(x) = k \exp(-x/2\sigma^2)$ and

$$R_{22}(x) = k_2 \exp\left(-\frac{x^2}{4\sigma^2}\right)$$

In the case where the amplitude of the edge is large compared to the noise, $R_{22} + R_{11}$ is approximately a Gaussian and R_{11} is the second derivative of a Gaussian of the same σ. Then the optimal form of K is the second derivative of an impulse function.

The filter K above is convolved with the output of the edge detection operator and the result is squared. The next step is the estimation of the mean-squared noise from the local values. Here there are several possibilities. The simplest is to average the squared values over some neighborhood, either using a moving average filter or by taking an average over the entire image. Unfortunately, experience has shown that the filter K is very sensitive to step edges, and that as a consequence the noise estimate from any form of averaging is heavily colored by the density and strength of edges.

In order to gain better separation between signal and noise we can make use of the fact that the amplitude distribution of the filter response tends to be different for edges and noise. By our model, the noise response should have a Gaussian distribution, while the step edge response will be composed of large values occurring very infrequently. If we take a histogram of the filter values, we should find that the positions of the low percentiles (say less than 80 percent) will be determined mainly the noise energy, and that they are therefore useful estimators for noise. A global histogram estimate is actually used in the current implementation of the algorithm.

Even with noise estimation, the edge detector will be susceptible to streaking if it uses only a single threshold. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the

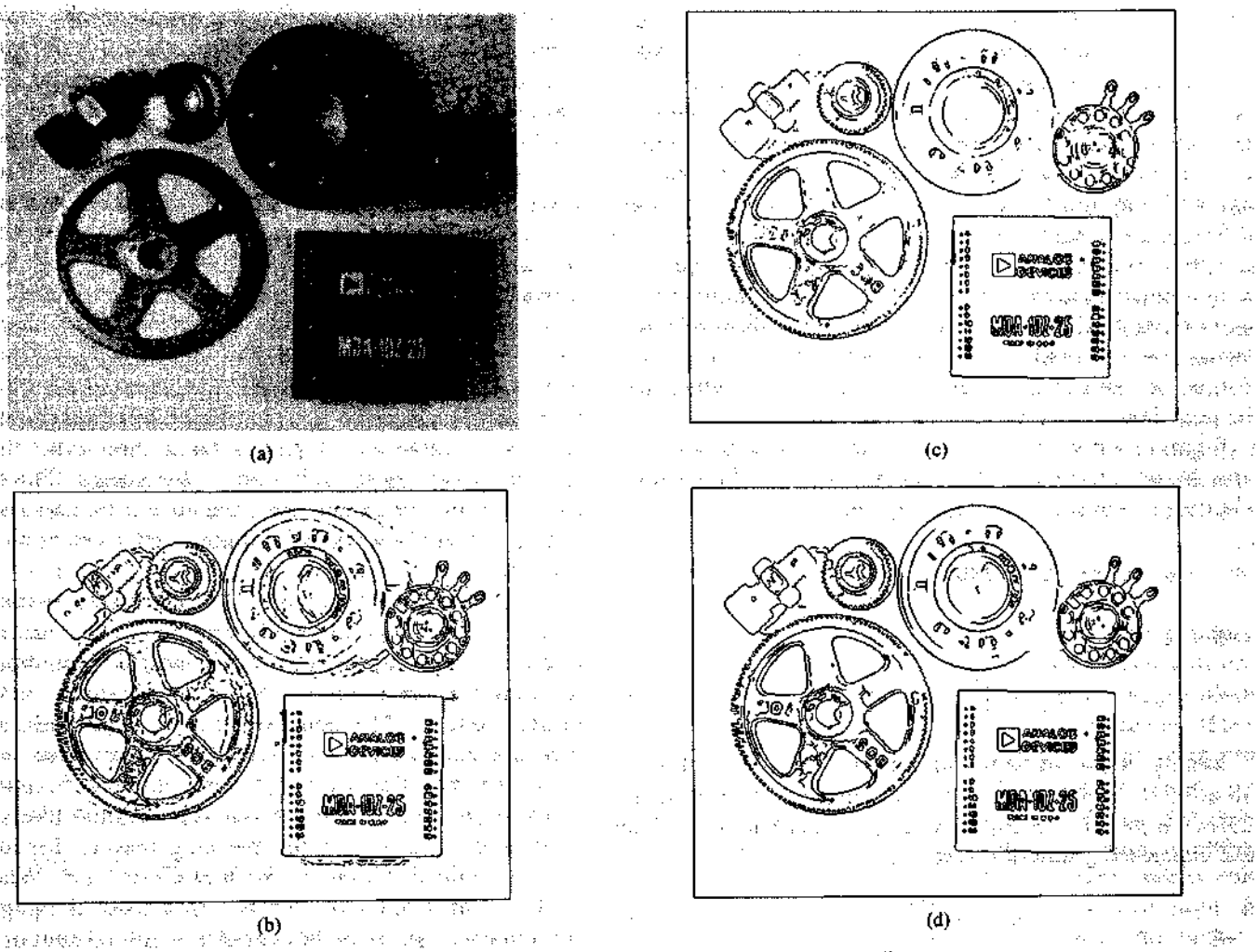


Fig. 7. (a) Parts image, 576 by 454 pixels. (b) Image thesholded at T_1. (c) Image thresholded at 2 T_1. (d) Image thresholded with hysteresis using both the thresholds in (a) and (b).

threshold along the length of the contour. Suppose we have a single threshold set at T_1, and that there is an edge in the image such that the response of the operator has mean value T_1. There will be some fluctuation of the output amplitude due to noise, even if the noise is very slight. We expect the contour to be above threshold only about half the time. This leads to a broken edge contour. While this is a pathological case, streaking is a very common problem with edge detectors that employ thresholding. It is very difficult to set a threshold so that there is small probability of marking noise edges while retaining high sensitivity. An example of the effect of streaking is given in Fig. 7.

One possible solution to this problem, used by Pentland [22] with Marr–Hildreth zero-crossings, is to average the edge strength of a contour over part of its length. If the average is above the threshold, the entire segment is marked. If the average is below threshold, no part of the contour appears in the output. The contour is segmented by breaking it at maxima in curvature. This segmentation is necessary in the case of zero-crossings since the zero-crossings always form closed contours, which obviously do not always correspond to contours in the image.

In the current algorithm, no attempt is made to presegment contours. Instead the thresholding is done with hysteresis. If any part of a contour is above a high threshold, those points are immediately output, as is the entire connected segment of contour which contains the points and which lies above a low threshold. The probability of streaking is greatly reduced because for a contour to be broken it must now fluctuate above the high threshold and below the low threshold. Also the probability of isolated false edge points is reduced because the strength of such points must be above a higher threshold. The ratio of the high to low threshold in the implementation is in the range two or three to one.

VII. Two or More Dimensions

In one dimension we can characterize the position of a step edge in space with one position coordinate. In two dimensions an edge also has an orientation. In this section we will use the term "edge direction" to mean the direction of the tangent to the contour that the edge defines in two dimensions. Suppose we wish to detect edges of a particular orientation. We create a two-dimensional mask for this orientation by convolving a linear edge detection

function aligned normal to the edge direction with a projection function parallel to the edge direction. A substantial savings in computational effort is possible if the projection function is a Gaussian with the same σ as the (first derivative of the) Gaussian used as the detection function. It is possible to create such masks by convolving the image with a symmetric two-dimensional Gaussian and then differentiating normal to the edge direction. In fact we do not have to do this in every direction because the slope of a smooth surface in any direction can be determined exactly from its slope in two directions. This form of directional operator, while simple and inexpensive to compute, forms the heart of the more elaborate detector which will be described in the next few sections.

Suppose we wish to convolve the image with an operator G_n which is the first derivative of a two-dimensional Gaussian G in some direction $\boldsymbol{n}$, i.e.,

$$G = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

and

$$G_n = \frac{\partial G}{\partial \boldsymbol{n}} = \boldsymbol{n} \cdot \nabla G. \tag{45}$$

Ideally, $\boldsymbol{n}$ should be oriented normal to the direction of an edge to be detected, and although this direction is not known *a priori*, we can form a good estimate of it from the smoothed gradient direction

$$\boldsymbol{n} = \frac{\nabla(G * I)}{|\nabla(G * I)|} \tag{46}$$

where $*$ denotes convolution. This turns out to be a very good estimator for edge normal direction for steps, since a smoothed step has strong gradient normal to the edge. It is exact for straight line edges in the absence of noise, and the Gaussian smoothing keeps it relatively insensitive to noise.

An edge point is defined to be a local maximum (in the direction $\boldsymbol{n}$) of the operator G_n applied to the image I. At a local maximum, we have

$$\frac{\partial}{\partial \boldsymbol{n}} G_n * I = 0$$

and substituting for G_n from (45) and associating Gaussian convolution, the above becomes

$$\frac{\partial^2}{\partial \boldsymbol{n}^2} G * I = 0. \tag{47}$$

At such an edge point, the edge strength will be the magnitude of

$$|G_n * I| = |\nabla(G * I)|. \tag{48}$$

Because of the associativity of convolution, we can first convolve with a symmetric Gaussian G and then compute directional second derivative zeros to locate edges (47), and use the magnitude of (48) to estimate edge strength. This is equivalent to detecting and locating the edge using the directional operator G_n, but we need not know the direction $\boldsymbol{n}$ before convolution.

The form of nonlinear second derivative operator in (47) has also been used by Torre and Poggio [28] and by Haralick [10]. It also appears in Prewitt [23] in the context of edge enhancement. A rather different two-dimensional extension is proposed by Spacek [26] who uses one-dimensional filters aligned normal to the edge direction but without extending them along the edge direction. Spacek starts with a one-dimensional formulation which maximizes the product of the three performance criteria defined in Section II, and leads to a step edge operator which differs slightly from the one we derived in Section IV. Gennert [8] addresses the two-dimensional edge detector problem directly, and applies a set of directional first derivative operators at each point in the image. The operators have limited extent along the edge direction and produce good results at sharp changes in edge orientation and corners.

The operator (47) actually locates either maxima or minima by locating the zero-crossings in the second derivative in the edge direction. In principle it could be used to implement an edge detector in an arbitrary number of dimensions, by first convolving the image with a symmetric n-dimensional Gaussian. The convolution with an n-dimensional Gaussian is highly efficient because the Gaussian is separable into n one-dimensional filters.

But there are other more pressing reasons for using a smooth projection function such as a Gaussian. When we apply a linear operator to a two-dimensional image, we form at every point in the output a weighted sum of some of the input values. For the edge detector described here, this sum will be a difference between local averages on different sides of the edge. This output, before nonmaximum suppression, represents a kind of moving average of the image. Ideally we would like to use an infinite projection function, but real edges are of limited extent. It is therefore necessary to window the projection function [9]. If the window function is abruptly truncated, e.g., if it is rectangular, the filtered image will not be smooth because of the very high bandwidth of this window. This effect is related to the Gibbs phenomenon in Fourier theory which occurs when a signal is transformed over a finite window. When nonmaximum suppression is applied to this rough signal we find that edge contours tend to "wander" or that in severe cases they are not even continuous.

The solution is to use a smooth window function. In statistics, the Hamming and Hanning windows are typically used for moving averages. The Gaussian is a reasonable approximation to both of these, and it certainly has very low bandwidth for a given spatial width. (The Gaussian is the unique function with minimal product of bandwidth and frequency.) The effect of the window function becomes very marked for large operator sizes and it is probably the biggest single reason why operators with large support were not practical until the work of Marr and Hildreth on the Laplacian of Gaussian.

It is worthwhile here to compare the performance of

this kind of directional second derivative operator with the Laplacian. First we note that the two-dimensional Laplacian can be decomposed into components of second derivative in two arbitrary orthogonal directions. If we choose to take one of the derivatives in the direction of principal gradient, we find that the operator output will contain one contribution that is essentially the same as the operator described above, and also a contribution that is aligned along the edge direction. This second component contributes nothing to localization or detection (the surface is roughly constant in this direction), but increases the output noise.

In later sections we will describe an edge detector which incorporates operators of varying orientation and aspect ratio, but these are a superset of the operators used in the simple detector described above. In typical images, most of the edges are marked by the operators of the smallest width, and most of these by nonelongated operators. The simple detector performs well enough in these cases, and as detector complexity increases, performance gains tend to diminish. However, as we shall see in the following sections, there are cases when larger or more directional operators should be used, and that they do improve performance when they are applicable. The key to making such a complicated detector produce a coherent output is to design effective decision procedures for choosing between operator outputs at each point in the image.

VIII. The Need for Multiple Widths

Having determined the optimal shape for the operator, we now face the problem of choosing the width of the operator so as to give the best detection/localization tradeoff in a particular application. In general the signal-to-noise ratio will be different for each edge within an image, and so it will be necessary to incorporate several widths of operator in the scheme. The decision as to which operator to use must be made dynamically by the algorithm and this requires a local estimate of the noise energy in the region surrounding the candidate edge. Once the noise energy is known, the signal-to-noise ratios of each of the operators will be known. If we then use a model of the probability distribution of the noise, we can effectively calculate the probability of a candidate edge being a false edge (for a given edge, this probability will be different for different operator widths).

If we assume that the *a priori* penalty associated with a falsely detected edge is independent of the edge strength, it is appropriate to threshold the detector outputs on probability of error rather than on magnitude of response. Once the probability threshold is set, the minimum acceptable signal-to-noise ratio is determined. However, there may be several operators with signal-to-noise ratios above the threshold, and in this case the smallest operator should be chosen, since it gives the best localization. We can afford to be conservative in the setting of the threshold since edges missed by the smallest operators may be picked up by the larger ones. Effectively the global tradeoff between error rate and localization remains, since choosing a high signal-to-noise ratio threshold leads to a lower error rate, but will tend to give poorer localization since fewer edges will be recorded from the smaller operators.

In summary then, the first heuristic for choosing between operator outputs is that *small operator widths should be used whenever they have sufficient* Σ. This is similar to the selection criterion proposed by Marr and Hildreth [18] for choosing between different Laplacian of Gaussian channels. In their case the argument was based on the observation that the smaller channels have higher resolution, i.e., there is less possibility of interference from neighboring edges. That argument is also very relevant in the present context, as to date there has been no consideration of the possibility of more than one edge in a given operator support. Interestingly, Rosenfeld and Thurston [25] proposed exactly the opposite criterion in the choice of operator for edge detection in texture. The argument given was that the larger operators give better averaging and therefore (presumably) better signal-to-noise ratios.

Taking the fine-to-coarse heuristic as a starting point, we need to form a local decision procedure that will enable us to decide whether to mark one or more edges when several operators in a neighborhood are responding. If the operator with the smallest width responds to an edge and if it has a signal-to-noise ratio above the threshold, we should immediately mark an edge at that point. We now face the problem that there will almost certainly be edges marked by the larger operators, but that these edges will probably not be exactly coincident with the first edge. A possible answer to this would be to suppress the outputs of all nearby operators. This has the undesirable effect of preventing the large channels for responding to "fuzzy" edges that are superimposed on the sharp edge.

Instead we use a "feature synthesis" approach. We begin by marking all the edges from the smallest operators. From these edges, we synthesize the large operator outputs than would have been produced if these were the only edges in the image. We then compare the actual operator outputs to the synthetic outputs. We mark additional edges only if the large operator has significantly greater response than what we would predict from the synthetic output. The simplest way to produce the synthetic outputs is to take the edges marked by a small operator in a particular direction, and convolve with a Gaussian normal to the edge direction for this operator. The σ of this Gaussian should be the same as the σ of the large channel detection filter.

This procedure can be applied repeatedly to first mark the edges from the second smallest scale that were not marked by at the first, and then to find the edges from the third scale that were not marked by either of the first two, etc. Thus we build up a cumulative edge map by adding those edges at each scale that were not marked by smaller scales. It turns out that in many cases the majority of edges are picked up by the smallest channel, and the later channels mark mostly shadow and shading edges, or edges between textured regions.

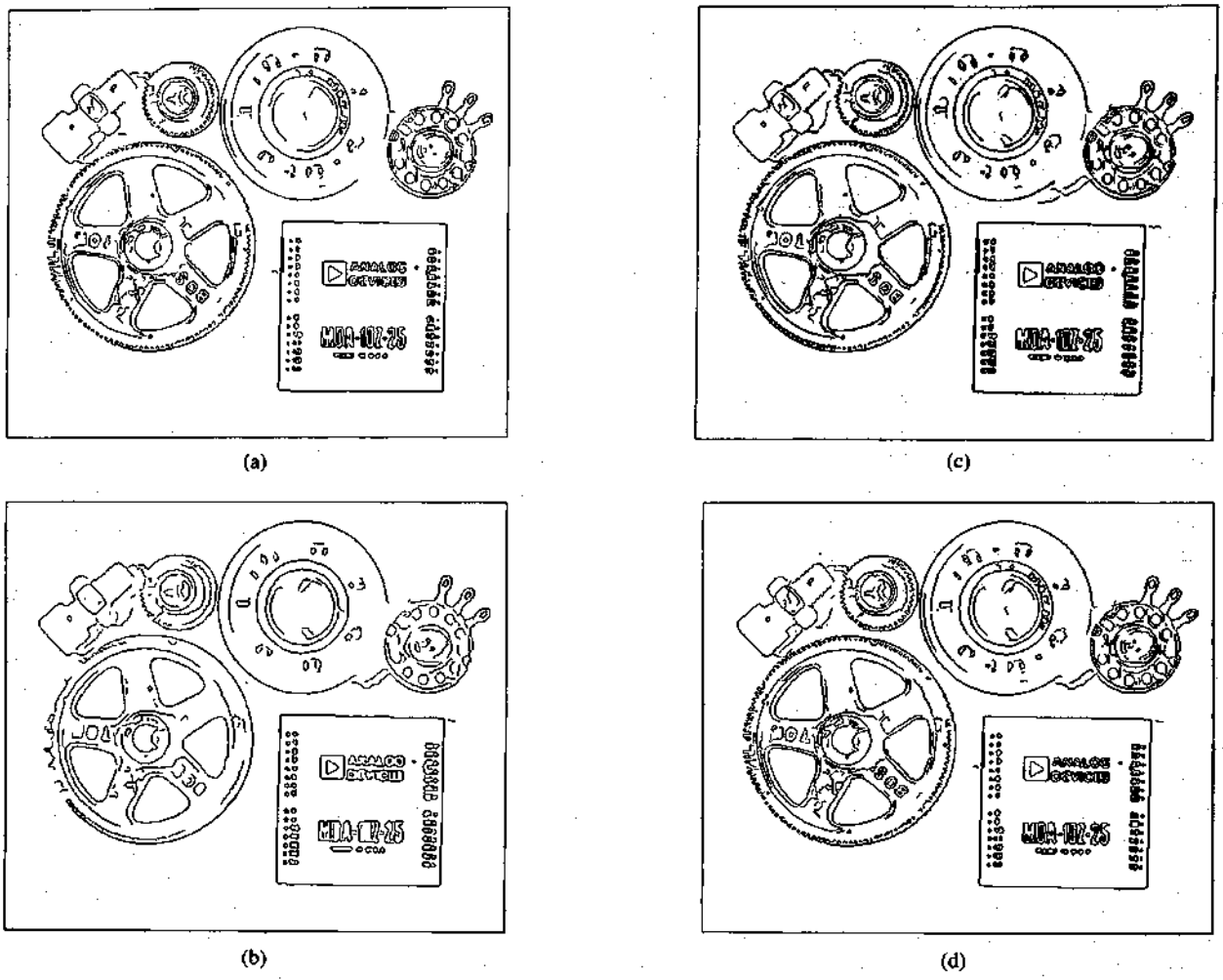


Fig. 8. (a) Edges from parts image at $\sigma = 1.0$. (b) Edges at $\sigma = 2.0$. (c) Superposition of the edges. (d) Edges combined using feature synthesis.

Some examples of feature synthesis applied to some sample images are shown in Figs. 8 and 9. Notice that most of the edges in Fig. 8 are marked by the smaller scale operator, and only a few additional edges, mostly shadows, are picked up by the coarser scale. However when the two sets of edges are superimposed, we notice that in many cases the responses of the two operators to the same edge are not spatially coincident. When feature synthesis is applied we find that redundant responses of the larger operator are eliminated leading to a sharp edge map.

By contrast, in Fig. 9 the edges marked by the two operators are essentially independent, and direct superposition of the edges gives a useful edge map. When we apply feature synthesis to these sets of edges we find that most of the edges at the coarser scale remain. Both Figs. 8 and 9 were produced by the edge detector with exactly the same set of parameters (other than operator size), and they were chosen to represent opposing extremes of image content across scale.

IX. The Need for Directional Operators

So far we have assumed that the projection function is a Gaussian with the same σ as the Gaussian used for the detection function. In fact both the detection and localization of the operator improve as the length of the projection function increases. We now prove this for the operator signal-to-noise ratio. The proof for localization is similar. We will consider a step edge in the x direction which passes through the origin. This edge can be represented by the equation

$$I(x, y) = Au_{-1}(y)$$

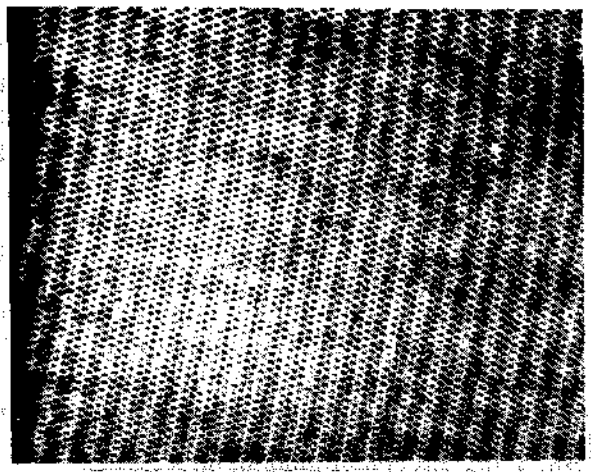
where u_{-1} is the unit step function, and A is the amplitude of the edge as before. Suppose that there is additive Gaussian noise of mean squared value n_{00}^2 per unit area. If we convolve this signal with a filter whose impluse response is $f(x, y)$, then the response to the edge (at the origin) is

$$\int_{-\infty}^{0} \int_{-\infty}^{+\infty} f(x, y)\, dx\, dy.$$

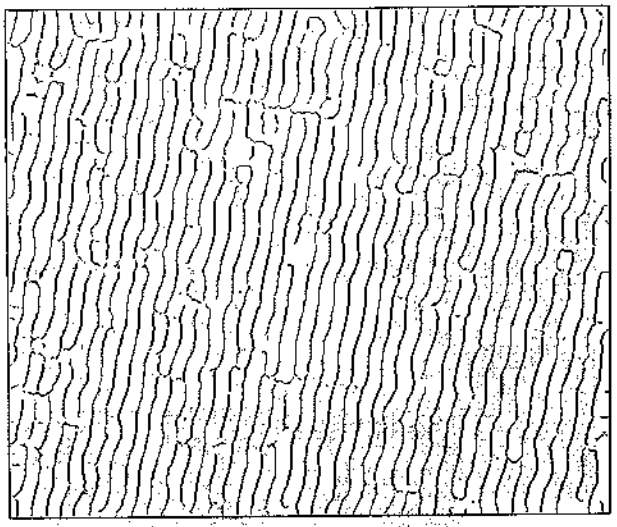
The root mean squared response to the noise only is

$$n_{00} \left(\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f^2(x, y)\, dx\, dy \right)^{1/2}$$

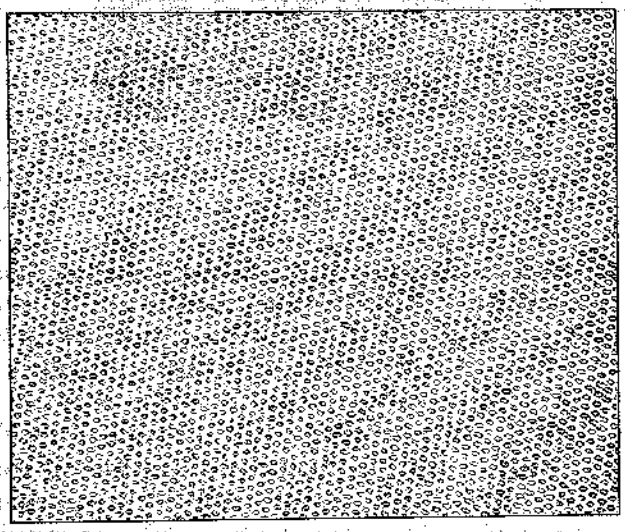
The signal-to-noise ratio is the quotient of these two



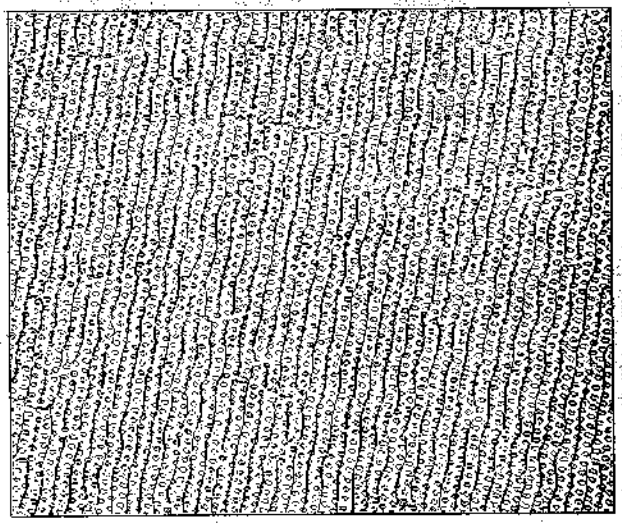
(a)



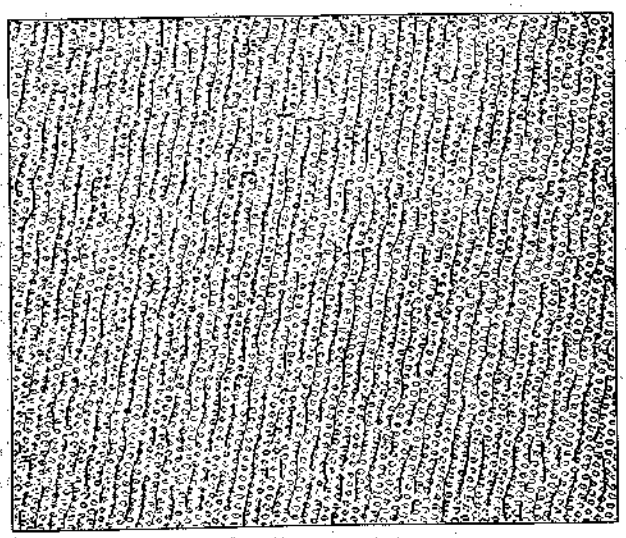
(c)



(b)



(d)



(e)

Fig. 9. (a) Handywipe image 576 by 454 pixels. (b) Edges from handywipe image at $\sigma = 1.0$. (c) $\sigma = 5.0$. (d) Superposition of the edges. (e) Edges combined using feature synthesis.

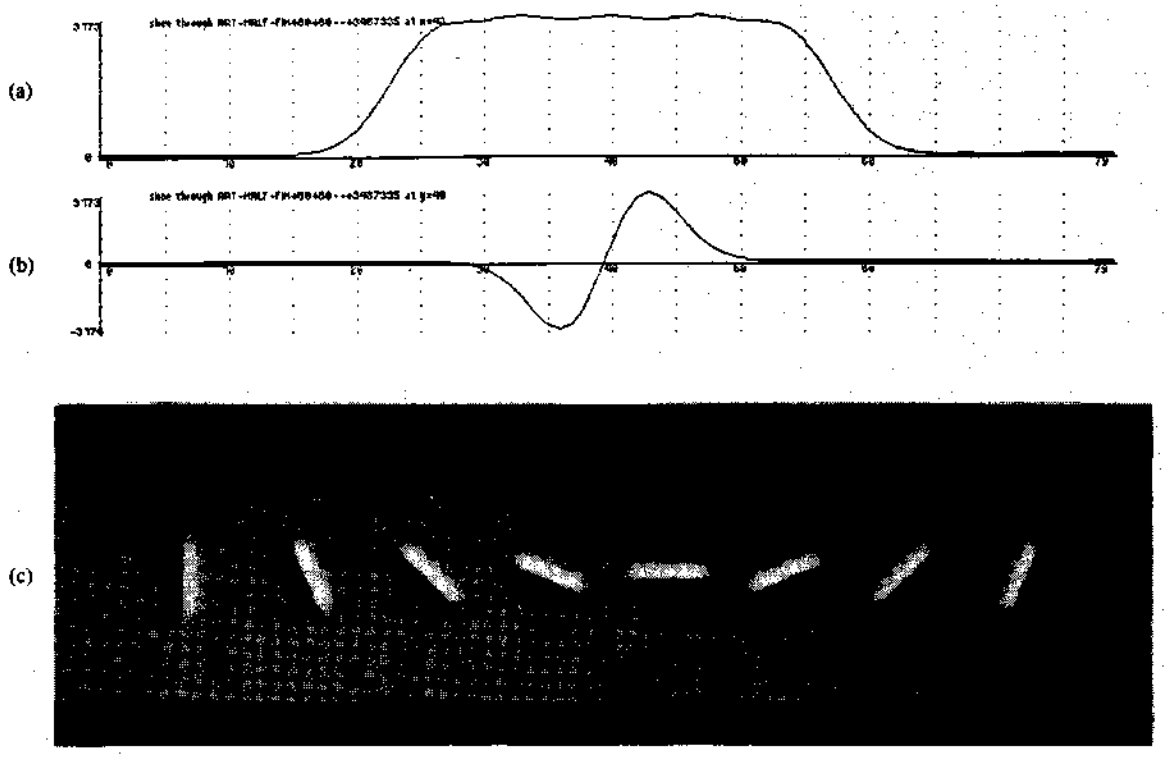


Fig. 10. Directional step edge mask. (a) Cross section parallel to the edge direction. (b) Cross section normal to edge direction. (c) Two-dimensional impulse responses of several masks.

integrals, and will be denoted by Σ. We have already seen what happens if we scale the function normal to the edge (21). We now do the same to the projection function by replacing $f(x, y)$ by $f_l(x, y) = f(x, (y/l))$. The integrals become

$$\int_{-\infty}^{0}\int_{-\infty}^{+\infty} f\left(x, \frac{y}{l}\right) dx\, dy = \int_{-\infty}^{0}\int_{-\infty}^{+\infty} f(x, y_1) l\, dx\, dy_1$$

$$n_{00}\left(\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f^2\left(x, \frac{y}{l}\right) dx\, dy\right)^{1/2} = n_{00}\left(\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f^2(x, y_1) l\, dx\, dy_1\right)^{1/2} \quad (49)$$

And the ratio of the two is now $\sqrt{l}\Sigma$. The localization Λ also improves as $\sqrt{l}$. It is clearly desirable that we use as large a projection function as possible. There are practical limitations on this however, in particular edges in an image are of limited extent, and few are perfectly linear. However, most edges continue for some distance, in fact much further than the 3 or 4 pixel supports of most edge operators. Even curved edges can be approximated by linear segments at a small enough scale. Considering the advantages, it is obviously preferable to use directional operators whenever they are applicable. The only proviso is that the detection scheme must ensure that they are used only when the image fits a linear edge model.

The present algorithm tests for applicability of each directional mask by forming a goodness-of-fit estimate. It does this at the same time as the mask itself is computed. An efficient way of forming long directional masks is to sample the output of nonelongated masks with the same direction. This output is sampled at regular intervals in a line parallel to the edge direction. If the samples are close together (less than 2σ apart), the resulting mask is essentially flat over most of its range in the edge direction and falls smoothly off to zero at its ends. Two cross sections of such a mask are shown in Fig. 10. In this diagram (as in the present implementation) there are five samples over the operator support.

Simultaneously with the computation of the mask, it is possible to establish goodness of fit by a simple squared-error measure. The mask is computed by summing some number of circular mask outputs (say 5) in a line. If the mask lies over a step edge in its preferred direction, these 5 values will be roughly the same. If the edge is curved or not aligned with the mask direction, the values will vary. We use the variance of these values as an estimate of the goodness of fit of the actual edge to an ideal step model. We then suppress the output of a directional mask if its variance is greater than some fraction of the squared output. Where no directional operator has sufficient goodness of fit at a point, the algorithm will use the output of the nonelongated operator described in Section VII. This simple goodness-of-fit measure is sufficient to eliminate the problems that traditionally plague directional operators, such as false responses to highly curved edges and extension of edges beyond corners; see Hildreth [12].

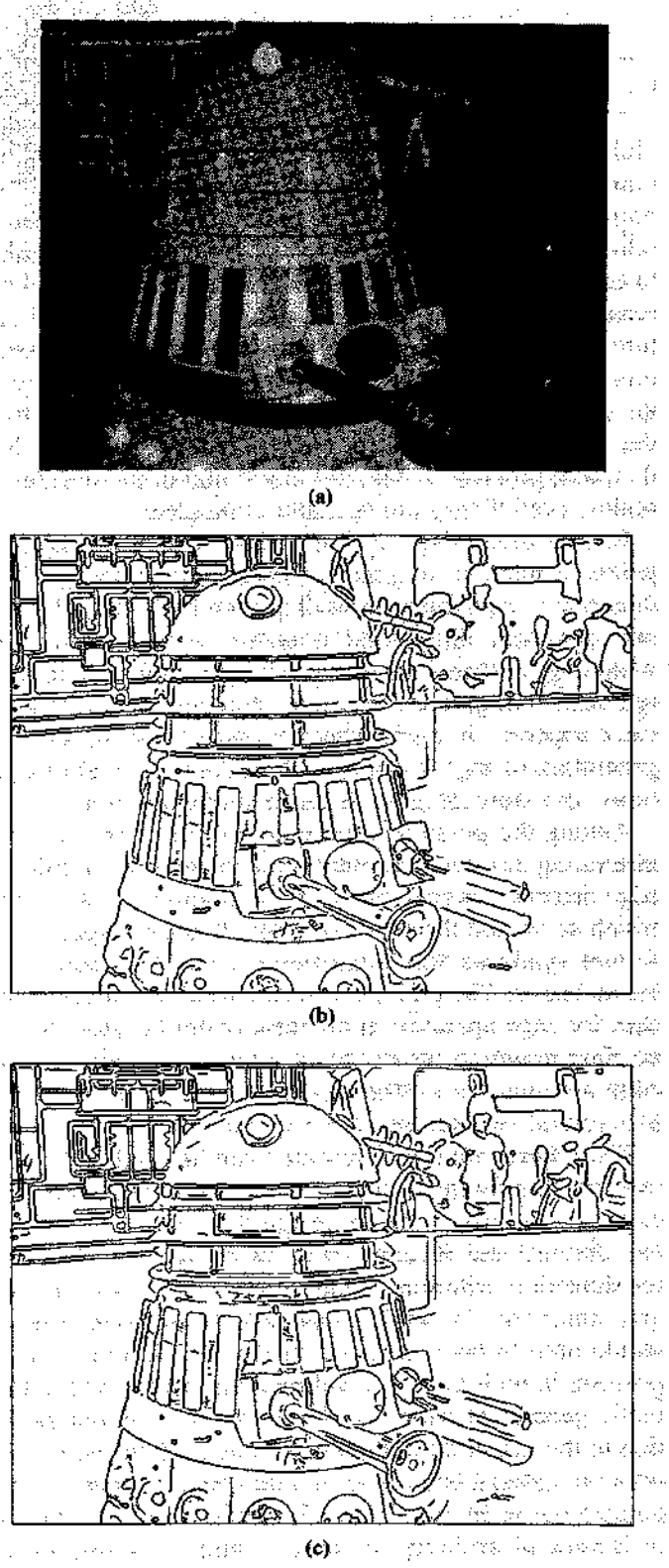
This particular form of projection function, that is a

function with constant value over some range which decays to zero at each end with two roughly half-Gaussians, is very similar to a commonly used extension of the Hanning window. This latter function is flat for some distance and decays to zero at each end with two half-cosine bells [2]. We can therefore expect our function to have good properties as a moving average estimator, which as we saw in Section VII, is an important role fulfilled by the projection function.

All that remains to be done in the design of directional operators is the specification of the number of directions, or equivalently the angle between two adjacent directions. To determine the latter, we need to determine the angular selectivity of a directional operator as a function of the angle θ between the edge direction and the preferred direction of the operator. Assume that we form the operator by taking an odd number $2N + 1$ of samples. Let the number of a sample be n where n is in the range $-N \cdots +N$. Recall that the directional operator is formed by convolving with a symmetric Gaussian, differentiating normal to the preferred edge direction of the operator, and then sampling along the preferred direction. The differentiated surface will be a ridge which makes an angle θ to the preferred edge direction. Its height will vary as $\cos \theta$, and the distance of the nth sample from the center of the ridge will be $nd \sin \theta$ where d is the distance between samples. The normalized output will be

$$O_n(\theta) = \frac{\cos \theta}{2N + 1} \left[\sum_{n=-N}^{N} \exp \left(-\frac{(nd \sin \theta)^2}{2\sigma^2} \right) \right]. \quad (50)$$

If there are m operator directions, then the angle between the preferred directions of two adjacent operators will be $180/m$. The worst case angle between the edge and the nearest preferred operator direction is therefore $90/m$. In the current implementation the value of d/σ is about 1.4 and there are 6 operator directions. The worst case for θ is 15 degrees, and for this case the operator output will fall to about 85 percent of its maximum value. Directional operators very much like the ones we have derived were suggested by Marr [17], but were discarded in favor of the Laplacian of Gaussian [18]. In part this was because the computation of several directional operators at each point in the image was thought to require an excessive amount of computation. In fact the sampling scheme described above requires only five multiplications per operator. An example of edge detection using five-point directional operators is given in Fig. 11.



(a) (b) (c)

Fig. 11. (a) Dalek image 576 by 454 pixels. (b) Edges found using circular operator. (c) Directional edges (6 mask orientations).

X. Conclusions

We have described a procedure for the design of edge detectors for arbitrary edge profiles. The design was based on the specification of detection and localization criteria in a mathematical form. It was necessary to augment the original two criteria with a multiple response measure in order to fully capture the intuition of good detection. A mathematical form for the criteria was presented, and numerical optimization was used to find optimal operators for roof and ridge edges. The analysis was then restricted to consideration of optimal operators for step edges. The result was a class of operators related by spatial scaling. There was a direct tradeoff in detection performance versus localization, and this was determined by the spatial

width. The impulse response of the optimal step edge operator was shown to approximate the first derivative of a Gaussian.

A detector was proposed which used adaptive thresholding with hysteresis to eliminate streaking of edge contours. The thresholds were set according to the amount of noise in the image, as determined by a noise estimation scheme. This detector made use of several operator widths to cope with varying image signal-to-noise ratios, and operator outputs were combined using a method called feature synthesis, where the responses of the smaller operators were used to predict the large operator responses. If the actual large operator outputs differ significantly from the predicted values, new edge points are marked. It is therefore possible to describe edges that occur at different scales, even if they are spatially coincident.

In two dimensions it was shown that marking edge points at maxima of gradient magnitude in the gradient direction is equivalent to finding zero-crossings of a certain nonlinear differential operator. It was shown that when edge contours are locally straight, highly directional operators will give better results than operators with a circular support. A method was proposed for the efficient generation of highly directional masks at several orientations, and their integration into a single description.

Among the possible extensions of the work, the most interesting unsolved problem is the integration of different edge detector outputs into a single description. A scheme which combined the edge and ridge detector outputs using feature synthesis was implemented, but the results were inconclusive. The problem is much more complicated here than for edge operators at different scales because there is no clear reason to prefer one edge type over another. Each edge set must be synthesized from the other, without a bias caused by overestimation in one direction.

The criteria we have presented can be used with slight modification for the design of other kinds of operator. For example, we may wish to design detectors for nonlinear two-dimensional features (such as corners). In this case the detection criterion would be a two-dimensional integral similar to (3), while a plausible localization criterion would need to take into account the variation of the edge position in both the x and y directions, and would not directly generalize from (9). There is a natural generalization to the detection of higher-dimensional edges, such as occur at material boundaries in tomographic scans. As was pointed out in Section VII, (47) can be used to find edges in images of arbitrary dimension, and the algorithm remains efficient in higher dimensions because n-dimensional Gaussian convolution can be broken down into n linear convolutions.

Acknowledgment

The author would like to thank Dr. J. M. Brady for his influence on the course of this work and for comments on early drafts of this paper. Thanks to the referees for their suggestions which have greatly improved the presentation of the paper. In particular thanks to the referee who suggested the simple derivation based on the Schwarz inequality that appears on p. 682.

References

[1] R. J. Beattie, "Edge detection for semantically based early visual processing," Ph.D. dissertation, Univ. Edinburgh, 1984.

[2] C. Bingham, M. D. Godfrey, and J. W. Tukey, "Modern techniques of power spectrum estimation," *IEEE Trans. Audio Electroacoust.*, vol. AU-15, no. 2, pp. 56-66, 1967.

[3] R. A. Brooks, "Symbolic reasoning among 3-D models and 2-D images," Dep. Comput. Sci., Stanford Univ., Stanford, CA, Rep. AIM-343, 1981.

[4] J. F. Canny, "Finding edges and lines in images," M.I.T. Artificial Intell. Lab., Cambridge, MA, Rep. AI-TR-720, 1983.

[5] F. S. Cohen, D. B. Cooper, J. F. Silverman, and E. B. Hinkle, "Simple parallel hierarchical and relaxation algorithms for segmenting textured images based on noncasual Markovian random field models," in *Proc. 7th Int. Conf. Pattern Recognition and Image Processing*, Canada, 1984.

[6] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, vol. 1. New York: Wiley-Interscience, 1953.

[7] J. R. Fram and E. S. Deutsch, "On the quantitative evaluation of edge detection schemes and their comparison with human performance," *IEEE Trans. Comput.*, vol. C-24, no. 6, pp. 616-628, 1975.

[8] M. Gennert, "Detecting half-edges and vertices in images," in *IEEE Conf. Comput. Vision and Pattern Recognition*, Miami Beach, FL, June 24-26, 1986.

[9] R. W. Hamming, *Digital Filters*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[10] R. M. Haralick, "Zero-crossings of second directional derivative edge operator," in *SPIE Proc. Robot Vision*, Arlington, VA, 1982.

[11] A. Herskovits and T. O. Binford, "On boundary detection," M.I.T. Artificial Intell. Lab., Cambridge, MA, AI Memo 183, 1970.

[12] E. C. Hildreth, "Implementation of a theory of edge detection," M.I.T. Artificial Intell. Lab., Cambridge, MA, Rep. AI-TR-579, 1980.

[13] ——, *The Measurement of Visual Motion*. Cambridge, MA: M.I.T. Press, 1983.

[14] B. K. P. Horn, "The Binford-Horn line-finder," M.I.T. Artificial Intell. Lab., Cambridge, MA, AI Memo 285, 1971.

[15] D. G. Luenberger, *Introduction to Linear and Non-Linear Programming*. Reading, MA: Addison-Wesley, 1973.

[16] I. D. G. Macleod, "On finding structure in pictures," in *Picture Language Machines*, S. Kaneff, Ed. New York: Academic, 1970, p. 231.

[17] D. C. Marr, "Early processing of visual information," *Phil. Trans. Roy. Soc. London*, vol. B 275, pp. 483-524, 1976.

[18] D. C. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. London.*, vol. B 207, pp. 187-217, 1980.

[19] D. C. Marr and T. Poggio, "A theory of human stereo vision," *Proc. Roy. Soc. London.*, vol. B 204, pp. 301-328, 1979.

[20] J. E. W. Mayhew and J. P. Frisby, "Psychophysical and computational studies toward a theory of human stereopsis," *Artificial Intell. (Special Issue on Computer Vision)*, vol. 17, 1981.

[21] T. Poggio, H. Voorhees, and A. Yuille, "A regularized solution to edge detection," M.I.T. Artificial Intell. Lab., Cambridge, MA, Rep. AIM-833, 1985.

[22] A. P. Pentland, "Visual inference of shape: Computation from local features," Ph.D. dissertation, Dep. Psychol., Massachusetts Inst. Technol., Cambridge, MA, 1982.

[23] J. M. S. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictorics*, B. Lipkin and A. Rosenfeld, Eds. New York: Academic, 1970, pp. 75-149.

[24] S. O. Rice, "Mathematical analysis of random Noise," *Bell Syst. Tech. J.*, vol. 24, pp. 46-156, 1945.

[25] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Trans. Comput.*, vol. C-20, no. 5, pp. 562-569, 1971.

[26] L. Spacek, "The computation of visual motion," Ph.D. dissertation, Univ. Essex at Colchester, 1984.

[27] K. A. Stevens, "Surface perception from local analysis of texture and contour," M.I.T. Artificial Intell. Lab., Cambridge, MA, Rep. AI-TR-512, 1980.

[28] V. Torre and T. Poggio, "On edge detection," M.I.T. Artificial Intell. Lab., Cambridge, MA, Rep. AIM-768, 1984.
[29] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, MA: M.I.T. Press, 1979.
[30] N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series*. Cambridge, MA: M.I.T. Press, 1949.
[31] A. P. Witkin, "Shape from contour," M.I.T. Artificial Intell. Lab., Cambridge, MA, Rep. AI-TR-589, 1980.

John Canny (S'81–M'82) was born in Adelaide, Australia, in 1958. He received the B.Sc. degree in computer science and the B.E. degree from Adelaide University in 1980 and 1981, respectively, and the S.M. degree from the Massachusetts Institute of Technology, Cambridge, in 1983.

He is with the Artificial Intelligence Laboratory, M.I.T. His research interests include low-level vision, model-based vision, motion planning for robots, and computer algebra.

Mr. Canny is a student member of the Association for Computing Machinery.

COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 30, 316–330 (1985)

Discontinuity Detection for Visual Surface Reconstruction

W. ERIC L. GRIMSON*

Artificial Intelligence Laboratory, M.I.T., Cambridge, Massachusetts 02139

AND

THEO PAVLIDIS

AT & T Bell Laboratories (2C-456), Murray Hill, New Jersey 07974

Received February 22, 1984; accepted February 5, 1985

A method is described for discontinuity detection in pictorial data. It computes at each point a planar approximation of the data and uses the statistics of the differences between the actual values and the approximations for detection of both steps and creases. The use of local statistical properties in the residuals provides a detection method that is sensitive to the local context of the data, and avoids the use of arbitrary thresholds. The subsequent reconstruction, bounded by the detected discontinuities, avoids Gibbs effects and provides reliable surface measurements.

INTRODUCTION

This papers suggests a new approach to the detection of discontinuities in pictorial data: the use of the distribution of the error in local approximations of the data by a smooth surface. (The common practice now is to use the size of an error norm.) While similar approaches have been used before for spline approximations (Powell [10], Ichida, Kiyono, and Yoshimoto [6], this is the first application of the idea in image processing. It is based on the following observation. If the data have been generated by adding noise to a smooth surface, then the errors of approximation of the data by such a surface should exhibit the statistical properties of the noise. Any deviations from such a behavior (i.e., the appearance of systematic errors) indicates that the smoothness assumption is false. A similar reasoning may be used for data that are sampled from a Gibbs distribution (or a Markov field) [3]. The potential functions of the Gibbs distribution could be used to design approximating surfaces.

While the basic approach is quite general we shall use it here in connection with the fitting of smooth surfaces to scattered samples in the presence of noise. Thus the detection of discontinuities is only an intermediate step. Smooth interpolation is a recurrent problem in image analysis, and indeed in other types of signal processing. Recently, a solution to this problem for the case of stereo and motion data has been developed, relying on a surface reconstruction from 2-dimensional data [4, 11]. One of the difficulties of smooth surface reconstruction illuminated by these investigations is that it gives rise to oscillations (Gibbs effects) when applied around discontinuities in the data (see Fig. 1). This is not surprising since such smooth reconstructions make sense only over coherent parts of the signal, and applying a smooth reconstruction across a discontinuity implies that the shape of one surface

*Consultant to AT & T Bell Laboratories.

0734-189X/85 $3.00

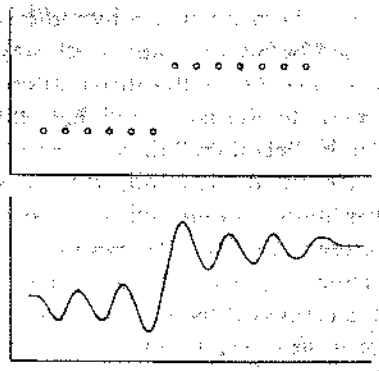


FIG. 1. Fitting a smooth curve (continuous with continuous derivatives) in the data shown at the top results in the oscillating curve shown at the bottom. The practically desirable interpolation is two horizontal lines with a discontinuity.

can influence the shape of a second, distinct surface, an implication which is clearly incorrect. The identification of such discontinuities has been the focus of considerable research in image processing and computer vision in general and we shall review such earlier approaches shortly.

Direct edge detection is a well-developed technique and can be found in any text on image processing. Its major disadvantage is that it is sensitive to noise, like any other method based on differentiation. Prefiltering the data may help in some cases but in general, the resulting smoothing introduces ambiguities in the location of the edges. Region growing or partition building techniques [8, 12] have been proposed as alternatives to edge detection but they also suffer from certain limitations. First they insist on finding coherent regions even if they do not exist. For example, Fig. 2 illustrates a case where $f(x, y)$ is continuous everywhere except along the line L. While finding discontinuities is equivalent to domain segmentation in the case of functions of a single variable, this is not the case for functions of two variables and partition building techniques tend to ignore that fact. Even where segmentation is feasible, it may be difficult when the form of the function in two adjacent regions is similar. In all these cases, partition building tends to produce regions whose shape reflects more the search strategy used than the true shape of regions. For example, segmentation using quad trees produces regions of square shape. (See Fig. 5.14 in [8].)

More recently, a third technique for detecting discontinuities in depth data has been proposed. If the surface reconstruction process is modeled as that of fitting a

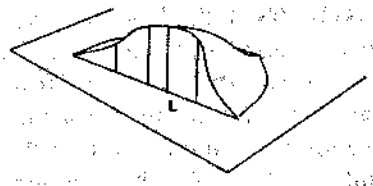


FIG. 2. Illustration of a function which is everywhere continuous except along the line L. One can modify the example slightly to achieve gradient continuity except along L.

thin, elastic plate to a set of known depth constraints [4, 1, 11], then it has been suggested [11] that the discontinuities in the data are associated with places of high tension in the plate. While this method does detect discontinuities, it is not completely error-free because there is no one-to-one correspondence between the two. It is possible to have many locations of high tension for a single discontinuity, because of Gibbs effects, or failure to detect high tension if the data points around a discontinuity are sparse.

Our approach to discontinuity detection is a hypothesis testing technique. While such techniques have been used before in image processing (see, e.g., [5, 2] etc.), our approach is novel because it focuses on the distribution of the error. We will fit locally simple surfaces (planes) to the data and then examine the distribution of residual error. If it appears to be "random," then we accept the hypothesis that there is no discontinuity. If there are systematic trends, then these imply discontinuities of various types. Once a discontinuity is detected, it is not necessary to worry about tracking edges or finding closed regions. The method of [4] computes the surface reconstructions by minimizing a particular functional, where the effect of the neighbors on a given point is expressed by a weight matrix $w(i, j)$ [4, pp. 183–184]. This matrix takes special forms near edges and corners, and such forms may be used near discontinuities as well. We shall say more on that later.

One of the advantages of detecting discontinuities first and fitting surfaces afterwards is that it is unlikely to have any oscillations due to Gibbs effects. Thus we may compute with confidence not only gradients of that surface but also higher order derivatives. We can then use these derivatives to detect changes in the form that correspond to different objects, in a manner similar to that used for 1-dimensional analysis by McClure [7].

METHODS

Let $\mathbf{x} = (x_1, x_2)$ be a point on the plane and $x(\mathbf{x})$ the value of the data there. For example, $z(\mathbf{x})$ could be depth or illumination intensity. Let $N(\mathbf{x}_0)$ be a neighborhood around $\mathbf{x}_0$ consisting of all points within a given distance from $\mathbf{x}_0$. (The exact manner in which $N(\mathbf{x}_0)$ is defined is not essential. For example, we could select the k nearest points to $\mathbf{x}_0$ in each quadrant for some selected value k.) Let S be a family of smooth functions defined over the plane and let $s(\mathbf{x}_0, \mathbf{x})$ be a member of S that is a good approximant to $z(\mathbf{x})$ over $N(\mathbf{x}_0)$. Again, the exact manner in which $s(\mathbf{x}_0, \mathbf{x})$ is selected is not essential.

We define the residual at a point x as

$$e(\mathbf{x}) = z(\mathbf{x}) - s(\mathbf{x}_0, \mathbf{x}). \tag{1}$$

Roughly speaking, $e(\mathbf{x})$ is the difference between the given value at $\mathbf{x}$ and a filtered, or expected, value there. We justify the use of the distribution of $e(\mathbf{x})$ for detection of discontinuities on the following grounds. Suppose that

$$z(\mathbf{x}) = q(\mathbf{x}) + n(\mathbf{x}), \tag{2}$$

where $q(\mathbf{x})$ is a smooth function of $\mathbf{x}$ and $n(\mathbf{x})$ is a zero-mean, high-frequency (relative to the signal $q(\mathbf{x})$) noise distribution. If $q(\mathbf{x})$ is a member of S, then for sufficiently large neighborhoods $N(\mathbf{x}_0)$, the function $s(\mathbf{x}_0, \mathbf{x})$ will be a good estimate of $q(\mathbf{x})$ and the residual $e(\mathbf{x})$ will approximately equal $n(\mathbf{x})$. Then the distribution of $e(\mathbf{x})$ will be determined by the statistics of $n(\mathbf{x})$. However, if $q(\mathbf{x})$ is a function outside S then the residual $e(\mathbf{x})$ will have properties that differ from those of the noise. Algorithms for spline approximation based on this idea have been proposed by Powell [10] and by Ichida *et al.* [6]. We shall discuss soon how these properties relate to discontinuities but first we would like to point out a connection with Bayesian methods for image restoration (e.g., [3]).

These techniques rely on estimates of the probability distribution of the values $z(\mathbf{x}_0)$ given the values of $N(\mathbf{x}_0)$. However, that density can be evaluated only when all the points in the neighborhood have values generated by the same stochastic process. Points where this is not true must be treated in a different way. Thus one is faced with two problems: first detecting such points, and second estimating a distribution for them. The proposed method can be seen as a test for checking the validity of the assumption that the points in $N(\mathbf{x}_0)$ have been generated by the same process.

In our present development we insist that $s(\mathbf{x}_0, \mathbf{x})$ be a plane. This is one of the simplest possible interpolants and its main justification is *a posteriori*: we shall show

that it allows an easy detection of discontinuities without breaking quadratic surfaces. We feel that it is important not to segment quadratic surfaces for the following reasons. If the data are from a depth map, then surfaces correspond directly to object shapes and spherical or cylindrical objects are far too common to be ignored. If the data represent light intensities, then we know that Lambertian reflection produces intensities that are not constant but vary in ways that can be approximated by quadratics. The selection of a plane for the approximant implies that if $q(\mathbf{x})$ is also a plane, $e(\mathbf{x})$ will follow the noise distribution. We consider now several special cases.

(1) $q(\mathbf{x})$ is a convex (concave) function. Then, if there is no noise, $e(\mathbf{x})$ will always be negative (positive). Moreover, not only will the sign of $e(\mathbf{x})$ not change, but its magnitude should vary little among adjacent points. If $q(\mathbf{x})$ is a parabolic surface and if the error norm to be minimized is the maximum error (uniform approximation), then $e(\mathbf{x})$ is a constant (see Appendix). The presence of noise will modify these conclusions only slightly (see below).

(2) $q(\mathbf{x})$ is a crease, i.e., an intersection of two planes. If a point $\mathbf{x}$ is far enough from the (projection of the) intersection, then all of the data points corresponding to $N(\mathbf{x})$ will be within a single plane, and $e(\mathbf{x})$ will approximately equal $n(\mathbf{x})$. If $N(\mathbf{x})$ includes the intersection of the planes, then $e(\mathbf{x})$ will, for the most part, have a fixed sign, but not fixed magnitude. It can be shown that for a neighborhood that is symmetric with respect to the projection of the intersection of the planes, the residual will be proportional to the difference in the slopes of the planes along the direction perpendicular to their intersection (see Appendix).

(3) $q(\mathbf{x})$ is a step function. Then $e(\mathbf{x})$ will be a pulse with amplitude one quarter that of the step and changing from a positive to a negative value in the case of a step increasing to the right and vice versa for a step decreasing to the right (see Appendix).

(4) $q(\mathbf{x})$ is a filtered step function. For example, $q(\mathbf{x}) = 0$ is x_1 is less than $-\delta$ (δ is a given small number), $q(\mathbf{x}) = 1$ if x_1 is greater than δ, and $q(\mathbf{x})$ varies linearly over $[-\delta, \delta]$. Then $q(\mathbf{x})$ can be considered as a superposition of neighboring creases and $e(\mathbf{x})$ will again be a pulse, but triangular, rather than rectangular, as in the step case.

These special cases reveal that a data discontinuity will manifest itself as a change in the sign of the residuals (a zero-crossing) with the values on either side having size proportional to the step size. A crease will appear as a high value surrounded by linearly decreasing values of the same sign. Thus all we need is an algorithm for detecting extrema in the residuals. If an extremum is isolated, then we decide in favor of a crease. If it is paired with another extremum of opposite sign, then we decide in favor of a discontinuity. This allows a smooth degradation from step discontinuities into creases as the discontinuity is stretched out.

How can we distinguish "true" steps from those due to noise? If the variance of the noise is σ and the distribution is Gaussian, then the amplitude will not exceed 2σ with probability better than 95%. Since steps are attenuated by a factor of 4 (see Appendix), we can detect all steps whose size exceeds 8σ as being distinct from noise. Since the noise statistics may not be known a priori, we can estimate them from the residuals obtaining an estimate $\hat{\sigma}$ for the variance σ. If we select steps where the size exceeds $2\hat{\sigma}$, then we can assume that they are "true" steps, since $\hat{\sigma}$ will be an overestimation of σ because of the presence of systematic errors in the sample.

EXAMPLES

We demonstrate the method on a series of examples, and indicate its limiting behavior under a variety of circumstances. In all the cases illustrated, the surface reconstruction technique used was a least-square fit to the k nearest neighbors on either side, although this is not critical.

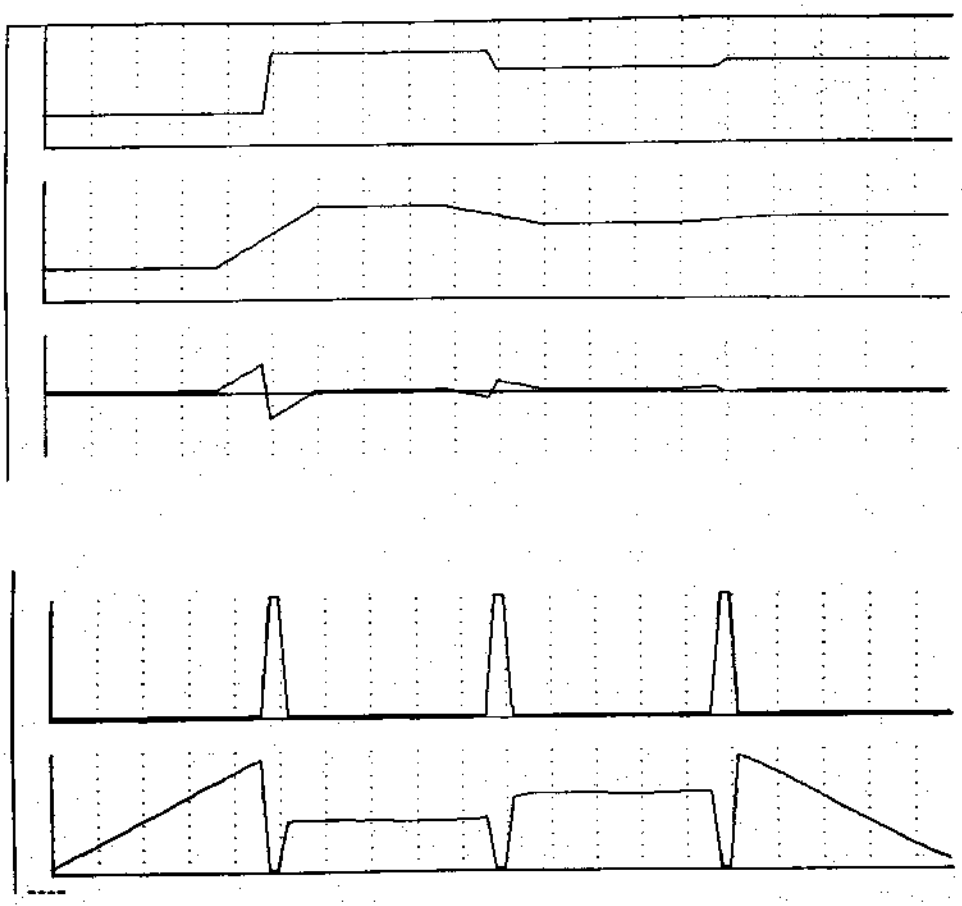


FIG. 3. Traces of plots of function (top), smooth approximation (second), residual (third), significant residual (fourth), and approximation after discontinuity detection (last).

Figure 3 shows the basic steps of the method. The first graph indicates the initial data, in this case a series of ideal, parallel planes. In the second graph, the filtered surface obtained by a point-wise, least-squares fit is shown. The third graph indicates the residuals of the surface under this filtering, and the fourth graph marks the points at which a step discontinuity is detected, based on the local distributions of the residuals. Finally, the fifth graph shows the final, segmented, least-squares reconstruction, in which the curve fitting is not propagated across marked discontinuities. The plotted approximation does not attempt to interpolate between different parts of the function. Whether this is desirable or not depends on the application and it can be easily accomplished.

In Fig. 4, zero-mean, high frequency noise has been added to the initial data, which consisted of four parallel planes, separated by increasingly larger step discontinuities. When the step size is large compared to the noise, then discontinuities are still detected but smaller steps are not. This is to be expected, since the magnitude of the noise has essentially reached the size of the step discontinuity. Thus, the method shows a graceful degradation in the presence of increasing corruptive noise.

An example of a quadratic surface with noise added to the data, is shown in Fig. 5. Note the form of the residuals. It can be seen that in this case, the connections between the cylindrical surface and the surrounding planes are detected as step discontinuities, since both the positive and negative extrema in the residuals around those points are detected as significant. As the gradient of the cylindrical surface is decreased, however, it is clear that the negative portion of the residual would eventually drop below detection range, leaving only positive extrema in the residuals. Hence, the connection between the two surfaces would gracefully degrade from a step discontinuity into a crease, as would be expected.

Figure 6 shows a series of arbitrary planes with noise in the data. Note that while the initial smoothed surface has extreme deviations from the data around the

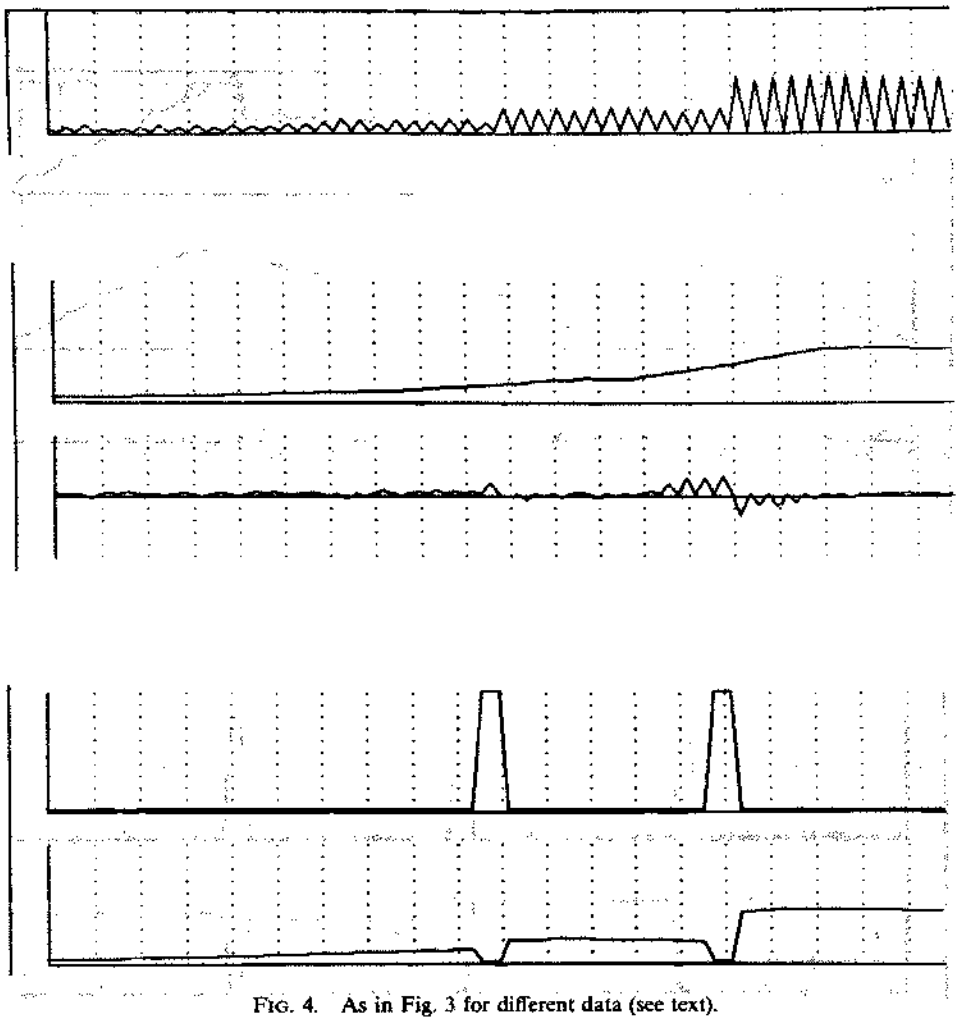


FIG. 4. As in Fig. 3 for different data (see text).

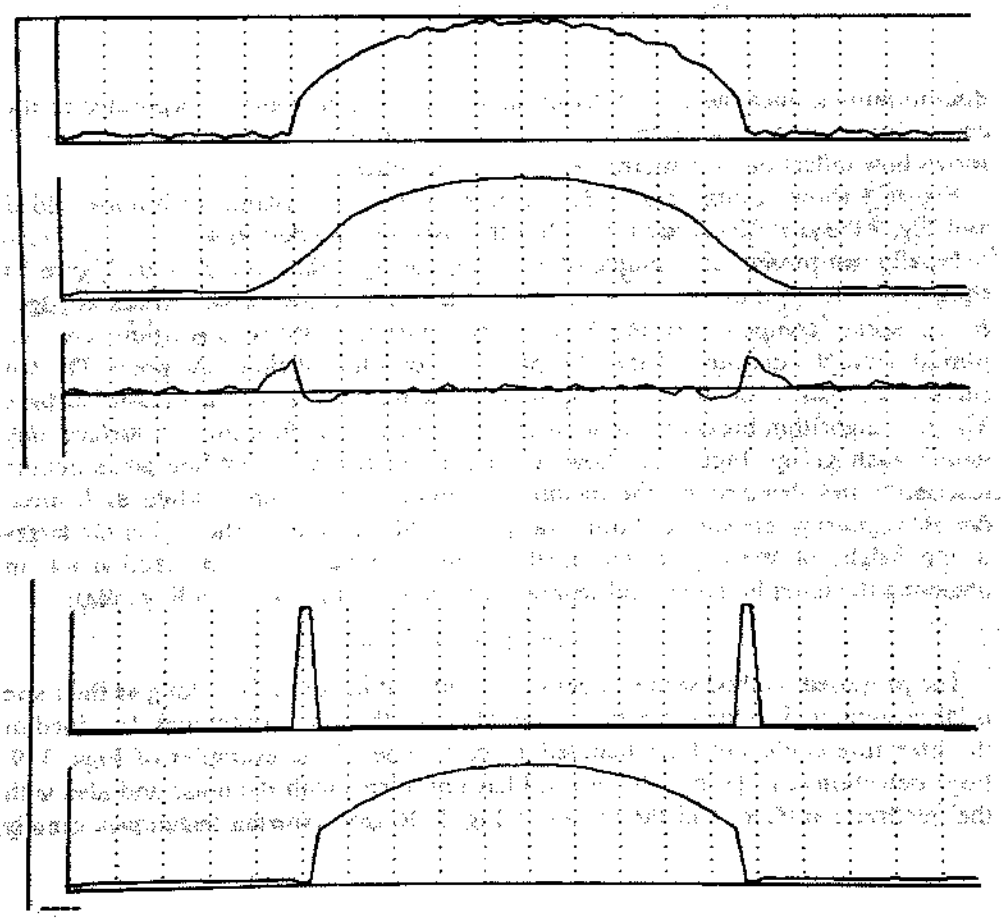


FIG. 5. As in Fig. 3 for different data (see text).

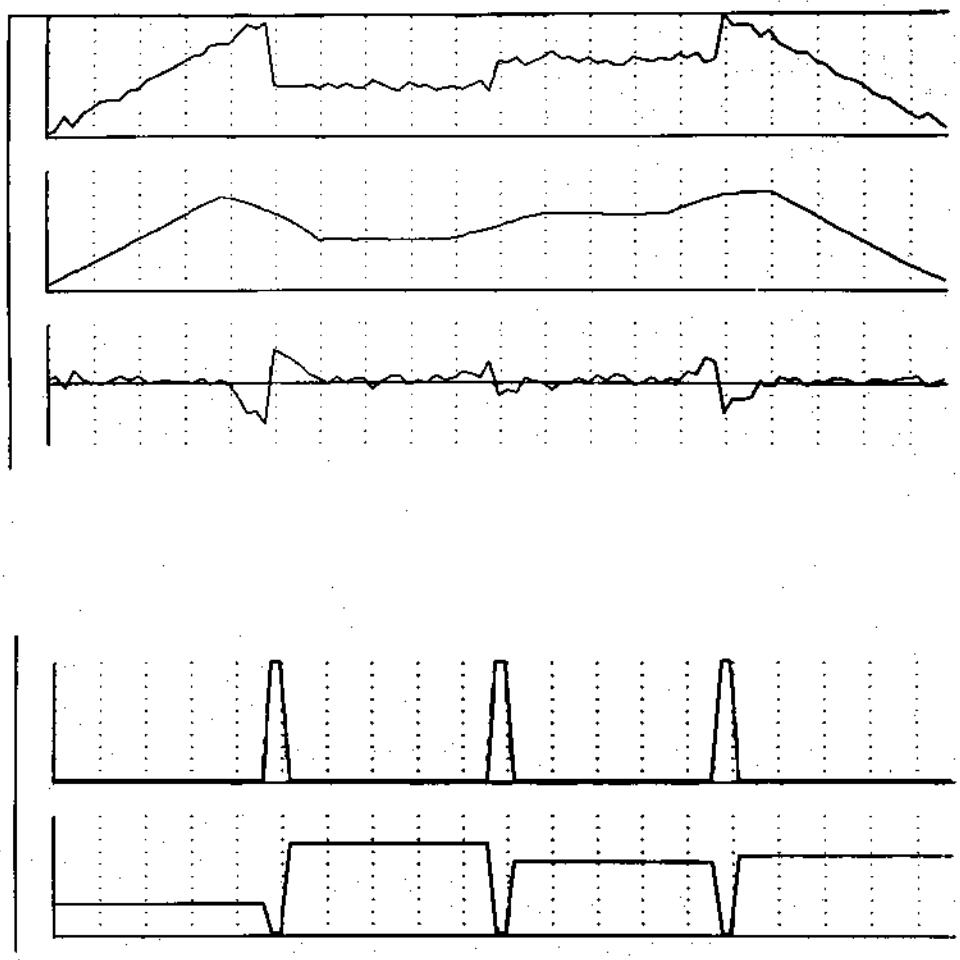


FIG. 6. As in Fig. 3 for different data (see text).

discontinuities, once the correct discontinuities are detected and incorporated in the curve fitting process, a much more accurate reconstruction is obtained. Figure 7 shows how inflections are interpreted as discontinuities.

Figure 8 shows 2-dimensional data generated from four planes with noise added and Fig. 9 the smooth surface fitted after the discontinuity detection.

Finally, we present an example of applying the algorithm to real data. Figure 10 shows left and right eye views of a scene, while in Fig. 11, the depth values at edges in the scene, computed by the Marr–Poggio–Grimson stereo algorithm, are displayed. Here the darkness of the edge points denotes the height of the point. The old surface reconstruction algorithm [4] fits directly these data with a smooth surface. The new algorithm breaks them in groups and attempts to fit a smooth surface only within each group. Figure 12 shows the results in two views. White areas denote discontinuities detected by the method described in this paper, while dark areas denote regions where surface fitting was performed. The darker the region the larger is the height of the object. The method seems to have done a credible job in detecting the taller buildings and separating them from their surrounding areas.

CONCLUSIONS

The proposed method seems to detect discontinuities correctly as long as their size is large compared to the noise amplitude. Many of the other techniques described in the literature could not have handled at least some of the examples of Figs. 3–9. Edge detection (e.g., [8, pp. 79–86]) will have problems with the noise and also with the quadratic surfaces and the planes of Fig. 5. Region growing techniques usually

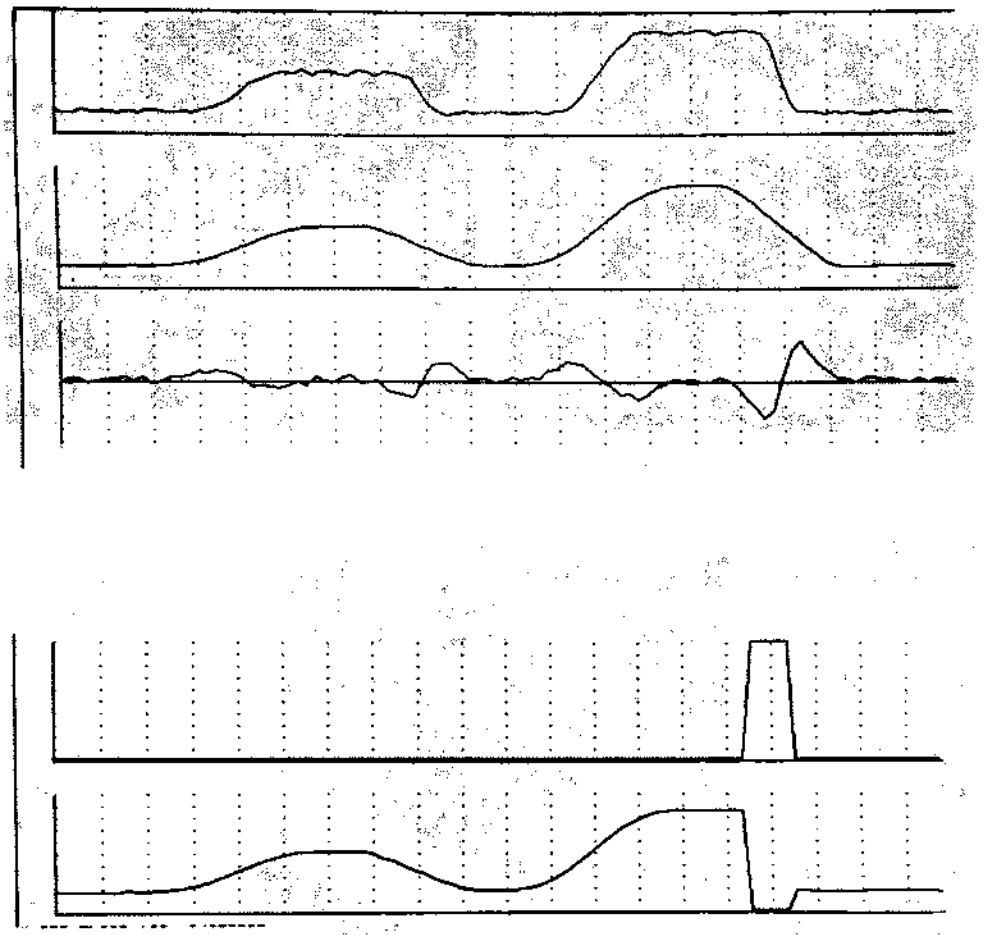


FIG. 7. As in Fig. 3 for different data (see text).

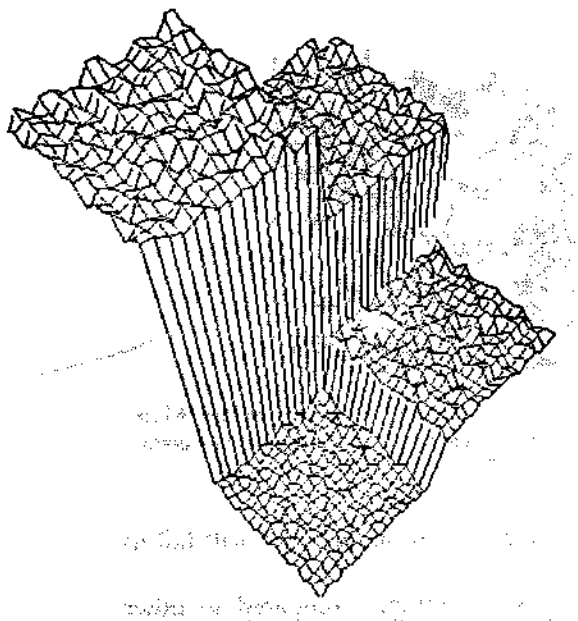


FIG. 8. Three dimensional input.

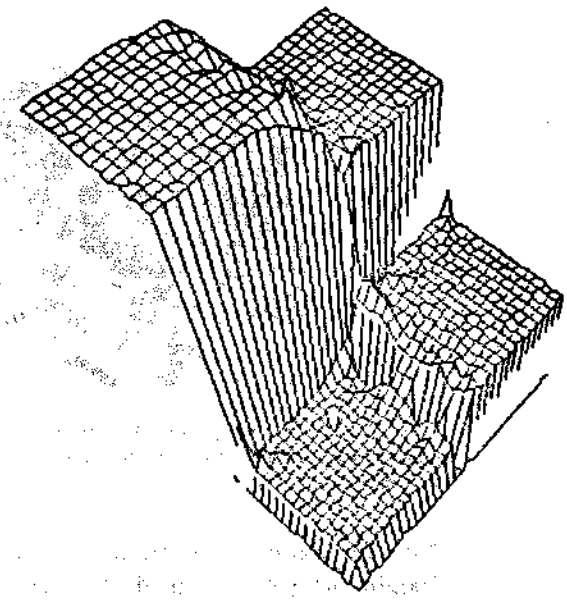


FIG. 9. Three dimensional approximation after discontinuity detection.

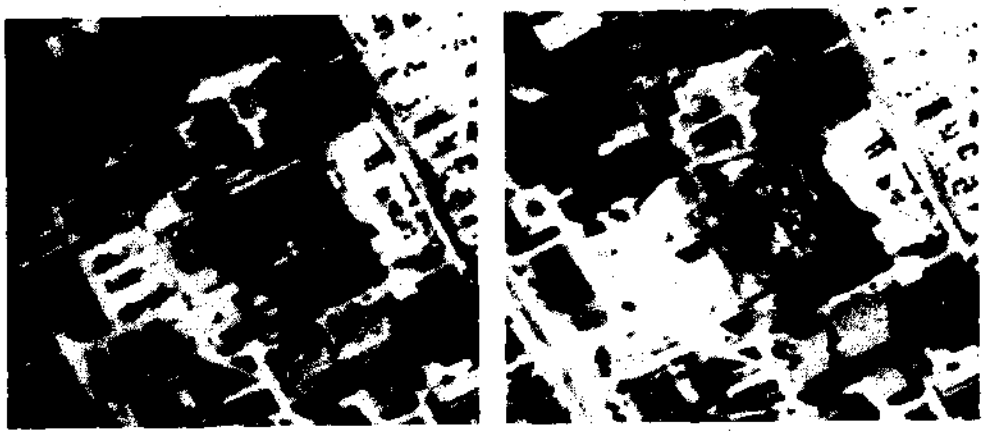


FIG. 10. Left and right eye views of a scene.

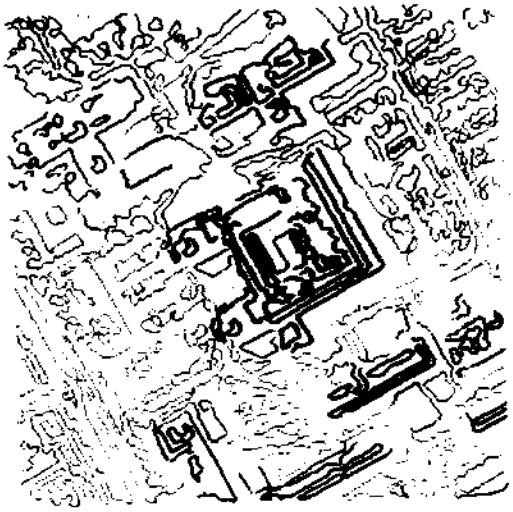


FIG. 11. Depth values of edge points in the scene of Fig. 10 detected by the Marr–Poggio–Grimson stereo algorithm. The darker a point, the greater is its height.

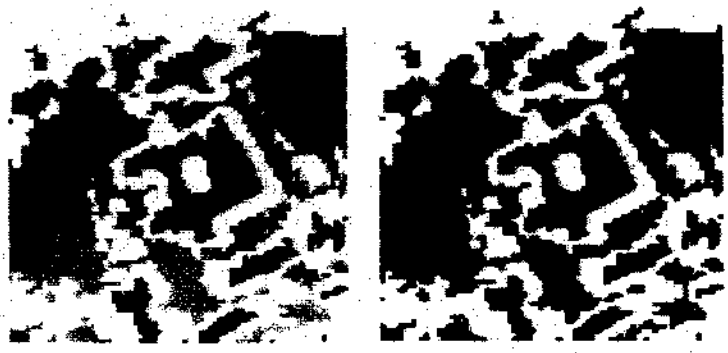


FIG. 12. Left and right eye views of the scene of Fig. 10 after processing by the new algorithm. White areas denote discontinuities and darker areas regions where surfaces were fitted. Again, the darker a point, the greater is its height.

look for piecewise constant or piecewise planar approximation and they will fail to handle properly the quadratic surfaces.

The method also has the following operational advantages compared to other techniques.

(1) There is no need to set arbitrary thresholds when deciding about discontinuities. The only threshold used is expressed in terms of the estimated variance of the noise.

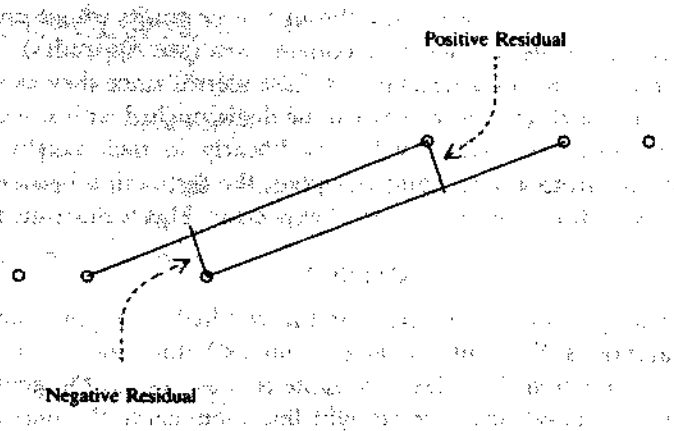


FIG. 13. The sign of the residuals is independent of the distance among the data points.

(2) It can detect discontinuities regardless of spatial resolution. If the data points are very sparse, then we can obtain reliable estimates by taking the k nearest to x (see Fig. 13.) Also no false alarms are introduced because of steep quadratic curves since "steepness" does not affect the sign of the residual.

(3) The only Gibbs effects that may occur are near the noise level. Therefore we have reliable derivatives in the computed surface.

(4) There is no need to be concerned about closing contours. The final surface fitting is done by using the method of [4], which relies on a 12-point neighborhood as shown in Fig. 14.

If the sign of the residual at C and S differs (and the magnitude exceeds two standard deviations), but the sign at N, W, C, and E is the same, then the point C is treated as an edge point. If the sign at C, N, and E is the same and opposite to the sign at W and S, then C is treated as a corner point. Other configurations can be treated in a similar fashion by modifying the equation of p. 182 of [4] and deriving the correct template.

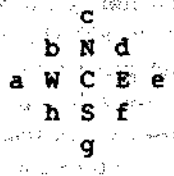


FIG. 14. Neighborhood arrangement used for surface fitting in [4].

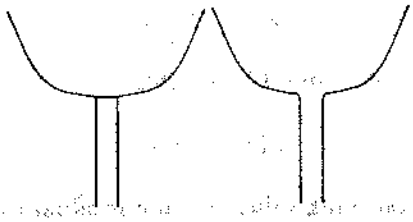


FIG. 15. A crease (left) and an inflection (right) that appear similar.

The method is computationally expensive because of the need to compute a planar approximation at each point, but since we do need an optimal approximation the cost is not as high as it might seem. For example, in one dimension we may use the interpolant between the extreme points of an interval as the approximating line. (See [9, p. 286] for a discussion of the quality of such an approximation.) In two

dimensions we may use the plane passing through three points whose projections on the plane are vertices of the triangle of maximum area (see Appendix).

Inflection points are one possible source of false alarms since they cause a change in the sign of the residual. However, they can be distinguished with some effort from discontinuities because the residual will vary linearly in their neighborhood (see Appendix). Anyway, from a perceptual viewpoint the distinction between inflection points and creases or discontinuities is not always clear. This is illustrated by Fig. 15.

APPENDIX

It is possible to give exact formulas for the residual at a point under certain simplifying assumptions. We consider the continuous 1-dimensional case and we let $f(t)$ to be a given function that plays the same role as $z(\mathbf{x})$ in the general case. If $f(t)$ is convex (or concave), then the straight line minimizing the maximum pointwise error over the interval $[-\delta, \delta]$ is the endpoint interpolant shifted halfway to the distance towards a point where the tangent to $f(t)$ is parallel to the interpolant [9]. If $f(t)$ is symmetric around $t = 0$, then a simple calculation shows that the residual will be exactly

$$e(t) = \frac{1}{2}\left[\frac{f(t-\delta) + f(t+\delta)}{2} - f(t)\right]. \tag{A.1}$$

One could make a good case that the quantity in brackets is always close to the residual, although the factor $\frac{1}{2}$ may be too big. Certainly the sign will be valid as long as there are no inflection points. One may also decide that the interpolant is always a good approximant [9] or that Eq. (A.1) is a good definition of the residual, anyway. Clearly $e(t)$ is zero if $f(t)$ is a linear function.

There is no similar simple formula in two dimensions but the following argument can be made. Consider four points in space. They form a tetrahedron of volume, say, V. Let us select the plane formed by three of them as the approximating plane and let A be the area of the triangle formed by the three points. Then the distance of the fourth point from the plane is $6V/A$. Thus if we insist that the approximating plane be an interpolating plane, then the best selection is the one defined by the three points that form the triangle of maximum area. Shifting the plane halfway towards the fourth point reduces the error of approximation to $3V/A$. If we have more than four points, then we may select from amongst them the triplet that forms the triangle of maximum area. If the surface formed by the points is not far from the horizontal then we may select a triplet whose projections on the plane form the triangle of greatest area. With these heuristics we will find a residual which will be given by a formula similar to Eq. (A.1). However, instead of the midvalue of the interval endpoints we will have a linear combination of the three points defining the maximum area triangle.

If $f(t)$ is the parabola

$$at^2 + bt + c \tag{A.2}$$

then a substitution of Eq. (A.2) into (A.1) yields

$$e(t) = \tfrac{1}{2}a\delta^2 \tag{A.3}$$

Similarly we obtain the following values for the residual for various forms of $f(t)$.

$$\text{Cubic: } f(t) = at^3 + b^2 + ct + d \tag{A.4a}$$

$$e(t) = \frac{\delta^2}{2}(3at + b). \tag{A.4b}$$

We see that for a cubic the residual varies linearly.

Piecewise Linear: $f(t) = at$ for $t \le 0$, $\quad f(t) = bt$ for $t > 0$. (A.5)

$$e(t) = 0, \qquad t \text{ not in } [-\delta, \delta], \tag{A.6a}$$

$$e(0) = \tfrac{1}{4}(a - b)\delta \quad \text{varying linearly elsewhere.} \tag{A.6b}$$

The last case represents a crease and we see that the residual at along the crease is proportional to the change in slope.

If $f(t)$ is a step function at 0 of size S, then we find

$$e(t) = 0 \, t \text{ not in } [-\delta, \delta], \qquad e(t) = \frac{S}{4} \, t \le 0, \qquad e(t) = -\frac{S}{4} \, t > 0. \tag{A.7}$$

The difference between Eqs. (A.4b) and (A.6) can be used to distinguish steps from inflection points.

REFERENCES

1. M. Brady and B. K. P. Horn, Rotationally symmetric operators for surface interpolation, *Comput. Vision Graphics Image Process.* **22** (1983), 70–94.
2. P. C. Chen and T. Pavlidis, Image segmentation as an estimation problem, *Comput. Graphics Image Process.* **12** (1980), 153–172; *Image Modeling* (A. Rosenfeld, Ed.), pp. 9–28, Academic Press, New York, 1981.
3. S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6** (1984), 721–741.
4. W. E. L. Grimson, *From Images to Surfaces: A Computational Study of the Human Early Visual System*, MIT Press, Cambridge, Mass., 1981.
5. R. M. Haralick, Edge and region analysis for digital image data, *Comput. Graphics Image Process.* **12** (1980), 60–73.
6. K. Ichida, T. Kiyono, and F. Yoshimoto, Curve fitting by a one-pass method with a piecewise cubic polynomial, *ACM TOMS* **3** (1977), 164–174.
7. D. E. McClure, Computation of approximately optimal compressed representations of discretized plane curves, in *Proc. IEEE Conf. Pattern Recognition Image Process.* Troy, N.Y., 1977, pp. 175–182.
8. T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, New York/Berlin, 1977.
9. T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, MD., 1982.
10. M. J. D. Powell, Curve fitting by splines in one variable, in *Numerical Approximation to Functions and Data* (J. G. Hayes, Ed.), pp. 65–83, Athlone, London, 1970.
11. D. Terzopoulos, Multilevel computational processes for visual surface reconstruction, *Computer Vision Graphics Image Process.* **24** (1983), 52–96.
12. S. W. Zucker, Region growing: Childhood and adolescence, *Computer Graphics Image Process.* **5** (1976), 382–399.

Reprinted from *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 10, Number 2, March 1988, pages 167-192.

Segmentation Through Variable-Order Surface Fitting

PAUL J. BESL, MEMBER, IEEE, AND RAMESH C. JAIN, SENIOR MEMBER, IEEE

Abstract—**Computer vision systems attempt to recover useful information about the three-dimensional world from huge image arrays of sensed values. Since direct interpretation of large amounts of raw data by computer is difficult, it is often convenient to partition (segment) image arrays into low-level entities (groups of pixels with similar properties) that can be compared to higher-level entities derived from representations of world knowledge. Solving the segmentation problem requires a mechanism for partitioning the image array into low-level entities based on a model of the underlying image structure. Using a piecewise-smooth surface model for image data that possesses surface coherence properties, we have developed an algorithm that simultaneously segments a large class of images into regions of arbitrary shape and approximates image data with bivariate functions so that it is possible to compute a complete, noiseless image reconstruction based on the extracted functions and regions. Surface curvature sign labeling provides an initial coarse image segmentation, which is refined by an iterative region growing method based on variable-order surface fitting. Experimental results show the algorithm's performance on six range images and three intensity images.**

Index Terms—**Image segmentation, range images, surface fitting.**

I. Introduction

COMPUTER vision systems attempt to recover useful information about the three-dimensional (3-D) world from huge image arrays of sensed values. Since direct interpretation of large amounts of raw data by computer is difficult, it is often convenient to partition (segment) image arrays into low-level entities (groups of pixels with particular properties) that can be compared to higher-level entities derived from representations of world knowledge. Solving the segmentation problem requires a mechanism for partitioning the image array into useful entities based on a model of the underlying image structure.

In most easily interpretable images, almost all pixel values are statistically and geometrically correlated with neighboring pixel values. This pixel-to-pixel correlation, or *spatial coherence*, in images arises from the spatial coherence of the physical surfaces being imaged. In range images, where each sensed value measures the distance to physical surfaces from a known reference surface, the pixel values collectively exhibit the same spatial coherence properties as the actual physical surfaces they represent. This has motivated us to explore the possibilities of a surface-based image segmentation algorithm that uses the spatial coherence (*surface coherence*) of the data to organize pixels into meaningful groups for later visual processes.

Many computer vision algorithms are based on inflexible, unnecessarily restricting assumptions about the world and the underlying structure of the sensed image data. The following assumptions are common: 1) all physical objects of interest are polyhedral, quadric, swept (as in generalized cylinders), convex, or combinations thereof; 2) all physical surfaces are planar, quadric, swept, or convex; 3) all image regions are rectangular or regularly shaped and are approximately constant in brightness; and 4) all image edges are linear or circular. The extensive research based on these assumptions solves many important application problems, but these assumptions are very limiting when analyzing scenes containing real-world objects with free-form, sculptured surfaces. Therefore, we have developed an image segmentation algorithm based only on the assumption that the image data exhibits *surface coherence* in the sense that the image data may be interpreted as noisy samples of a piecewise-smooth surface function. A preliminary grouping of pixels is based on the sign of mean and Gaussian surface curvature. This initial, coarse segmentation is refined by an iterative region growing procedure based on variable-order bivariate surface fitting. The order of the surface shape hypotheses is automatically controlled by *fitting surfaces* to the image data and *testing the surface fits* by 1) checking the spatial distribution of the signs of residual fitting errors (the regions test) and 2) comparing the mean square residual error of the fit to a threshold proportional to an estimate of the image noise variance. In this iterative process, images are not only segmented into regions of arbitrary shape, but the image data in those regions is also approximated with flexible bivariate functions such that it is possible to compute a complete, noiseless image reconstruction based on the extracted functions and regions. We believe that an explicit image description based on flexibly shaped approximating functions defined over arbitrary connected image regions can be useful in many computer vision applications, but will be critically important to object reconstruction and object recognition algorithms based on range imaging sensors when object volumes are bounded by free-form, smooth surfaces.

Although this segmentation algorithm may be most useful for range images, it is capable of segmenting any type

Manuscript received January 29, 1986; revised March 12, 1987. Recommended for acceptance by W. E. L. Grimson. This work was supported by IBM Corporation, Kingston, NY.

P. J. Besl was with the Computer Vision Research Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109. He is now with the Department of Computer Science, General Motors Research Laboratories, Warren, MI 48090.

R. C. Jain is with the Computer Vision Research Laboratory, Department of Electrical Engineering and Computer Science, the University of Michigan, Ann Arbor, MI 48109.

IEEE Log Number 8718602.

0-8186-9102-6/91 $1.00 © 1988 IEEE

of image that can be adequately represented as a noisy, sampled version of a piecewise-smooth graph surface. Therefore, we first include a brief discussion of the relationship between general computer vision and range image understanding. Since other segmentation methods in the literature involve several ideas closely related to those presented here, we also include a brief discussion of previous work in intensity image segmentation, followed by a survey of more recent work in range image analysis, in order to clarify the differences of the surface-based segmentation algorithm. Mathematical preliminaries are then presented to precisely define the problem we are attempting to solve, followed by a qualitative description of a method for general smooth surface decomposition. Several key ideas behind the algorithm philosophy are described next. Then the entire algorithm is outlined to introduce the role of the individual algorithm elements, followed by a detailed explanation of each element. Experimental results show the algorithm's excellent performance on a variety of six range and three intensity images from a database of successful test results on over forty images. We conclude with comments on future improvements and applications to other types of multidimensional image data.

A. Vision and Range

Most past computer vision research has been concerned with extracting useful information from one or more intensity images of a scene. The desired "useful" information has often been depth or range information. Indeed, the dominant images-to-surfaces vision paradigm [73], [44] dictates that various visual cues can be used to infer the distance of many scene points from sensed light intensity values as the human visual system does. Several methods for obtaining range (shape) from intensity images based on various visual cues are summarized below.

When the sampled values in an image array represent light intensity at each point, knowledge of the intensity image formation process and an appropriate set of constraints can be used to recover the shape of the physical 3-D surfaces represented by the data. Vision researchers have developed many techniques (see survey [18]) for obtaining 2.5-D descriptions (registered range images) of intensity images that indicate the sensor-to-physical-surface distance at many points in a scene: shape from shading [58], shape from texture [103], shape from contour [66], shape from binocular stereo [43], shape from photometric stereo [105], [25], shape from motion [101], [63], shape from shadows [67].

The above are predominantly *passive* approaches for obtaining range information in the sense that energy is not projected into the environment. Many *active* approaches for obtaining range images have also been developed [64] including amplitude-modulated laser radar [108]; frequency-modulated laser radar [7]; time-of-flight laser radar [71]; structured light with lines [93], grids [46], and coded binary patterns [61]; intensity ratio [21]; moire interferometry [86]; and focussing methods [68]. But once a range image has been acquired for a given scene by any of the above methods, the extraction of useful information still requires processing a huge array of values where each value represents the distance to a physical surface from a known reference surface. Hence, the ability to obtain range at each pixel in an image does not in itself solve computer vision problems. Range images provide sampled geometric information in an *explicit form* rather than in an *implicit form* dependent on surface reflectance and illumination. The data must still be organized into a more structured form for interpretation purposes.

Witkin and Tenenbaum [104] have argued that perceptual organization mechanisms exist in the early stages of human visual processing that are independent of the high-level knowledge necessary for correct image interpretation and are independent of the image formation process. That is, people can visually segment image regions into meaningful entities even when they know nothing about the entities or the image formation process. Consider the fact that people with no knowledge of or experience with the formation of images from electron microscopes, X-ray imagers, ultrasonic sensors, and imaging radars can often partition images into important regions that are meaningful to experts in the respective fields. Therefore, it should be possible to group pixels in many types of images using only relatively low-level information. However, it is not at all clear how these general-purpose low-level grouping mechanisms operate.

We believe that perception of surfaces is a low-level grouping operation that plays a fundamental role in many image understanding tasks. Therefore, a segmentation algorithm that groups pixels based on a surface interpretation should be valuable to many applications. For example, explicit surface approximations over range image regions is directly useful for surface inspection, assembly verification, automatic shape acquisition, and autonomous navigation. If early vision processes focus on segmenting range images (however they are acquired) into surfaces defined over image regions, we believe that it will eventually be possible to achieve robust recognition of arbitrary 3-D objects by matching *perceived surface descriptions* with known object models. Although many matching approaches are based on lower dimensional features, such as points (i.e., object vertices) and edges (i.e., occluding edges and separating boundaries between surfaces), we believe that matching based on surface shape holds the most promise for general-purpose vision because surface matching would not be hindered by occlusion of individual point or edge features. Moreover, our experimental results show that surface-based segmentation is also promising for other types of images, such as intensity images, whenever the image data exhibits surface coherence properties.

B. Intensity Image Segmentation

A problem with many computer vision techniques is the assumption that there is only one physical surface or object represented in an image. When many surfaces of many

objects are present, it is often necessary to organize pixels into connected groups or image regions that correspond to individual objects or surfaces, and then apply higher-level algorithms to the isolated image regions. The fundamental, complementary issues in organizing image pixels into regions are similarity (uniformity) and difference (contrast). Given the sensed values at two image pixels and their neighbors, the computer must answer the question: "does this pixel possess enough of the same properties as that pixel to say that these two pixels are similar?"

Segmentation of digital images has been an active area of research for many years (see surveys [47], [31], [37], [65], [90], [88]). Many popular segmentation techniques use histogram-based thresholding or template matching, but these methods provide little information when the image data does not conform to the restrictive image model assumptions. Edge detection techniques (see survey [29]) attempt to define regions by locating pixels that lie on the boundaries between regions using difference measures on neighboring pixels (e.g., image gradient magnitude). Region growing techniques (see survey [107]) attempt to group pixels into connected regions based on similarity measures, such as approximate equality (e.g., [20]). Edge-detection and region-growing can be data-driven operations based on generic notions of difference and similarity that make no commitment to the set of possible image interpretations, or they can be model-driven operations based on application-specific object models and domain knowledge. Model-driven techniques can reduce computational requirements by incorporating high-level knowledge about the scenes represented in images to restrict the search for valid interpretations [41].

A commonly used definition of image segmentation [57] states that if I is the set of all image pixels and $P(\,)$ is a *uniformity precidate* defined on groups of connected pixels, a segmentation of I is a partitioning set of connected subsets or image regions $\{R_1, \cdots, R_N\}$ such that

$$\bigcup_{l=1}^{N} R_l = I \quad \text{where} \quad R_l \cap R_m = \phi \quad \forall l \neq m, \qquad (1)$$

the uniformity predicate $P(R_l) = \text{True}$ for all regions, and

$$P(R_l \cup R_m) = \text{False} \qquad (2)$$

whenever R_l is adjacent to R_m. Different segmentation algorithms may be viewed as implementations of different uniformity predicates. Uniformity predicates may be classified according to knowledge requirements [65]: signal-level methods are based purely on the numbers in a digital image, physical-level methods include knowledge about image formation, and semantic-level methods include even more knowledge about the type of scenes being viewed. The surface-based segmentation algorithm in this paper is a signal-level method where the uniformity predicate on groups of pixels is true if almost all the pixel data in a region can be represented well by an approximating (surface) function.

Functional approximation ideas have been used on intensity images in the past to define uniformity measures for region-growing segmentation at the signal-level. Pavlidis [80] developed a region-growing segmentation approach based on a piecewise-linear scanline function approximation. Scanline intervals with similar slopes were merged to define regions. The uniformity predicate requires pixels in the same region to be approximated by straight lines with similar slopes. Haralick and Watson [48] proved the convergence of the facet iteration algorithm for flat (constant), sloped (planar), and quadratic polynomial facets (local surfaces) defined over preselected image window sizes. This algorithm was intended more for noise removal than segmentation, but may be considered as a segmentation algorithm where the image segments are the resulting small facets. A physical surface in an image is typically represented by many image facets. The uniformity predicate in this case requires pixels are well approximated by the facet surfaces. The window operator size for the surface fits, which limits the facet size, and the surface type are preselected parameters independent of the data. Pong *et al.* [82] have obtained good results with a similar algorithm based on property vectors of facets rather than the facet surface fits.

Functional approximation ideas are also used to derive window coefficients [85] for edge detection approaches to segmentation. In most edge-based techniques, pixels are simply labeled as edge or non-edge, and an edge-linking step is required to create refined region descriptions. A good example of a more complete pixel labeling scheme based on local surface function (facet) approximations is the topographic primal sketch [49]. In this approach, the output consists of 1) step edge, ridge, and valley lines, 2) peak, pit, saddle, and flat points, and 3) planar slopes, convex, concave, and saddle-shaped regions. The uniformity predicate in this case groups pixels with the same topographic label. This method is purely local however and does not prescribe the integration of global similarity information. The surface type labeling used in the surface-based segmentation algorithm also suffers from the same problem, but global information is effectively integrated by the iterative region growing algorithm.

C. Range Image Segmentation

Region growing based on function approximation ideas are used commonly in range image analysis (see survey [9]). The uniformity predicate in the work listed below requires that region pixels are well approximated by planar or quadric surfaces. Shirai and Suwa [94], Milgrim and Bjorklund [75], Henderson and Bhanu [53], Henderson [52], Bhanu [12], and Boyter [17] segment range images into fitted planar surfaces extracted via region growing. Other work has been geared toward detecting cylinders in range data [1], [77], [83], [15], [69]. Hebert and Ponce [51] segmented planes, cylinders, and cones from range data. Sethi and Jayaramamurthy [92] handled spheres and ellipsoids in addition to planes, cylinders, and cones. Oshima and Shirai [79] used planes, spheres, cylinders,

and cones. Dane [27] and Faugeras *et al.* [34] at INRIA allow for region growing based on planar or general quadric surface primitives. The above do not directly address other types of surfaces except that the INRIA [33] and Henderson/Bhanu approaches have worked with arbitrary curved surfaces represented by many-faceted polyhedral approximations. Many of these methods obtain an initial segmentation of small primitive regions and then iteratively merge the small primitives until all merges (allowed by a smoothness or approximation error constraint) have taken place. The RANSAC method of Bolles and Fischler [15] has used iterative model fitting directly on the data based on randomly selected initial points (seeds). Our approach also works directly on the data, but seed regions are extracted deterministically and the model itself may change as required by the data.

Concepts and techniques from differential geometry have been useful in describing the shape of arbitrary smooth surfaces arising in range images [19], [74], [96], [62], [36], [10], [102], [32]. (This approach has also been applied to intensity image description [72], [81], [11].) For segmentation based on differential geometric quantities, such as lines of curvature or surface curvature, the uniformity predicate requires region pixels to possess similar geometric properties. As the name implies, the *differential* geometry of surfaces analyzes the *local differences* of surface points. Although *global similarities* in surface structure are also analyzed, most theorems in differential geometry address only global topological similarities, such as the one-hole equivalence of a doughnut and a coffee cup with a handle. Global shape similarity theorems do exist for the surfaces of convex objects, and they have been successfully utilized in extended Gaussian image (EGI) convex surface shape matching schemes [55]. Difficulties arise when local descriptors are used to identify the shape and global similarities of arbitrary nonconvex object surfaces from arbitrary-viewpoint range-image projections. The mathematics of differential geometry gives little guidance for an integrated global shape description or for computational matching methods in this general case. Brady *et al.* [19] extract and analyze a dense set of integrated 3-D lines of curvature across entire surface patches to describe the global surface properties. This method can take hours to describe simple objects. Our approach integrates global information into parametric surface descriptions and runs in minutes on similar objects with similar computing power.

Many researchers have favored the extraction of lower-dimensional features, such as edges, to describe range images instead of surfaces [97], [59], [60], [39], [76], [100], [95], [54], [50], [16], [13]. The uniformity predicate in these approaches requires that range and the range gradient are continuous for all pixels in region interiors where only the region boundaries are computed explicitly. By detecting and linking range and range gradient discontinuities, the space curves that bound arbitrary smooth surfaces are isolated creating an image segmentation. However, most of the above edge-based work has focussed on straight and circular edges for matching with polyhedra and cylinders and their combinations. Although edge-based approaches offer important computational advantages for today's computer vision systems, we believe that such systems cannot provide the detailed surface information that will be required from future general-purpose range-image vision systems. Only more research experience will determine the advantages and disadvantages of these approaches for different applications, but the generality of an arbitrary surface segmentation and surface description approach is necessary today for automated free-form, sculptured surface reconstruction and shape acquisition tasks as in [84].

II. Problem Definition

In the surface-based approach to segmentation, the relevant structure of an image is viewed as a piecewise-smooth graph surface contaminated by noise as defined below. We emphasize the geometric shape of the image data in this approach, *not* the noise process as in random field image models [106], [26], [30], [24]. Several terms are introduced to give a reasonably precise description of the problem we are attempting to solve.

A 3-D *smooth graph surface* is a twice-differentiable function of two variables:

$$z = f(x, y). \tag{3}$$

A *piecewise-smooth graph surface* $g(x, y)$ can be partitioned into smooth surface primitives $f_l(x, y)$ over support regions R_l:

$$z = g(x, y) = \sum_{l=1}^{N} f_l(x, y)\, \chi(x, y, R_l) \tag{4}$$

where $\chi(x, y, R_l)$ is the characteristic function of the region R_l, which is unity if $(x, y) \in R_l$ and zero otherwise. For each piecewise-smooth surface $g(x, y)$, it is convenient to associate a *region label function* $l_g(x, y)$ defined as

$$l_g(x, y) = \sum_{l=1}^{N} l\, \chi(x, y, R_l). \tag{5}$$

If $\vec{a}_l$ is the vector of all parameters needed to precisely specify the smooth function $f_l(x, y)$, then any piecewise-smooth surface may be represented as the piecewise-constant function $l_g(x, y)$ (with minimum value 1 and maximum value N), which contains all segmentation information, and the list of N parameter vectors $\{\vec{a}_l\}$, which contains all shape information.

A *digital surface* is a noisy, quantized, discretely sampled version of piecewise-smooth graph surface:

$$z_{ij} = \tilde{g}(i, j) = \lfloor a\big(g(x(i), y(j)) + n(x(i), y(j))\big) + b \rfloor \tag{6}$$

where a and b are the quantizer's scale factor and offset respectively, the floor function indicates truncation (quantization) to an integer, and the additive noise process $n(x, y)$ is nominally zero-mean with finite variance

$\sigma^2(x, y)$ at each point. The discrete image location (i, j) need not be linearly related to the Euclidean (x, y) location allowing for the nonlinear relationships involved in some range sensors (see Appendix). A *range image* is a particular type of a digital surface where the z_{ij} values represent the distance to a physical surface from a reference surface. An *intensity image* is another type of digital surface where the z_{ij} values represent the number of visible photons incident at the (i, j) location in the focal plane of a camera. Other image types are defined based on the meaning of the sensed z_{ij} values. This underlying model is quite general and can be used to represent many types of images unless multiplicative noise or some other type of nonadditive noise is present. Many textured surfaces may also be considered as an approximating smooth surface plus random sensor noise along with structured noise to represent the given texture.

The *segmentation/reconstruction* problem that we are attempting to solve is a generalization of the segmentation problem and may be stated as follows. Given only a digital surface, denoted $\bar{g}(i, j)$ and specified by the z_{ij} values, find $\hat{N}$ approximating functions $\hat{f}_l(x, y)$ and $\hat{N}$ image regions $\hat{R}_l$ over which those functions are evaluated such that the total image representation error

$$\epsilon_{\text{tot}} = \left\| \bar{g}(i, j) - \hat{g}(x(i), y(j)) \right\|_I \tag{7}$$

between the reconstructed image function

$$\hat{g}(x, y) = \sum_{l=1}^{\hat{N}} \hat{f}_l(x, y)\, \chi(x, y, \hat{R}_l) \tag{8}$$

evaluated at the points $(x(i), y(j))$ and the data $\bar{g}(i, j)$ is small and the total number of functions and regions $\hat{N}$ is small. The function norm is left unspecified, but may be the max norm, the (Euclidean) root-mean-square error norm, or the mean absolute error norm. The implicit logical segmentation predicate in the above problem statement may be written as the *surface coherence* predicate:

$$P_\epsilon(R_l) = \begin{cases} \text{TRUE} & \text{if } \left\| \bar{g} - \hat{f}_l \right\|_{R_l} < \epsilon \\ \text{FALSE} & \text{otherwise} \end{cases} \tag{9}$$

where the value of ϵ depends on the mean variance of the noise process $n(x, y)$ in the image region.

The two trivial solutions may be discarded immediately. The "one pixel per region" solution minimizes the approximation error (zero error), maximizes the number of regions, and requires no work, but is of course also useless. The "one function per image" solution minimizes the number of regions (one region), maximizes the approximation error, requires work, may be useful for some purposes, but does not solve the real problem. We seek an algorithm that tends to segment images into regions that can be directly associated with meaningful high-level entities. In the case of range images, the surface functions defined over the image regions should mathematically represent the 3-D shape of visible physical surfaces in the scene.

The problem statement places no constraints on the functions except that they are smooth, no constraints on the image regions except that they are connected, no constraints on the form of the additive noise term except that it is zero-mean. We want the total approximation error and the number of regions to be small, but we have not attempted to weight the relative importance of each. Without such weights, it is difficult to form an objective function and apply existing optimization methods.

It is not at all clear from the above statement that such a "chicken-and-egg" segmentation problem can be solved at the signal level. It is straightforward to fit functions to pixel data over regions if the regions have been determined, but how are the regions to be determined? Similarly, it is possible to determine the image regions if the set of functions are known, but how are the functions extracted? But even the number of functions/regions present in the data is not known. We seek a signal-level, data-driven segmentation procedure based only on knowledge of piecewise-smooth surfaces.

III. Smooth Surface Decomposition

The problem statement says that the smooth component functions $f_l(x, y)$ of the underlying model $g(x, y)$ are allowed to be arbitrary smooth surfaces, which can be arbitrarily complicated. However, arbitrary smooth surfaces can be subdivided into simpler regions of constant surface curvature sign based on the signs of the mean and Gaussian curvature at each point [10]. As shall be discussed in more detail later, there are only eight possible surface types surrounding any point on a smooth surface based on surface curvature sign: peak, pit, ridge, valley, saddle ridge, saddle valley, flat (planar), and minimal. These fundamental surface shapes, shown in Fig. 1, are very simple and do not contain inflection points (compare to codons for planar curve description as in [87]. Our hypothesis is that these simple surface types are well approximated for image segmentation (i.e., perceptual organization) purposes by bivariate polynomials of order M or less where M is small. The experimental results included here and in [8] attempt to show that this assumption is reasonable for a large class of images when $M = 4$ (biquartic surfaces). Even the range image surfaces of quadric primitives can be approximated well enough for segmentation purposes with such polynomial surfaces. This assumption is only limiting in the context of the segmentation algorithm when a large smooth surface bends much faster than x^4. If a particular application encounters a significant number of such surfaces, the limit of $M = 4$ can be raised. If a range imaging application can guarantee that only planar and quadric surfaces will appear, they can use only those types of functions for fitting purposes. In fact, any ordered set of bivariate approximating functions can be used if they satisfy the set of requirements defined below. In summary, *arbitrary smooth surfaces may be decomposed into a union of simple surface-*

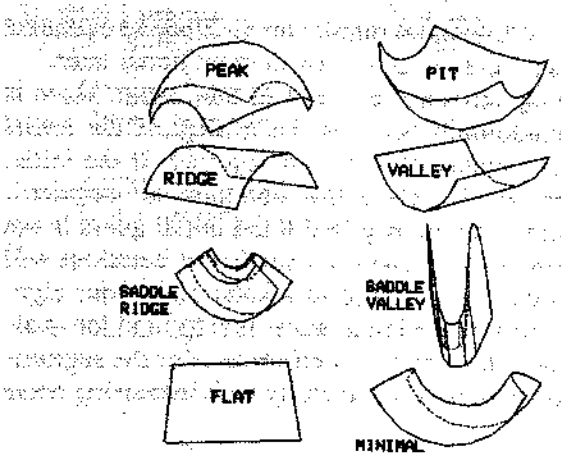


Fig. 1. Eight fundamental surface types from surface curvature sign.

curvature-sign primitives that are well approximated by low-order bivariate polynomials.

The intermediate goal of a segmentation algorithm then is to isolate all underlying *simple surfaces* (surface-curvature-sign primitives) in the image data and fit those simple surfaces with simple bivariate approximating functions. This creates an image segmentation in terms of the support regions of the simple approximating functions and an image reconstruction in terms of those simple approximating functions evaluated over the support regions. If the boundaries of the *smooth surfaces* of the underlying piecewise-smooth surface model are desired, then smoothly joining, adjacent simple surface regions can be merged to create the smooth surface support regions. This noniterative postprocessing step is covered in [8] and is not discussed here. We currently leave the function description over the smooth surface regions in the form of a collection of simple polynomial surfaces. The final collection of smooth surfaces and their support regions is the underlying piecewise-smooth image description that we wished to recover in the problem definition. In applications, it may be desirable to go back and fit the segmented smooth surface with application surfaces, such as quadrics, composite Bezier patches [35], or rational B-splines [99], rather than leaving it as a set of polynomial surfaces.

A. Approximating Function Requirements

Arbitrarily complicated smooth surfaces can be decomposed into a disjoint union of surface-curvature-sign surface primitives. If these surface primitives can be approximated well by a small set of approximating functions, a composite surface description for arbitrary smooth surfaces can be obtained. There are several constraints that the set of approximating functions must satisfy. Of course, the approximating functions must be able to approximate the fundamental surface-curvature-sign surface-primitives well. For *dimensionality reduction* reasons, the approximating functions should be representable by a small amount of data. For *generality*, the approximating surfaces must be well-defined over any arbitrary connected region in the image plane. The approximating functions of the digital surface segmentation must be useful for *extrapolation* into neighboring areas of a surface in order for the region growing method to be successful. *Interpolation* capabilities are also useful for evaluating points between pixels if surface intersections are required. The approximating functions should also be easily *differentiable* so that differential geometric shape descriptors can be recomputed from them and so that other processes may compare surface normals and other differential quantities. Finally, the complete set of approximating functions should be *totally ordered* so that each approximant is capable of describing lower order approximants exactly, but cannot adequately approximate higher order functions. This provides the basis for a set of increasingly complicated hypotheses about the form of the underlying data. Note that general 3-D surface representation capability is not needed because digital surfaces are discrete representations of graph surface functions.

Low-order bivariate polynomials satisfy all of the above requirements, and the surface fitting procedure requires only a linear least-squares solver for the $p \times q$ $(p > q)$ equation $[A]\vec{x} = \vec{b}$ [70], [5]. We have found that the set of planar, biquadratic, bicubic, and biquartic polynomials performed well in our experiments without significant computational requirements (a few seconds per fit on a VAX 11-780). *However, any set of approximating functions that satisfy the above constraints may be used instead.* To maintain generality in the algorithm description, it is only assumed that there is a set of approximating functions, denoted as F, that contains $|F|$ discrete types of functions that can be ordered in terms of the "shape potential" of each surface function relative to the set of fundamental surface-curvature-sign surface primitives.

In our case $|F| = 4$ and the set of approximating functions F can be written in the form of a single equation:

$$\hat{f}(m, \vec{a}; x, y) = \sum_{i+j \le m} a_{ij} x^i y^j \tag{10}$$

$$= a_{00} + a_{10}x + a_{01}y + a_{11}xy + a_{20}x^2 + a_{02}y^2 + a_{21}x^2y + a_{12}xy^2 + a_{30}x^3 + a_{03}y^3 + a_{31}x^3y + a_{22}x^2y^2 + a_{13}xy^3 + a_{40}x^4 + a_{04}y^4. \quad (m = 4) \tag{11}$$

Planar surfaces are obtained by restricting the parameter vector space $\Re^{15}$ to three-dimensional subspace where only a_{00}, a_{10}, a_{01} may be nonzero. Biquadratic surfaces are restricted to a six-dimensional subspace, and bicubic surfaces to a ten-dimensional subspace. A least-squares solver computes the parameter vector $\vec{a}$ and the RMS fit error ϵ from the digital surface data over a region quickly and efficiently. Moreover, a QR least-squares solution approach allows surface region fits to be updated recursively during the region growing process as new data points are added [42], [22] for better computational efficiency.

B. Simple to Complex Hypothesis Testing

The key idea behind the algorithm, which is independent of the set of approximating functions actually chosen, is that one should start with the simplest hypothesis about the form of the data and then gradually increase the complexity of the hypothesized form as needed. This is the *variable-order* concept, which has not been used in previous segmentation algorithms. In our case, surface type labels at each pixel allow us to find large groups of identically labeled pixels. Then, a small subset of those pixels, known as a *seed region*, is chosen using a simple shrinking method that attempts to ensure that every pixel in the seed region is correctly labeled. The simplest hypothesis for any surface fitting approach is that the data points represented in the seed region lie in a plane. The hypothesis is then tested to see if it is true. If true, the seed region is grown based on the planar surface fit. If the simple hypothesis is false, the algorithm responds by testing the next more complicated hypothesis (e.g., a biquadratic surface). If that hypothesis is true, the region is grown based on that form. If false, the next hypothesis is tested. This process continues until either 1) all preselected hypotheses have been shown to be false or 2) the region growing based on the surface fitting has converged in the sense that the same image region is obtained twice. Since all smooth surfaces can be partitioned into simple surfaces based on surface curvature sign, false hypotheses may occur only when the isolated seed region surface-type labels are incorrect (due to noise) or when the underlying surface bends faster than the highest order approximating surface. During execution of the algorithm, bad seed regions are rejected immediately when the surface fit error is poor and large quickly bending surfaces are broken into two or more surface regions.

IV. Algorithm Philosophy

This section includes qualitative comments about the system structure of the surface-based segmentation algorithm. The success of the algorithm is based on the effective combination of simple component algorithms, not on the capabilities of any single processing step.

A. Initial Guess Plus Iteration

Like many region growing schemes, the basic approach of this algorithm might be summarized as "make an initial guess and then iteratively refine the solution." This idea is at least as old as Newton's method for finding the zeros of a complicated function. Unlike other region growing schemes, the initial guess at the underlying surface segmentation is based on invariant differential geometric principles and is quantified in terms of surface curvature sign labels, or surface type labels [10]. The iterative refinement process is based on function approximation and region growing. Once a surface has been fitted to the kth group of connected pixels, the $(k + 1)$th group of pixels is obtained by finding all new connected pixels that are compatible with the fitted surface of the previous group. When the same group of pixels is obtained twice, the iteration terminates yielding an extracted region. This process is described in more detail later.

Although we shall not prove it in this paper, there is the usual relationship between the quality of the initial guess and the number of iterations required. If the initial guess is very good, only a few iterations are required. Many iterations may be required if the initial guess is not good. For bad initial guesses, no number of iterations will yield the proper convergence to a solution. In our algorithm, the quality of the initial guess is related to the quality of the image data, and the performance of the segmentation algorithm degrades gracefully with increasing noise levels.

B. Stimulus Bound Image Analysis

The variable-order surface fitting approach may be thought of as a hypothesize and test (hypothesize and verify) algorithm where the hypotheses can be automatically changed by the input data and each surface fit is bound by (must conform to) the input data. Therefore, we suggest the use of the adjective *stimulus bound* [89] for the type of hypothesis testing done by the surface-based segmentation algorithm, where the *stimulus* is the original sensed data values. In a stimulus bound process, all interpretive processing of the data is *bound* to or constrained by the original data or stimulus in each stage of processing to reduce the probability of interpretation errors. In our case, each simple surface function hypothesis is tested against the original data via surface fitting followed by two tests: 1) an RMS fit error test (related to the chi-square test), and 2) a regions test (related to the nonparametric statistics runs test). Hence, each iteration and the final interpretation are bound by the original stimulus.

It is generally acknowledged that vision algorithms should function at several different levels using associated vision modules to process the signal and symbol information at different levels. It often occurs that each level's vision module accepts input only from the previous, lower level and provides output only to the subsequent, higher level. Fig. 2(a) shows a typical example of such a process. This assumption may be rooted in human visual models where retinal information is not directly available to the high level cerebral processes. However, human vision is a fundamentally dynamic perceptual process in which subsequent, highly correlated "video frames" are always immediately available to the visual system after any given instant in time. Therefore, it may be inappropriate to apply dynamic human visual model principles to static computational vision problems. The *stimulus bound* philosophy states that the output from all lower level vision modules should be available to high-level vision modules. In particular, the original image from the sensor must be available to every vision module in a static vision system as shown in Fig. 2(b). In the surface-based segmentation algorithm, *every pixel in every region* is constantly checked to see how close the sensed value at a given pixel is to the approximating surface function for the given region. The global grouping of pixels relies on

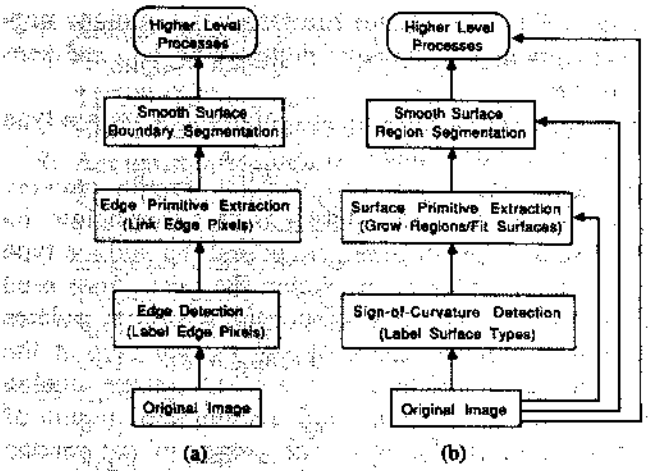


Fig. 2. Conventional edge approach versus stimulus bound approach.

simple differencing between pixel values and the interpreted surface primitives. These concepts are expressed in more detail in later sections. Without constantly checking symbolic interpretations against the original data, a vertical chain of interpretive vision modules is only as robust as the weakest module. Many edge-based intensity image vision schemes have failed in practice because precise, correctly linked edges could not be extracted from real images.

We believe that surface-based range image vision systems have an advantage over edge-based systems in that it is possible to quickly check final object or surface interpretations against the original data (via simple image differencing) because synthetic range images can be generated from models with a depth-buffer algorithm [78] using only the object or surface geometry; light and surface reflectance are not involved. The final interpretation in the form of a reconstructed range image can be subtracted from the original range image to create an *interpretation error image*, which can then be used to evaluate the quality of the image interpretation globally and locally. This is only possible when an image interpretation includes segmentation and reconstruction information as described in the problem definition.

C. Emerging Commitment

The surface-based segmentation algorithm is primarily data-driven in that only generic knowledge of surfaces, curvature, noise, and approximation are used. Of course, data-driven and model-driven elements must cooperate in any algorithm that attempts to interpret a digital image in terms of specific model information. An important feature of any image interpretation approach is the *process of commitment* to the final interpretation. A special-purpose model-driven program can make a commitment to its set of possible interpretations when the program is written or compiled [41], thus avoiding certain computations that might otherwise be required. A data-driven program may postpone making a commitment to a final interpretation in order to be more generally applicable, but it should reduce the amount of information that must be manipulated by later higher-level processes that use specific world model information by generating intermediate symbolic primitives. We believe that our approach follows a *principle of emerging commitment* that is *gradual* and *locally reversible*, but not random. One must make steps toward image interpretation, yet it is impossible to always avoid errors that necessitate steps or labels being undone. An algorithm should make a series of small steps towards the goal, where each step need not produce perfect results, can easily be undone, but still produces useful information for the next step. Simulated annealing algorithms [4], [38] might also be said to follow a principle of emerging commitment, but the surface-based segmentation algorithm described here is very directed in its search process and provides a much more structured output.

V. Algorithm Description

The algorithm presented in this paper uses a general piecewise-smooth surface model to do pixel grouping assuming the image data exhibits *surface coherence* properties. If all pixel values are viewed as noisy samples of an underlying *piecewise-smooth surface* function defined over the entire image, the segmentation process should not only provide detailed definitions of the segmented regions, but should also provide the component surface functions of the underlying piecewise-smooth surface. Surface-based segmentation includes surface (image) reconstruction.

In the first stage of the segmentation algorithm described below, each pixel in an image is given a label based on its value and the values of its neighboring pixels. This label can only take on eight possible values based on two surface-curvature signs and indicates the qualitative shape of an approximating surface that best-fits the image data surrounding that point. This surface-type label image can be analyzed for connected regions using a standard connected component analysis algorithm [91], [3]. Any individual pixel label can be wrong, but it is likely to be correct if it lies in the interior of a large region of identically labeled pixels. Moreover, due to the constrained nature of the surface types represented by the eight labels, it is also likely that a simple surface function will approximate a group of correctly labeled pixels. The surface type label image is used to provide seed regions to the region-growing algorithm. A pixel's similarity with a group of other pixels is measured by comparing 1) the difference between the pixel's value and the other pixels' approximating surface value (at the given pixel location) and 2) a parameter that measures the goodness of the fit of the surface to the other pixels. This similarity measure allows for a pixel to enter and leave a group of pixels depending on the other pixels currently in that group. Hence, a mistake in grouping can be undone. Regions are grown until convergence criterion are met, and a concise, parametrically defined surface description is recorded along with a definition of the image region. It is common for images reconstructed from the segmentation description to be almost indistinguishable from the original image.

This algorithm can be viewed as a two-stage process. The first stage computes an "initial-guess" coarse segmentation in terms of regions of identical surface type labels. The second stage iteratively refines the coarse image segmentation and simultaneously reconstructs image surfaces. The entire algorithm is outlined below.

The first stage creates a surface type label image $T(i, j)$ from the original image $\bar{g}(i, j)$ in the following manner:

- Compute partial derivative images $\bar{g}_u(i, j)$, $\bar{g}_v(i, j)$, $\bar{g}_{uu}(i, j)$, $\bar{g}_{vv}(i, j)$, $\bar{g}_{uv}(i, j)$ from the original image $\bar{g}(i, j)$ using local fixed-window surface fits that are accomplished via convolution operators.
- Using the partial derivative images, compute the mean curvature image $H(\bar{g}_u, \bar{g}_v, \bar{g}_{uu}, \bar{g}_{vv}, \bar{g}_{uv})$ and the Gaussian curvature image $K(\bar{g}_u, \bar{g}_v, \bar{g}_{uu}, \bar{g}_{vv}, \bar{g}_{uv})$.
- Compute the sign $(+, -, 0)$ of mean curvature, denoted sgn (H), and the sign of Gaussian curvature, denoted sgn (K). The signum function sgn (x) maps negative numbers to -1, positive numbers to $+1$, and zero maps to zero.
- Use surface curvature sign to determine a surface type label $T(i, j)$ for each pixel (i, j).

The second stage performs iterative region growing using variable-order surface fitting as described below. Its input consists of the original image and the surface type label image. In order to determine the next (first) seed region to use, a connected component algorithm isolates the largest connected region of any surface type in the $T(i, j)$ image, and then a 3×3 binary image erosion operator shrinks the region until a small seed region of appropriate size is obtained. The output of the second stage consists of a region label image $\hat{l}_g(i, j)$, which contains all region definitions in one image (the segmentation information), and a list of coefficient vectors $\{\vec{a}_l\}$, one for each region (the shape reconstruction information).

---

1) **Declarations:**
   Surface-Order: $m \in F = \{1$ (Planar), 2 (Biquadratic), 3 (Bicubic), 4 (Biquartic)$\}$;
   Max-Surface-Order: $|F| = 4$ (Biquartic);
   Surface-Fit: $\{\vec{a}$ = Coefficient Vector (3, 6, 10, or 15 numbers), σ = RMS Fit Error$\}$;
   Surface-Type-Image: $T(i, j)$ where $T \in \{1, 2, 3, 5, 6, 7, 8, 9\}$;
   Region-Label-Image: $\hat{l}_g(i, j)$ where $\hat{l} \in \{1, \cdots, \hat{N}\}$;
   Surface-Fit-List: $\{\vec{a}_l\}$ where $l \in \{1, \cdots, \hat{N}\}$;
   Reconstruction-Image: $\hat{g}(i, j)$;
   Error-Image: $e(i, j) = |\hat{g}(i, j) - \bar{g}(i, j)|$;
   Current-Region, New-Region, Seed-Region: Four-Connected Subsets of Image I
2) **Initialization:**
   Set *Error-Image* = Big Error Value;
   Set *Reconstruction-Image* = No Value;
   Set *Region-Label-Image* = No Label;
3) **Start-Iteration:**
   Set *Surface-Order* = Planar $(z = a + bx + cy)$;
   Set *Seed-Region* = Next-Seed-Region (*Surface-Type-Image*);
   IF *Seed-Region* Smaller Than Threshold Size (e.g., 30 pixels),
   THEN GoTo **All-Done;**
   ELSE Set *Current-Region* = *Seed-Region*;
4) **Surface-Fitting:**
   Perform *Surface-Order* Fit to z_{ij} Values in *Current-Region* to obtain *Surface-Fit*;
5) **Surface-Fit-Testing:**
   IF *Surface-Fit* OK using RMS Error Test and Regions Test,
   THEN GoTo **Region-Growing;**
   ELSE Increment *Surface-Order*;
   (Example: Planes become Biquadratics: $z = a + bx + cy + dxy + ex^2 + fy^2$)
   IF *Surface-Order* > *Max-Surface-Order*,
   THEN GoTo **Accept-Reject;**
   ELSE GoTo **Surface-Fitting**;
6) **Region-Growing:**
   Find *New-Region* Consisting of Compatible Connected Neighboring Pixels where Compatibility means Pixel Values must be Close to Surface and Residual Error must be Smaller Than Current Value in Error Image and Derivative Estimates from Pixel Values must be Close to Surface Derivatives;
   IF *Current-Region* = *New-Region*,
   THEN GoTo **Accept-Reject;**
   ELSE Set *Current-Region* = *New-Region*; GoTo **Surface-Fitting**;

7) **Accept-Reject:**
IF *Surface-Fit* OK using RMS Error Test,
THEN GoTo **Accept-Surface-Region;**
ELSE Zero Out *Seed-Region* Pixels in *Surface-Type-Image*; GoTo **Start-Iteration;**
8) **Accept-Surface-Region:**
Zero Out *Current-Region* Pixels in *Surface-Type-Image*;
Label *Current-Region* Pixels in *Region-Label-Image*;
Evaluate *Current-Region* Pixels in *Reconstruction-Image* using *Surface-Fit*;
Update *Current-Region* Pixels in *Error-Image* with Absolute Residual Errors;
Add *Surface-Fit* to *Surface-Fit-List*;
GoTo **Start-Iteration;**
9) **All-Done:**
Surface-Fit-List Contains All Function Definitions for Image Reconstruction
Region-Label-Image Contains All Region Definitions for Image Segmentation
Reconstruction-Image Contains Noiseless, Smooth Surface Version of Original Image;
Error-Image Contains Approximation Error at Each Pixel to Evaluate Reconstruction Quality;

It is not necessary to maintain a separate version of the reconstructed image as this can always be recomputed from the surface fit list and the region label image. However, displaying this image during program execution is an excellent way to monitor the progress of the algorithm. The error image can also be recomputed from the surface fit list, the region label image, and the original image, but it is maintained throughout the iteration process to counteract the tendency of surfaces without sharp boundaries to grow slightly beyond their actual boundaries. The error image is updated at each pixel with the absolute error between the approximating surface and the original data when a surface/region is accepted. During the region growing procedure, the error image is consulted to see if the current approximating function represents a given pixel better than it has been represented before. If so, a pixel that was labeled as a member of a previously determined region is free to be labeled with a better fitting region as long as the pixel is connected to the better fitting region. Thus, *labeling decisions are reversible*. Later surfaces in this sequential algorithm can relabel a pixel even though it was already labeled as part of another surface.

The algorithm above terminates when the next seed region extracted from the surface type label image is too small (e.g., less than 30 pixels). However, some pixels may still be unlabeled at this point. These pixels are coalesced into a binary surface type image in which all pixels that have already been labeled are turned off (black) leaving all unlabeled pixels on (white). This new "left-overs" surface type image is then processed by extracting and fitting the next seed region as usual except that the region growing constraints are relaxed (e.g., the allowable RMS fit error limit is doubled). When the next seed region from the left-overs surface type image is too small, the algorithm finally terminates.

The outline above provides a high-level description of all the main elements of the segmentation algorithm. We have omitted several details that are covered in subsequent sections. The algorithm as stated here does not always yield clean high-quality edges between regions, and it is still possible that some pixels may be left unlabeled (ungrouped with a surface). Hence, a local region refinement operator capable of cleaning up pixel-size irregularities was used to create the final segmentations shown in the experimental results section. Also, as mentioned above, surface curvature sign primitive regions must be merged at polynomial surface primitive boundaries that lie within the boundaries of a smooth surface. The details on a region refinement operation and a one-step region merging method for smoothly joining surface primitive boundaries are available in [8], and further enhancements are currently being developed. These fine points are not at all related to the performance of the segmentation algorithm as described here since the necessary procedures are performed after the termination of the iterative region growing.

VI. Noise Estimation for Threshold Selection

Digital surfaces exhibit the property of surface coherence when sets of neighboring pixels are spatially consistent with each other in the sense that those pixels can be interpreted as noisy, quantized, sampled points of some relatively smooth surface. In order for the surface-based segmentation algorithm to group pixels based on underlying smooth surfaces, it needs to know how well the approximating functions should fit the image data. This information should be derived from the image data in a data-driven algorithm. If the noise in the image is approximately stationary ($\sigma^2(x, y) \approx \sigma^2_{\text{img}} = \text{constant}$), we can compute a single estimate of the noise variance σ^2_{img} (that should be applicable at almost all image pixels) by averaging estimates of the noise variance at each pixel. To compute an estimate of the noise variance at each pixel, we perform a equally-weighted least-squares planar fit in the 3×3 neighborhood W_3 surrounding the pixel. If the pixel lies in the interior portion of a smooth surface region and if the radius of the mean surface curvature is larger than a few pixels, the error in the planar surface fit will be primarily due to noise. In contrast, steeply sloped image regions typically have large mean curvatures and

bad planar fits. To get a good estimate of the magnitude of the additive noise and the quantization noise in the image, it is necessary to exclude these pixels where the gradient magnitude is large. Therefore, we only include pixels in the mean noise variance calculation if the gradient magnitude is below a preset threshold (8 levels/pixel was used in our experiments). A more detailed discussion of this idea is given in [8]. The equation for the mean image noise variance σ^2_{img} may be expressed as

$$\sigma^2_{\text{img}} = E(\sigma^2_{W_3}) = \frac{1}{N_{\text{int}}} \sum_{I=1}^{N_{\text{int}}} \left(\sum_{p \in (R_I - \partial R_I)} \sigma^2_{W_3}(p) \right) \quad (12)$$

where ∂R represents the boundary of the region R, N_{int} is the total number of surface interior pixels contributing to the sum, and where $\sigma_{W_3}(p)$ is the root-mean-square-error (RMSE) of the least-squares planar surface fit (a_{00}, a_{10}, a_{01}) in the 3×3 window W_3 around the pixel p:

$$\sigma^2_{W_3}(p) = \tfrac{1}{9} \sum_{(i,j) \in W_3} \left(z_{ij} - (a_{00} + a_{10} i + a_{01} j) \right)^2 \quad (13)$$

where i and j are interpreted as integer row and column coordinates. Although the regions themselves are not known at the time the noise variance is estimated, we get a good approximation to σ_{img} by not averaging pixels with high slopes using the preset threshold.

The noise variance estimate allows us to automatically set the ϵ parameter of the surface coherence predicate (the maximum allowable RMS surface fit error) and two other thresholds to an appropriate value as described later. Note that we are attempting to estimate noise variance for *continuous smooth surface detection* purposes, *not for discontinuity detection* as in [45]. Although we do not claim to have solved the automatic threshold selection problem, the three relevant thresholds are directly tied to the geometric and statistical properties of the data via empirical relationships providing good performance for many images. Other noise variance estimation techniques, such as computing the mean square difference between a median filtered version of an image and the original image, are currently being evaluated.

VII. Surface Type Labeling

Differential geometry states that local surface shape is *uniquely determined* by the first and second fundamental forms. Gaussian and mean curvature combine these first and second fundamental forms in two different ways to obtain scalar surface features that are *invariant to rotations, translations, and changes in parameterization* [8]. Therefore, visible surfaces in range images have the same mean and Gaussian curvature from any viewpoint under orthographic projection. Also, *mean curvature uniquely determines the shape of graph surfaces* if a boundary curve is also specified [40] while *Gaussian curvature uniquely determines the shape of convex surfaces and convex regions of nonconvex surfaces* [23], [55]. There are eight fundamental viewpoint independent surface types that can be characterized using only the sign of the mean curvature (H) and Gaussian curvature (K) as shown in Fig. 3.

	$K > 0$	$K = 0$	$K < 0$
$H < 0$	Peak T=1	Ridge T=2	Saddle Ridge T=3
$H = 0$	*(none)* T=4	Flat T=5	Minimal Surface T=6
$H > 0$	Pit T=7	Valley T=8	Saddle Valley T=9

Fig. 3. Surface type labels from surface curvature sign.

3. Gaussian and mean curvature can be computed directly from a range image using window operators that yield least squares estimates of first and second partial derivatives as in [2], [6], [48]. The key point is that every pixel in an image can be given a surface type label based on the values of the pixels in a small neighborhood about that pixel.

Surface curvature estimates are extremely sensitive to noise because they require the estimation of second derivatives, in which high frequency noise is amplified. In fact, 8-bit quantization noise alone can seriously degrade the quality of surface curvature estimates unless large window sizes are used (at least 9×9). Yet reliable estimates of surface curvature sign can still be computed in the presence of additive noise and quantization noise [10]. Since we need to compute five different derivative estimates to compute surface curvature, we could use large $N \times N$ derivative estimation window operators (N odd), or we can smooth the image with a small $L \times L$ window operator (L odd), store the smoothed values at higher precision, and operate on the smoothed image with smaller $M \times M$ derivative estimation window operators (M odd) where $L + M = N + 1$. Assuming window separability and therefore linear time requirements, the former requires time proportional to $5N$ whereas the latter requires time proportional to $N + 4M + 1$. The relative weighting factors used in determining the derivative and smoothing window coefficients have an important influence on the quality of the derivative estimates. In our experiments with 8-bit images, we obtained good consistent results using one 7×7 binomial weight (approximately Gaussian) smoother and five 7×7 equally weighted least squares derivative estimation operators with over 30 percent fewer computations than the equivalent 13×13 windows. For reference purposes, we list the specific numbers needed for this particular computation.

Since all our operators are separable, window masks can be computed as the outer product of two column vectors. The binomial smoothing window may be written as $[S] = \vec{s}\,\vec{s}^{\,T}$ where the column vector $\vec{s}$ is given by

$$\vec{s} = \tfrac{1}{64} [1 \;\; 6 \;\; 15 \;\; 20 \;\; 15 \;\; 6 \;\; 1]^T. \quad (14)$$

For 7×7 binomial smoothing window, it is clear that we should try to maintain an extra 12 bits ($12 = 2 \log_2(64)$) of fractional information in the intermediate image smoothed by $[S]$. For an $L \times L$ binomial smoother, $2L - 2$ bits of fractional information should be maintained. The equally weighted least-squares derivative estimation window operators are given by

$$[D_u] = \vec{d}_0 \vec{d}_1^T \quad [D_v] = \vec{d}_1 \vec{d}_0^T \quad [D_{uu}] = \vec{d}_0 \vec{d}_2^T$$
$$[D_{vv}] = \vec{d}_2 \vec{d}_0^T \quad [D_{uv}] = \vec{d}_1 \vec{d}_1^T \tag{15}$$

where the column vectors $\vec{d}_0$, $\vec{d}_1$, $\vec{d}_2$ for a 7×7 window are given by

$$\vec{d}_0 = \tfrac{1}{7}[1\ 1\ 1\ 1\ 1\ 1\ 1]^T \tag{16}$$

$$\vec{d}_1 = \tfrac{1}{28}[-3\ -2\ -1\ 0\ 1\ 2\ 3]^T \tag{17}$$

$$\vec{d}_2 = \tfrac{1}{84}[5\ 0\ -3\ -4\ -3\ 0\ 5]^T \tag{18}$$

The partial derivative estimate images are computed via the appropriate 2-D image convolutions (denoted *):

$$\begin{aligned} \tilde{g}_u(i,j) &= D_u * S * g(i,j) \\ \tilde{g}_v(i,j) &= D_v * S * g(i,j) \end{aligned} \tag{19}$$

$$\begin{aligned} \tilde{g}_{uu}(i,j) &= D_{uu} * S * g(i,j) \\ \tilde{g}_{vv}(i,j) &= D_{vv} * S * g(i,j) \\ \tilde{g}_{uv}(i,j) &= D_{uv} * S * g(i,j) \end{aligned} \tag{20}$$

Mean (H) and Gaussian (K) curvature images are computed using the partial derivative estimate images:

$$H(i,j) = \frac{\left(1 + \tilde{g}_v^2(i,j)\right)\tilde{g}_{uu}(i,j) + \left(1 + \tilde{g}_u^2(i,j)\right)\tilde{g}_{vv}(i,j) - 2\tilde{g}_u(i,j)\,\tilde{g}_v(i,j)\,\tilde{g}_{uv}(i,j)}{2\left(\sqrt{1 + \tilde{g}_u^2(i,j) + \tilde{g}_v^2(i,j)}\right)^3} \tag{21}$$

$$K(i,j) = \frac{\tilde{g}_{uu}(i,j)\,\tilde{g}_{vv}(i,j) - \tilde{g}_{uv}^2(i,j)}{\left(1 + \tilde{g}_u^2(i,j) + \tilde{g}_v^2(i,j)\right)^2}. \tag{22}$$

A toleranced signum function

$$\operatorname{sgn}_\epsilon(x) = \begin{cases} +1 & \text{if } x > \epsilon \\ 0 & \text{if } |x| \le \epsilon \\ -1 & \text{if } x < \epsilon \end{cases} \tag{23}$$

is used to compute the individual surface curvature sign images $\operatorname{sgn}_\epsilon(H(i,j))$ and $\operatorname{sgn}_\epsilon(K(i,j))$ using a preselected zero threshold ϵ. For our experimental results, we used $\epsilon_H = 0.03$ and $\epsilon_K = 0.015$ for 7×7 windows. Ideally, these thresholds should depend on the noise variance estimate, but the algorithm performance is not very sensitive to these numbers for reasonable quality images.

The surface curvature sign images are then used to determine the surface type image:

$$T(i,j) = 1 + 3\left(1 + \operatorname{sgn}_{\epsilon_H}(H(i,j))\right) + \left(1 - \operatorname{sgn}_{\epsilon_K}(K(i,j))\right). \tag{24}$$

This equation is shown in table form in Fig. 3. Fig. 1 displayed the eight fundamental shapes. Depending on the number of digitized bits and the amount of noise in the original image and the window sizes used in derivative estimation, regions of a given surface type label tend to connect (in the sense of four-connectedness) with distinct, but adjacent regions with the same label. Therefore, it is not always possible, in general, to simply isolate a four-connected region of pixels of a particular surface type label and identify that region as a single surface of the appropriate type for surface fitting purposes. Hence, there is a need for a general purpose method to isolate useful interior seed regions from the larger regions of identical surface type labels extracted from the surface type label image $T(i,j)$.

VIII. Seed Region Extraction

We adopted the following strategy that breaks the unwanted connections with other adjacent regions and attempts to provide small, maximally interior regions that are good for surface fitting. The largest connected region of any fundamental surface type in the surface type label image is isolated (denoted R_T^0) and is then eroded (contracted) repetitively (using a 3×3 binary region erosion operator) until the region disappears. After the kth contraction (erosion), there exists a largest four-connected subregion R_T^k in the pixels remaining after the k contractions of the original region. If we record $|R_T^k|$, the number of pixels in the largest connected subregion, as a function of the number of contractions k, a *contraction profile* for the original region is created. Contraction profiles for five regions of a surface type label image (for the coffee cup range image) are shown in Fig. 4. A seed region size threshold t_{seed} for the minimum number of pixels required to be in a seed region (e.g., 10) is a preselected parameter. If we examine the contraction profile, there will always be an contraction number k such that $|R_T^k| \ge t_{seed}$ and $|R_T^{k+1}| < t_{seed}$. The region R_T^k is selected as the *seed region* (or kernel region) for subsequent surface fitting and region growing. The circles in Fig. 4 indicate the size of the selected seed region. The threshold t_{seed} must always be greater than or equal to the minimum number of points required for the simplest surface fit (i.e., 3 points for a plane).

The fundamental purpose of the contraction profile computation for seed region extraction is to find a small enough isolated region that 1) is not inadvertently connected to any separate, but adjacent surface regions, and 2) is far enough inside the boundaries of the actual surface primitive to provide good surface fitting. The 3×3 erosion operation (i.e., zero out pixels that have zero-valued neighbors and leave other pixels alone) is a simple, common image processing operation that can be accomplished in less than a video frame time on existing image processing hardware. Other methods for obtaining seed re-

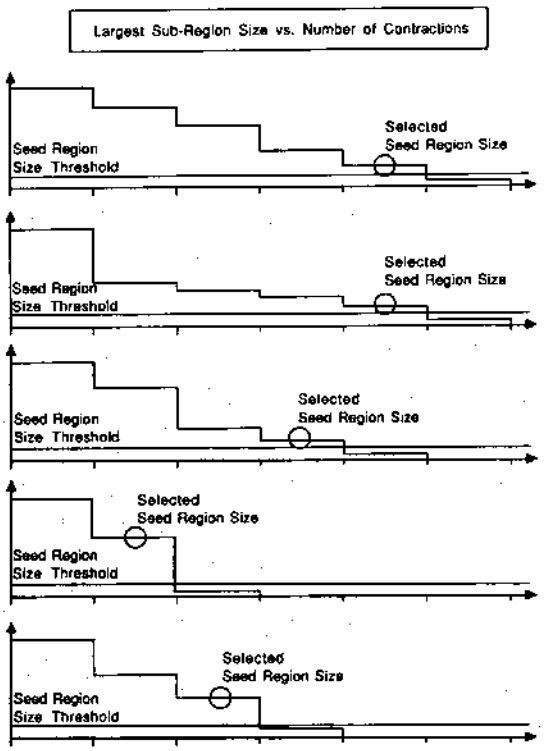


Fig. 4. Contraction profiles for five surface type regions.

gions are possible, but this method is simple and potentially very fast.

IX. Iterative Variable Order Surface Fitting

A plane is always fitted first to the small seed region using equally weighted least squares. If the seed region belongs to a surface that is not extremely curved, a plane will fit quite well to the digital surface defined by the original image data. If the plane fits the seed region within the maximum allowable RMS error threshold $\epsilon_{\max} = w_1\sigma_{\text{img}}$, then the seed is allowed to grow. The value of w_1 was empirically set to 2.5 in our experiments to allow for variations in the image noise process. If not, the seed is fitted with the next higher-order surface (e.g., biquadratic), and the algorithm proceeds similarly. When the seed is allowed to grow, the functional description of the surface over the seed region is tested over the entire image to determine what pixels are compatible with the seed region as described in the next section.

This surface fitting process may be stated more precisely as follows. Let I be the rectangular image region over which a hypothetical piecewise smooth function $z = g(x, y)$ is defined. Let $\hat{R}_i^{k=0}$ denote the seed region provided by the seed extraction algorithm that is assumed to be contained in the unknown actual region R_i in the image: $\hat{R}_i^{k=0} \subseteq R_i \subseteq I$. The seed region $\hat{R}_i^{k=0}$ must be converted to a full region description $\hat{R}_i$ that approximates the desired region description R_i.

Now, let $\vec{a}_i^k$ be the parameter vector associated with the functional fit to the pixel values in the given region $\hat{R}_i^k$ of the kth iterative surface fit. Let $\Re^n$ denote the set of all parameter vectors for the set of approximating functions F, and let $|F|$ be the number of different types of surface functions to be used. A particular function type (or fit order) is referred to as m^k where $1 \le m^k \le |F|$. The general fitting function of type m^k is denoted $z = \hat{f}(m^k, \vec{a}_i^k; x, y)$. The general surface fitting process, denoted L_f, maps the original image data $\tilde{g}(x, y)$, a connected region definition $\hat{R}_i^k$, and the current fit order m^k into the range space $\Re^n \times \Re^+$ where $\Re^+$ is the set of possible errors (nonnegative real numbers):

$$(\vec{a}_i^k, \epsilon_i^k) = L_f(m^k, \hat{R}_i^k, \tilde{g}) \tag{25}$$

and has the property that the error metric

$$\epsilon_i^k = \left\| \hat{f}(m^k, \vec{a}_i^k; x, y) - \tilde{g}(x, y) \right\|_{\hat{R}_i^k} \tag{26}$$

is the minimum value attainable for all functions of the form specified by m^k. Equally weighted least-squares surface fitting minimizes the error metric

$$(\epsilon_i^k)^2 = \frac{1}{|\hat{R}_i^k|} \sum_{(x,y)\in\hat{R}_i^k} \left(\hat{f}(m^k, \vec{a}_i^k; x, y) - \tilde{g}(x, y)\right)^2 \tag{27}$$

where $|\hat{R}_i^k|$ is the number of pixels in the region $\hat{R}_i^k$ (the area of the region). The parameter vector $\vec{a}_i^k$ and the surface fit order m^k are passed onto the region growing procedure if the RMS fit error test and the regions test are passed. Otherwise, m^k is incremented and the higher order surface is fitted. If all four fit orders were tried and the error was never less than the threshold $\epsilon_{\max}$, the seed region is rejected by marking off the pixels in the surface type label image, and then continuing by looking for the next largest connected region of any surface type.

A. RMS Fit Error Test

The RMS fit error test tests the surface fit error, which measures the variance of the error of the fit due to the noise in the data, against the maximum allowable fit error as determined from the noise variance estimate for the image: $\epsilon_i^k < \epsilon_{\max} = w_1\sigma_{\text{img}}$. If the error is small enough, the surface fit passes the test; otherwise, it fails. The coefficient $w_1 = 2.5$ is an empirically determined parameter.

B. Regions Test

The regions test is required because it is possible for a lower order function to fit a higher order function over a finite region within the maximum allowable fit error threshold even though the lower order fit is not appropriate. It is possible to detect the presence of a higher order function in the data (without letting the fit error increase all the way up to the error threshold) by analyzing the distribution of the sign of the fit errors (residual errors) at each individual pixel of the fit. We have generalized the runs test of nonparametric statistics [28] to assist in the detection of higher order behavior. This test is discussed in detail in [8], and is summarized here.

Consider that three long residual-sign intervals occur when fitting a line to a slowly bending curve as in Fig. 5. Fitting a plane to a small portion of a sphere is very similar except that two large residual-sign regions occur as is

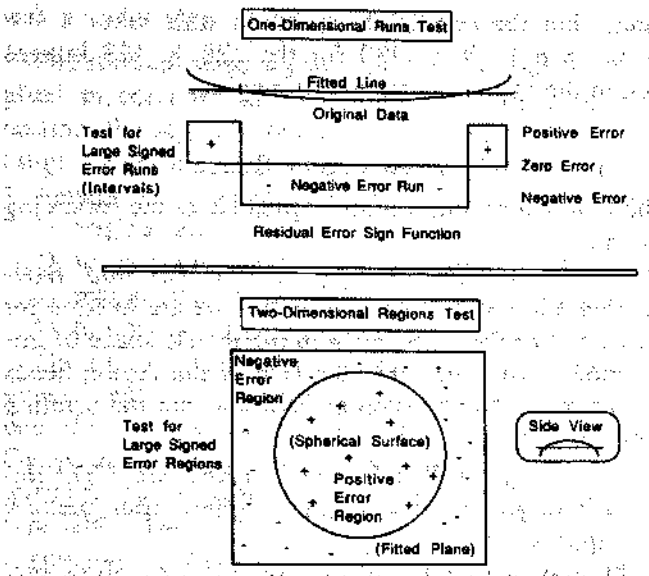


Fig. 5. Runs test and regions test ideas for noiseless data examples.

also shown in Fig. 5. The regions test is performed as follows: 1) For each original pixel value in the region $\hat{R}_i^k$ lying *above* the fitted surface $\hat{f}(m^k, a_i^k; x, y)$, turn a pixel *on* in a *positive residual error* image; for each pixel below the surface, turn a pixel on in a negative error image (all pixels initially off in both images). 2) Perform one 3×3 erosion on each error image and count the pixels left in the largest connected region in each image. 3) If either count is greater than r_t percent of the size of the current region $|\hat{R}_i^k|$, then increase the fit order m^k. The experimental results used a regions test threshold of $r_t = 0.9 + 0.2\sigma_{\text{img}}$ percent, which was determined empirically.

X. Region Growing

After a surface of order m^k is fitted to the region $\hat{R}_i^k$ in the kth iteration, the surface description is used to grow the region into a larger region where all pixels in the larger region are connected to the original region and compatible with the approximating surface function for the original region. The parallel region growing algorithm accepts as input the original digital surface $\tilde{g}(x, y)$, the approximating function $\hat{f}(m^k, \vec{a}_i^k; x, y)$ and the surface fit error ϵ_i^k from the surface fitting algorithm. It does not use the region definition until later. To determine the zeroth-order "surface continuity" compatibility of each pixel $p \in I$ with the approximating surface description, the polynomial based prediction for the pixel value and the actual pixel value

$$\hat{z}(p) = \hat{f}(m^k, \vec{a}_i^k; x(p), y(p)) \quad \text{and}$$
$$z(p) = \tilde{g}(x(p), y(p)) \tag{28}$$

are compared to see if the pixel p is compatible with the approximating surface function. If the magnitude of the difference between the function value and the digital surface value is less than the allowed tolerance value, denoted $w_0\epsilon_i^k$, then the pixel p is added to the set of compatible pixels, denoted $C(m^k, \vec{a}_i^k, \epsilon_i^k)$, which are compatible with the surface fit to the region $\hat{R}_i^k$. Otherwise, the pixel is incompatible and discarded. The result of this process is the compatible pixel list:

$$C(m^k, \vec{a}_i^k, \epsilon_i^k) = \left\{ p \in I : \left| \hat{z}(p) - z(p) \right| \leq w_0 \epsilon_i^k \right\}. \tag{29}$$

This set of compatible pixels $C(\cdot)$ is essentially a thresholded absolute value image of the difference between the original image data and the image created by evaluating the function $\hat{f}$ at each pixel. For our experimental results, the factor $w_0 = 2.8$ was used. This ensures that approximately 99.5 percent of all samples of a smooth surface corrupted by normally distributed measurement noise will lie within this error tolerance. This factor has been found to work well in the presence of other types of noise also.

The compatible pixel list is then post processed to remove any pixels that do not possess "surface normal continuity" compatibility with the approximating surface. Let $\tilde{g}_u(p)$ and $\tilde{g}_v(p)$ denote the first partial derivative estimates of the local surface as computed from the image data at the pixel p via convolutions as mentioned earlier. Let $\hat{g}_u(p)$ and $\hat{g}_v(p)$ denote the first partial derivatives of the approximating surface as computed from the polynomial coefficients at the pixel p. Let $\tilde{n}$ be the unit normal vector as determined by the data, and let $\hat{n}$ be the unit normal vector as determined by the approximating surface:

$$\tilde{n} = \frac{[-\tilde{g}_u \ -\tilde{g}_v \ 1]^T}{\sqrt{1 + \tilde{g}_u^2 + \tilde{g}_v^2}} \qquad \hat{n} = \frac{[-\hat{g}_u \ -\hat{g}_v \ 1]^T}{\sqrt{1 + \hat{g}_u^2 + \hat{g}_v^2}}. \tag{30}$$

A pixel is compatible in the sense of surface normal continuity if the angle between the two unit normals is less than some threshold angle θ_t:

$$\cos^{-1}(\hat{n} \cdot \tilde{n}) \leq \theta_t. \tag{31}$$

For our experimental results, the threshold angle is given by $\theta_t = 12 + 16\sigma_{\text{img}}$ degrees, where the coefficients were determined empirically. The test may be rewritten in the following form to avoid square roots and to incorporate the derivative values directly:

$$\frac{(\tilde{g}_u - \hat{g}_u)^2 + (\tilde{g}_v - \hat{g}_v)^2 + (\tilde{g}_u\hat{g}_v - \tilde{g}_v\hat{g}_u)^2}{(1 + \tilde{g}_u^2 + \tilde{g}_v^2)(1 + \hat{g}_u^2 + \hat{g}_v^2)} \leq \sin^2(\theta_t). \tag{32}$$

Since the compatibility test for surface normal continuity involves many computations per pixel, it is only applied to those pixels that have passed the compatibility test for surface continuity. Excellent segmentation results have been obtained without the surface normal continuity test on many images that lack small orientation discontinuities. However, a data-driven smooth-surface segmentation algorithm must always perform the test to ensure that growing regions do not inadvertently grow over small or noisy orientation discontinuities.

A. Region Iteration

When the parallel region growing computation has operated on every pixel, the compatible pixel list $C(m^k, \vec{a}_l^k, \epsilon_l^k) \subseteq I$ is complete. The largest connected region in this set of pixels that overlaps the seed region $\hat{R}_l^k$ must then be extracted to create the next region $\hat{R}_l^{k+1}$. This process is denoted $\Lambda(\cdot)$. The output region $\hat{R}_l^{k+1}$ must have the property that it is the largest connected region in the list of compatible pixels satisfying

$$\hat{R}_l^k \cap \hat{R}_l^{k+1} \neq \phi = \text{Null Set} \tag{33}$$

because it is possible to get larger connected regions in the compatible pixel list than the connected region corresponding to the seed region. The iterative process of region definition via largest, overlapping, connected region extraction may be expressed as follows:

$$\hat{R}_l^{k+1} = \Lambda\big(C(m^k, \vec{a}_l^k, \epsilon_l^k), \hat{R}_l^k\big) = \Phi(\hat{R}_l^k) \tag{34}$$

where $\Phi(\cdot)$ represents all operations required to compute the region $\hat{R}_l^{k+1}$ from the region $\hat{R}_l^k$. It is interesting to note that since the regions of an image form a metric space [8], the desired solution region is a fixed point $R = \Phi(R)$ of the mapping $\Phi(\cdot)$.

The new region is then considered as a seed region and processed by the surface fitting algorithm

$$(\vec{a}_l^{k+1}, \epsilon_l^{k+1}) = L_f(m^{k+1}, \hat{R}_l^{k+1}, \hat{g}) \tag{35}$$

to obtain a new parameter vector and a new surface fit error. If this region is allowed to grow again $\epsilon_l^{k+1} < \epsilon_{\max}$, then the compatible pixel list is recomputed $C(m^{k+1}, \vec{a}_l^{k+1}, \epsilon_l^{k+1})$, the largest connected overlapping region of $C(\cdot)$ is extracted, and so on until the termination criteria are met.

B. Sequential Versus Parallel Region Growing

The region growing process is formulated above for a parallel implementation where bivariate polynomials are evaluated over images and regions. It must be noted that this simple, parallel region growing formulation is equivalent to more complicated, sequential, spiraling region growing approaches *until the last iteration*. At the last iteration, the processing of the compatible pixel list becomes an important feature of the segmentation algorithm. After the growing region has been accepted, any other sufficiently large, reasonably shaped regions in the compatible pixel list are also accepted as part of the same surface. For example, in the coffee cup range image shown in the experimental results section, the flat background visible through the handle of the cup is correctly assigned to the larger background surface without high-level knowledge, only the surface compatibility concepts. Thus, nonadjacent compatible regions can be labeled as such during the surface acceptance stage without further postprocessing operations because of the parallel region growing process during the last iteration prior to acceptance. On a sequential machine, sequential region growing methods can offer faster performance for the other iterations, but the parallel formulation only takes a few seconds on a VAX 11/780 for the 128 × 128 images shown in the experimental results section.

XI. Termination Rules

The termination criteria are expressed as the following set of rules:

1) IF $|\hat{R}_l^k| \approx |\hat{R}_l^j|$ for any $j < k$, THEN stop! Basically, this rule states the condition that we are looking for a fixed point of the mapping Φ in the metric space of image regions. Note that only the size of the region needs to be checked from iteration to iteration, not the detailed region description.

2) IF $\epsilon_l^k > \epsilon_{\max}$ AND $m^k \geq |F|$, THEN stop! The image data is varying in a way that the highest order function cannot approximate.

3) At least two iterations are required for a given surface fit order m^k before the algorithm is allowed to stop.

These rules state the essential concepts involved in terminating the surface fitting iteration. There is also a maximum limit on the number of possible iterations to prevent extremely long iterations. In all tests done to this point, the maximum limit of 30 iterations has never been reached and the average number of iterations is approximately eight.

XII. Surface Acceptance and Rejection Decisions

After the surface growing iterations have terminated, we are left with the set of compatible pixels and the connected surface region itself along with the function parameters and the fit error. For growth surface regions that exceed the error threshold $\epsilon_{\max}$, but not by much, an acceptance zone is defined above the error threshold such that surface regions within the acceptance zone are accepted. The acceptance threshold used for our experiments is 50 percent greater than $\epsilon_{\max} = w_1\sigma_{\text{img}}$ where $w_1 = 2.5$. Surface regions with fit errors beyond the acceptance zone are rejected.

When a surface region is rejected for any reason, the seed region responsible for the surface region is marked off in the surface type label image as having been processed, which prohibits the use of the same original seed region again. When a surface region is accepted, all pixels in that region are similarly marked off in the surface type label image so that they are not considered for future seed regions. In this respect, surface rejection and surface acceptance are similar. However, the surface acceptance process also updates the region label image, the reconstruction image, and the error image. In addition, the acceptance process dilates the accepted region description and checks if there are any connected groups of pixels in that dilated region that are surface-continuity compatible with the accepted surface and connected with the accepted region. Surface-normal compatibility is not required when adding these pixels because of the difficulty in getting accurate surface normal estimates near surface region boundaries.

XIII. Experimental Results

The surface-based segmentation algorithm has been applied successfully to more than 40 test images. In this section, the segmentation algorithm's performance on six range images and three intensity images is discussed. The following set of images is displayed for each input image:

1) original gray scale image (upper left).
2) surface type label image (lower left).
3) region label image segmentation plot (lower right).
4) reconstructed gray scale image (upper right).

The surface type label image shows the coarse "initial guess" segmentation provided by labeling each pixel with one of eight labels according to the sign of the mean and Gaussian curvature. Each region in this image is an isolated set of connected pixels that all have the same surface type label. The region label image shows the final refined segmentation obtained from the iterative region-growing algorithm. Each region in this image is the support region over which a particular polynomial surface function is evaluated. The reconstructed image is computed from the region label image and the list of surface parameters, and it shows the visual quality of the approximate surface representation. For each image, we also include the noise variance estimate computed from the orginal image and an error statistic computed from the original-reconstruction difference image.

When a user runs the program on an image, the name of the image is typically the only input required by the program. All internal parameters are either fixed or automatically varying based on the noise variance estimate. The user does have the option to change five of the fixed internal parameters and to override the three automatically set thresholds: 1) the maximum allowed RMS fit error ϵ_{max}, 2) the surface normal compatibility angle threshold θ_t, and 3) the regions test threshold r_t. Eight of the nine images shown here were obtained without any adjustments whatsoever, but more interesting results were obtained by overriding the automatically set thresholds for the computer keyboard range image, which has nonstationary noise. This was necessary because of the stationarity assumption of the current noise variance estimation algorithm, which allows us to describe the image noise with a single number.

A. Interpretation of Intensity Image Results

The entire segmentation algorithm is based only on the knowledge of piecewise-smooth surfaces and digital surfaces. Since intensity images are also digital surfaces, the algorithm can be applied to intensity images for segmentation purposes. It is important to understand that the dimensionality of a digital surface is the same regardless of the meaning of the sensed values at each pixel. And since the difference between range images and intensity images is the interpretation of the sensed values (depth versus light intensity), the difference in the algorithm output lies in how the surface segmentation is interpreted. Intensity image surface primitives are only surface function approximations to the intensity image data and nothing more. The segmentation results will be useful when intensity image surfaces correspond to physical surfaces in a scene. This is of course equivalent to an implicit assumption that edge detection approaches use: the boundaries of intensity image surfaces correspond to the boundaries of physical surfaces in a scene. However, our image description is much richer than most edge-based image descriptions because not only are guaranteed closed-curve edges of regions detected, but the *approximate value of every single image pixel* is encoded in the polynomial coefficients. If intensity image surface primitives can be reliably extracted, it is possible to apply shape from shading ideas [58] to intensity surface primitives [14].

B. Coffee Cup Range Image (ERIM)

The coffee cup range image is a 128 × 128 8-bit image from an ERIM phase-differencing range sensor [108]. The segmentation results are shown in Fig. 6. The measured noise variance is $\sigma_{img} = 1.02$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i, j)|) = 1.46$. The final segmentation clearly delineates the outside cylindrical surface of the cup, the foot of the cup, the inside cylindrical surface of the cup, the background table surface (which was recognized as a single surface with three subregions despite the nonadjacency of the region visible through the handle and the small hole in the side of the cup), and the cup handle surface (which is represented as two surfaces due to the twisting of the surface from this view). Although this image is easy to segment by many other methods, the subtle difference in surface variations between the foot and the main body of the cup is difficult to detect with an edge detector. Two small meaningless surfaces did arise on the steeply sloped sides of the cup because the laser range sensor has difficulty obtaining good results when most of the laser energy is reflected away from the sensor. Note that although this algorithm knows nothing about *cylinders*, the cylindrical surface of the cup is adequately segmented.

Fig. 7 shows discrete contour lines for the original image (left) and the reconstructed image (right). These contour lines bound regions of constant range. This presentation is needed to adequately appreciate the shape information in the noiseless reconstructed image as compared to the noisy original image. The background appears to be curved due to image parameterization distortions caused by the range sensor's two orthogonal axis mirrors and equal angle increment sampling as discussed in the Appendix. Fig. 8 shows the variations in RMS fit error, region size, and surface fit order as a function of the region growing iteration number for the background surface. Fig. 9 shows the actual polynomial coefficients used in the image reconstruction for the six primary regions: 1) background, 2) cup body, 3) cup interior, 4) top of handle, 5) bottom of handle, and 6) foot of cup. The mean absolute error (e1), the standard deviation (e2), and

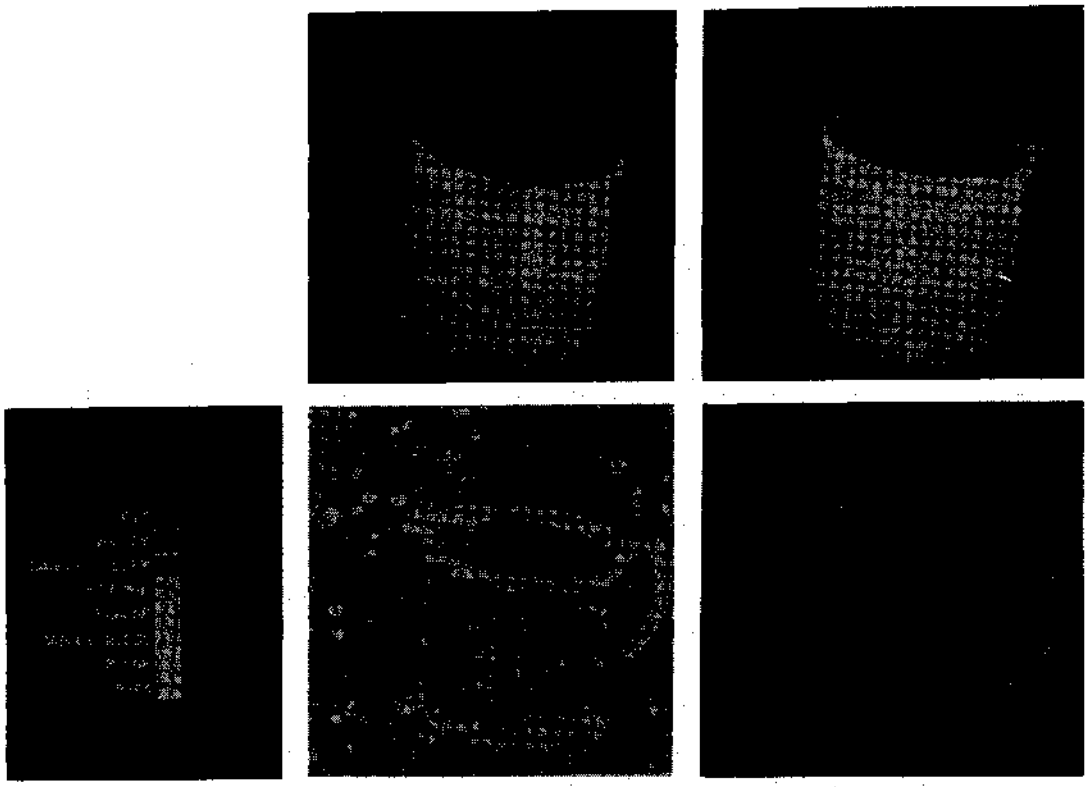


Fig. 6. Segmentation results for coffee cup range image.

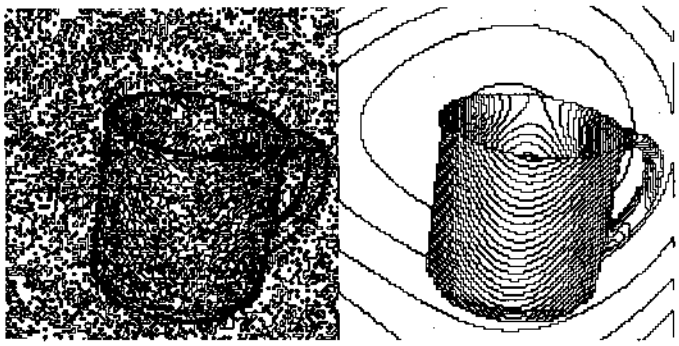


Fig. 7. Range contour lines for original and reconstructed images.

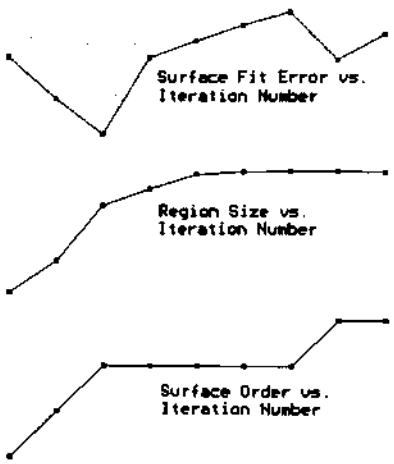


Fig. 8. Fit error, region size, and surface order versus iteration number.

```
Polynomial Graph Surfaces for '/img/range/cofcup2'  (128 x 128 Image)
                2   2   2     2   3   3   3     2 2     3   4   4
z=a+bx+cy+dxy+ex +fy +gx y+hxy +ix +jy +kx y+lx y +mxy +nx +oy

Surf# 1 Biquartic Surface
a= 41.1084863 b= 0.141577803 c= 0.2713745965
d= -0.002495014122 e= -0.00077555763671 f= -0.003601368877
g= 1.481712253e-06 h= 2.726318379e-05 i= -2.309573368e-06
j= 8.758485509e-08 k= -1.803061683e-08 l= -8.351990453e-08
m= -8.546141676e-09 n= -5.871564019e-10 o= 1.594933765e-08
0.733465 0.890638 2.81471 (e1,e2,emax) 9321 Pxls in Rgn

Surf# 2  Bicubic Surface
a= 64.72905408 b= 4.004568601 c= -0.8523437561
d= -0.008861604524 e= -0.02260194283 f= 0.008512778066
g= -4.525700906e-05 h= 4.449644712e-05 i= 7.129137939e-06
j=-5.132800061e-05
0.894967 1.12435 4.17196 (e1,e2,emax) 3301 Pxls in Rgn

Surf# 3  Biquartic Surface
a= -183.4152524 b= -5.291701634 c= 35.73599936
d= -0.822060872 e= 0.303405367 f= -0.4267766869
g= 0.000534934843 h= 0.01667119329 i= -0.002583054026
j= -0.003845895577 k= 8.273502856e-06 l= -1.84330636e-05
m= -0.0001033402269 n= 7.022812258e-06 o= 6.765861748e-05
0.862173 1.04452 2.94707 (e1,e2,emax) 846 Pxls in Rgn

Surf# 4  Bicubic Surface
a= -56553.34566 b= 1085.103359 c= 550.9673791
d= -6.935330046 e= -7.005112872 f= -1.772838004
g= 0.02117382282 h= 0.01207493125 i= 0.01546967941 j=0.001442358458
0.693086 0.870873 1.94343 (e1,e2,emax) 85 Pxls in Rgn

Surf# 5  Biquartic Surface
a= 58478.67644 b= -4177.347605 c= 5553.461039
d= -62.08405574 e= 61.15735537 f= -78.3777114
g= 0.02145723219 h= 0.980323446 i= -0.3041497425
j= 0.2363115629 k= 0.001864292974 l= -0.005593080286
m= 0.001788868803 n= 0.0003022405187 o= -0.001835806151
1.57884 1.93 5.13262 (e1,e2,emax) 107 Pxls in Rgn

Surf# 6  Biquadratic Surface
a= 16.6160474 b= 4.368181058 c= 0.9102155505
d= 0.0009597935563 e= -0.03381975184 f= -0.01009901358
1.30028 1.51381 3.97869 (e1,e2,emax) 146 Pxls in Rgn
```

Fig. 9. Bivariate polynomial coefficients describing coffee cup surfaces.

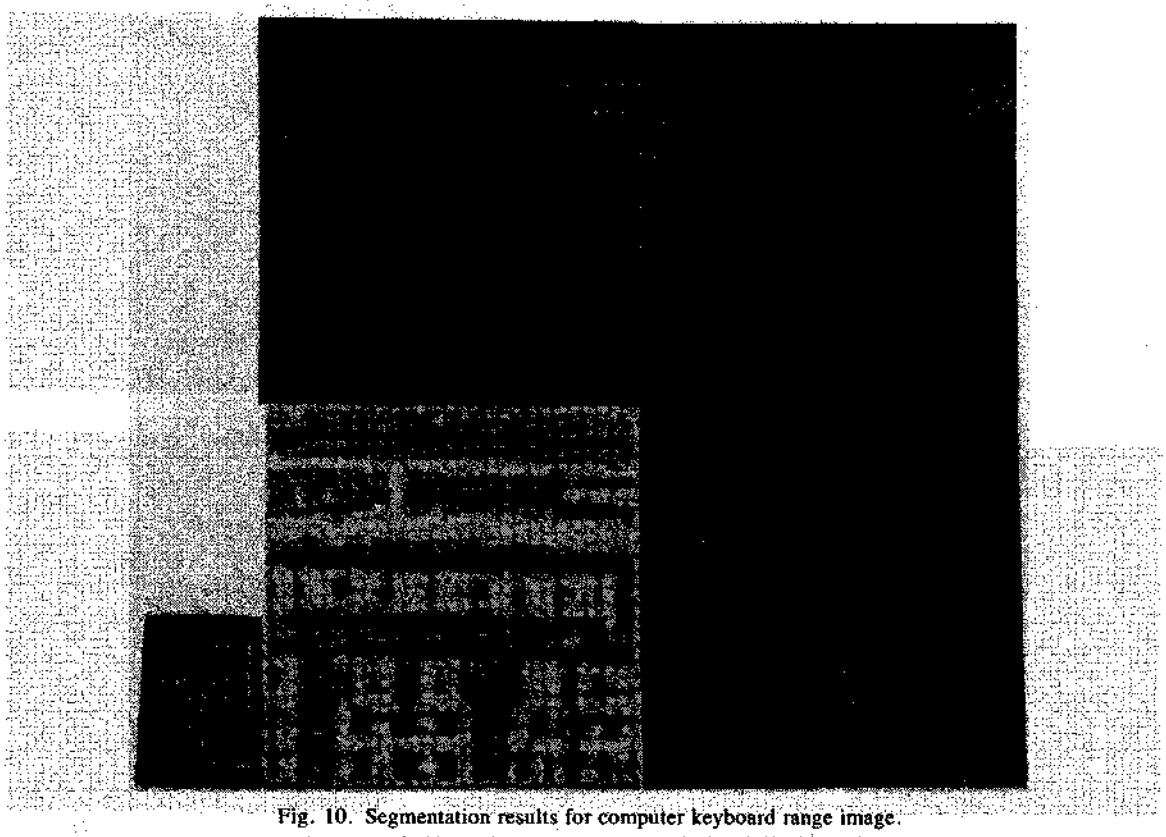


Fig. 10. Segmentation results for computer keyboard range image.

the maximum error (emax) are listed for the region as well as the number of pixels in the accepted region (statistical outliers not included).

C. Computer Keyboard Range Image (ERIM)

The computer keyboard range image is a 128 × 128 6-bit image from the ERIM range sensor. The segmentation results are shown in Fig. 10. The measured noise variance is $\sigma_{\text{img}} = 1.68$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i,j)|) = 1.96$. The surface type label image shows the uneven distribution in the additive noise field. The smooth surface of the keyboard body has very little noise in comparison with the noise on the keys themselves. This results from the specularity of the key surfaces and diffuseness of the keyboard body. This nonstationarity of the noise disobeyed the stationarity assumption of the noise estimation program and the automatically set thresholds did not provide the segmentation quality of other images. Therefore, manually set thresholds were used for the results shown here. Some individual keys have been segmented whereas other keys were grouped together. However, the center of each key is available from the surface type label image if needed. Although it cannot be seen in this presentation of results, each key center is represented as a small isolated pit or valley region surrounded by ridge and peak regions representing the surrounding parts of the key.

D. Ring on Steps (ERIM)

The ring on steps range image is a noisy 128 × 128 8-bit image from the ERIM range sensor. The segmentation results are shown in Fig. 11. The ring has a rectangular cross section and the step lower part of the steps is cut off at an oblique angle. The measured noise variance is $\sigma_{\text{img}} = 2.05$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i,j)|) = 3.31$. This image is the noisiest range image in results documented here. The steeply sloped surfaces are much noisier than the other surfaces as occurred in the coffee cup image. Fig. 12 shows the contour lines for the original image (left) and the reconstructed image (right). The noiseless quality of the reconstructed image is quite apparent in this presentation.

E. Auto Part (INRIA)

The original data for the auto part was acquired from the INRIA range sensor (made available courtesy of Prof. T. Henderson of Univ. of Utah and INRIA) and was formatted as a long list of (x, y, z) points. Although the data was easily divided into scan lines, a different number of points occurred on each scan line, and the points were not regularly spaced. This data was converted to 128 × 128 8-bit range image by a separate processing step not documented here. The segmentation results for this auto part range image are shown in Fig. 13. The measured noise variance is $\sigma_{\text{img}} = 0.60$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i,j)|) = 1.48$. This 2.5-D segmentation is similar to 3-D segmentations published in [33], [12], [52].

F. Cube with Three Holes

The cube with three holes drilled through it provides an interesting nonconvex combination of flat and cylindrical

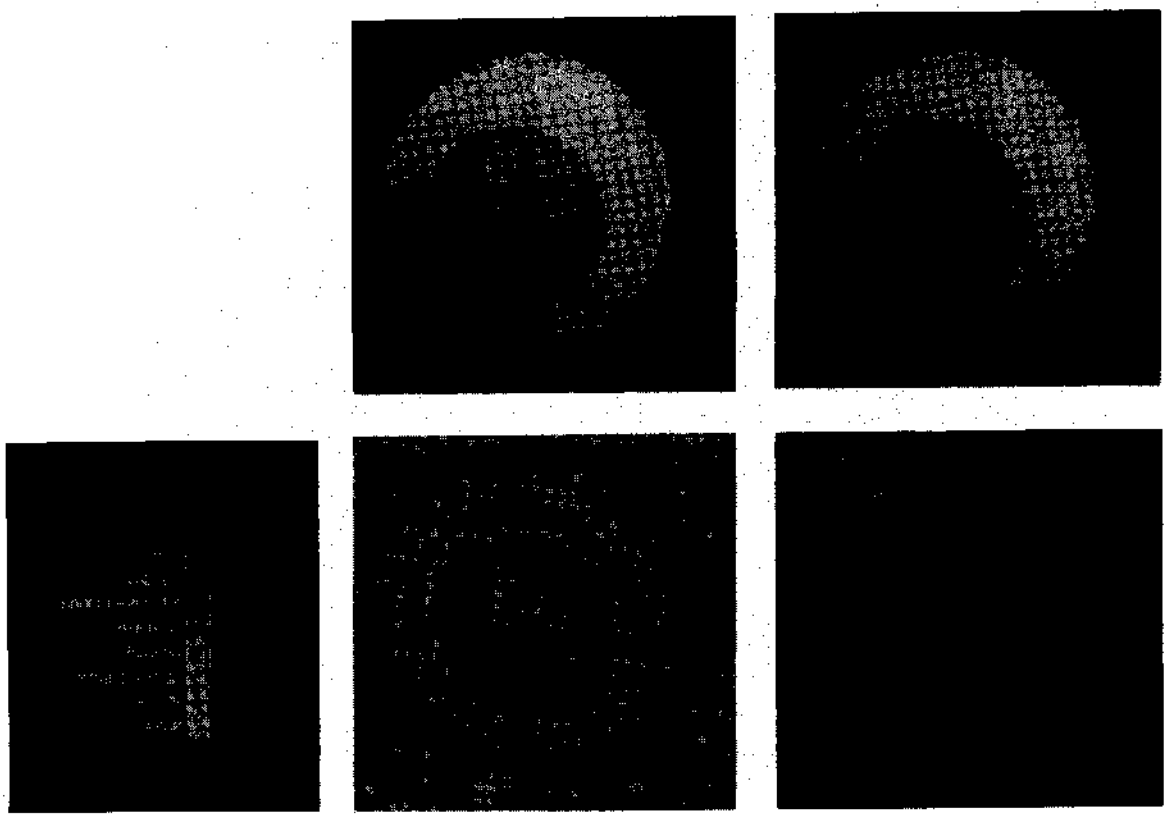
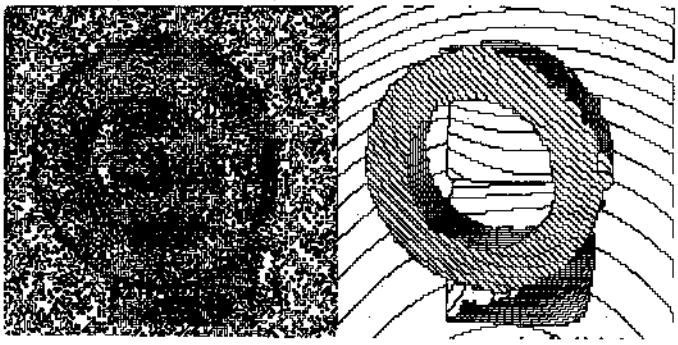


Fig. 11. Segmentation results for ring on steps range image.



/b/img/range/ringbloc

Fig. 12. Range contour lines for original and reconstructed images.

surfaces. This range image was created using a depth-buffer algorithm on a 3-D solid model created using SDRC/GEOMOD and adding pseudo-Gaussian noise. The segmentation results are shown in Fig. 14. The measured noise variance is $\sigma_{\text{img}} = 1.89$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i,j)|) = 2.94$. The three linear dihedral edges of the cube have been determined to subpixel precision by intersecting the planar descriptions for the three planes. The results here show the raw segmentation in the region label image.

G. Road Scene Range Image (ERIM)

The road scene range image is a 128 × 128 range image from the ERIM sensor. The segmentation results are shown in Fig. 15. The measured noise variance is $\sigma_{\text{img}} = 1.82$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i,j)|) = 0.96$. The edges of the road are clearly delineated in the segmentation results. The false edge crossing the road results from the limited bending capability of the biquartic polynomial within the tolerances specified by the automatic threshold setting mechanisms. This edge can be removed in several ways: 1) the error tolerances can be increased manually, 2) higher order surfaces can be used, or 3) the range data can be precorrected (resampled) to eliminate the geometric distortions produced by equal angle increment sampling in scanning laser radars that use two mirrors rotating around orthogonal axes as discussed in the Appendix.

H. Road Scene Intensity Image

A different road scene is represented in the 128 × 128 8-bit intensity image. The intensity image segmentation results are shown in Fig. 16. The measured noise variance is $\sigma_{\text{img}} = 2.27$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i,j)|) = 5.48$. The edges of the road are clearly delineated, and the quality of the image reconstruction is quite good. A faster version of the segmentation algorithm might be used for navigation by growing fixed image regions directly in front of the vehicle in both registered range and intensity images. The polynomial surface primitives will grow only over the image regions corresponding to the road. The complementary information in the range and intensity images can be combined to avoid obstacles and plan paths over smooth surfaces.

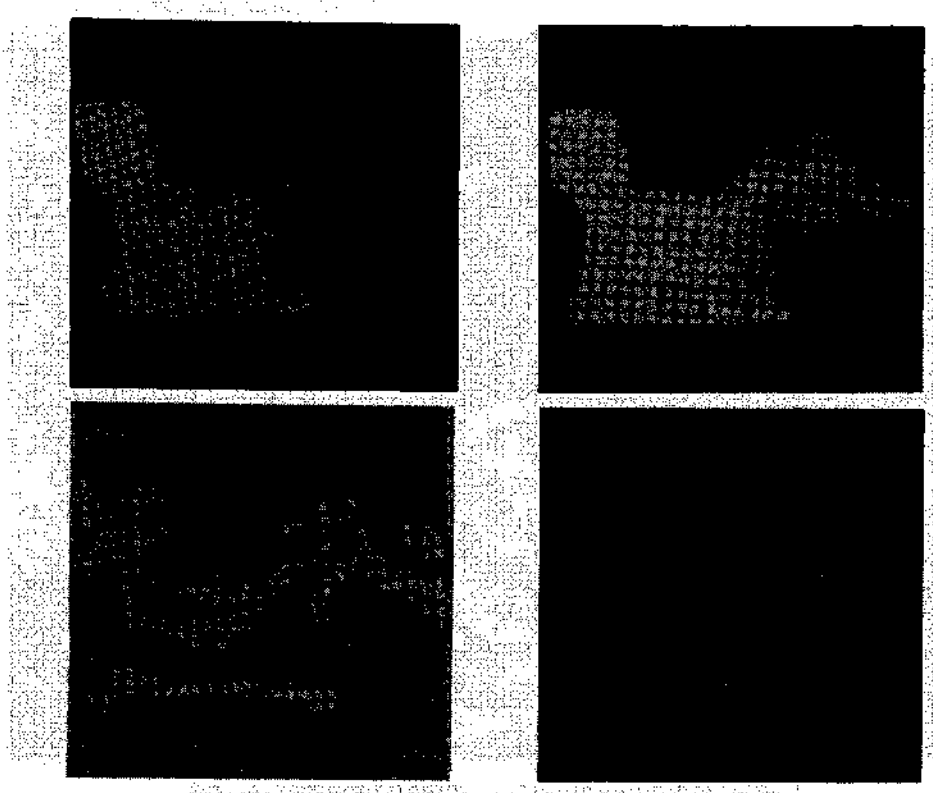


Fig. 13. Segmentation results for auto part range image.

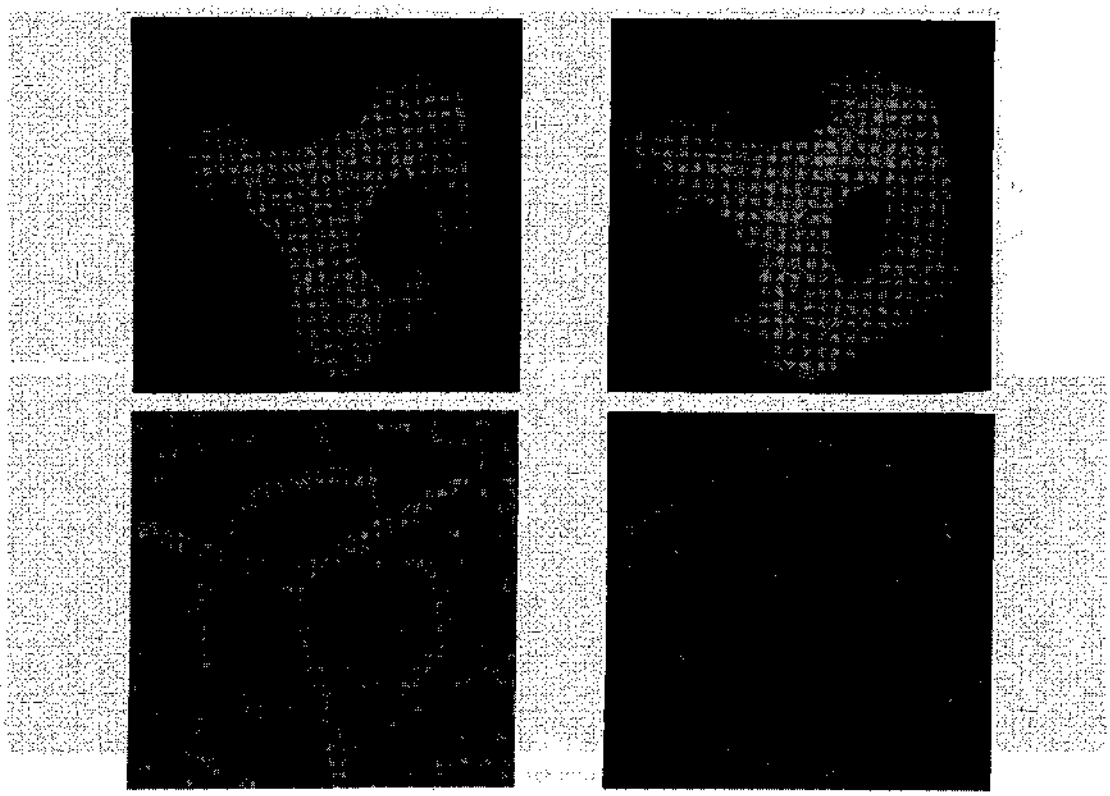


Fig. 14. Segmentation results for cube with three holes range image.

1. Space Shuttle Intensity Image

The segmentation results for an image of a space shuttle launch are shown in Fig. 17. The measured noise variance is $\sigma_{img} = 2.71$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i,j)|) = 4.32$. The reconstructed image lacks detail whenever the detail in the original image consists of only a few pixels (10 or less) or is only one pixel wide. For example, a small piece of the gantry tower is missing in the reconstructed image. The surface type label image segmentation appears completely incoherent when com-

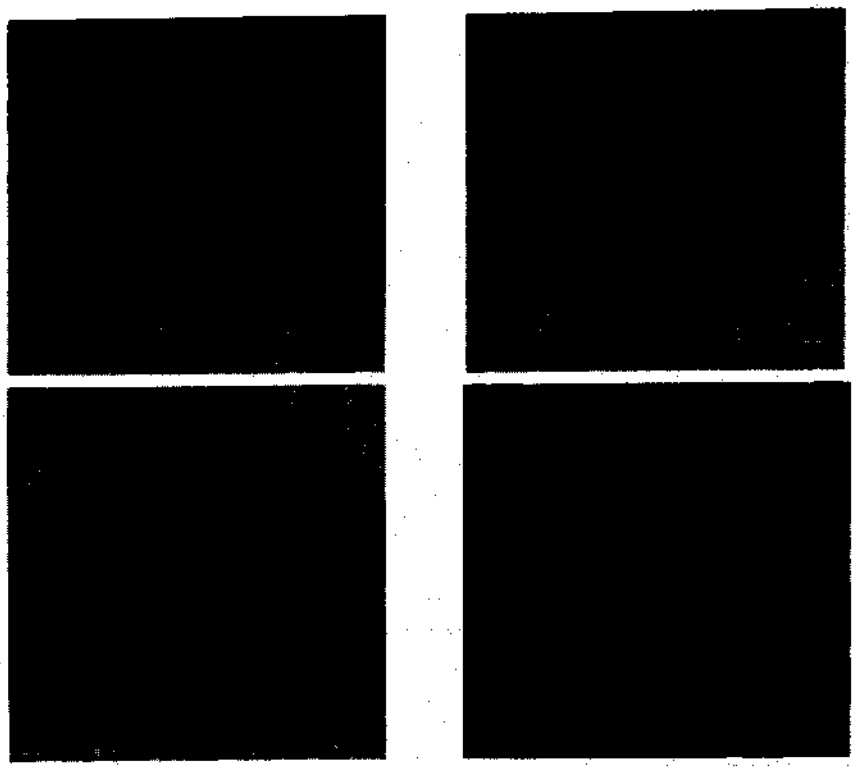


Fig. 15. Segmentation results for road scene range image.

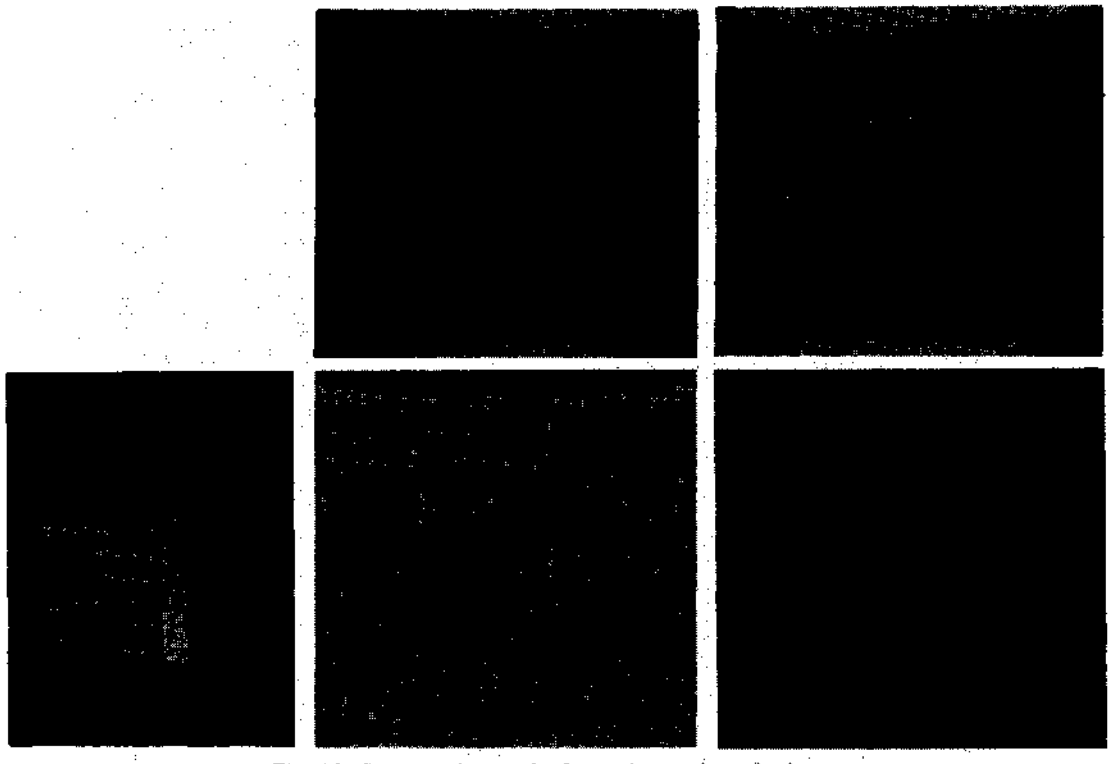


Fig. 16. Segmentation results for road scene intensity image.

pared to the original shuttle image. This is unlike most range images where some structure is usually perceivable. However, it still provided enough grouping information to the region growing algorithm to produce the final segmentation. The sky, the smoke clouds, the main tank, and the bright flames are isolated as intensity-image surface primitives.

J. House Scene Intensity Image (Univ. of Mass.)

A house scene, which has been segmented by many other techniques in the literature, was used to test the performance of the surface-based segmentation algorithm. The segmentation results for the 256 × 256 8-bit house scene image are shown in Fig. 18. Owing to the sequen-

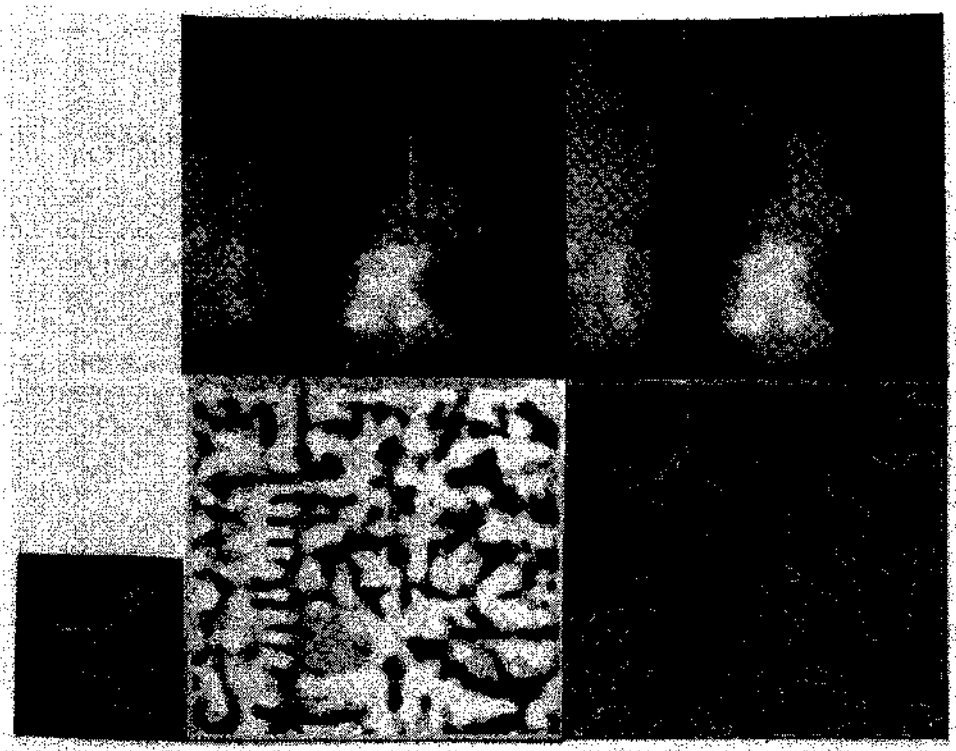


Fig. 17. Segmentation results for space shuttle intensity image.

Fig. 18. Segmentation results for house scene intensity image.

tial nature of the algorithm, segmentation processing time is related to image complexity, and several CPU hours on a VAX 11-780 were needed to compute these results consisting of 136 intensity surface primitives. The measured noise variance is $\sigma_{img} = 3.93$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i,j)|) = 9.44$. The sky, the roof of the house, the lawn, the garage door, the man's pants, the shutters, the side of the house, the chimney, and the trees are all well segmented. The tree regions are very textured, but are still adequately segmented because of the final processing of coalesced unexplained pixels. Note the smoothness of the tree region in the reconstructed image. The quality of the image reconstruction and the segmentation were obtained with the exact same set of internal parameters used for five of the range images. We know

of no other algorithm that can claim this type of segmentation performance on such a wide variety of images.

XIV. Conclusions and Future Directions

The experimental results obtained by applying the surface-based segmentation algorithm with a fixed set of input parameters to a large test database of over 40 images, including range and intensity images, indicates that data-driven segmentation of digital surfaces based on a piecewise-smooth surface model, surface curvature sign, and polynomial surface approximations is feasible and provides excellent results. Nine sets of image results are included here to document these claims.

This surface-based approach is very general in many respects. Flat surfaces are described explicitly as being flat, and arbitrary curved surfaces are described as being curved within the context of the same variable-order surface fitting algorithm. Most techniques in the literature need to handle flat and curved (quadric) surfaces as separate special cases. No *a priori* assumptions about surface convexity, surface symmetry, or object shape are used. The final segmentation/reconstruction description is driven by the content of the data, not by expected high-level models as is done in many other approaches. Moreover, the exact same algorithm with the exact same set of parameters is shown to be capable of segmenting range images and intensity images. We believe that any image that can be represented by a piecewise-smooth surface over sufficiently large regions (more than 10–30 pixels) can be segmented well by this algorithm.

The basic sign-of-curvature/iterative variable-order fitting approach is applicable to the segmentation of signals in any number of dimensions, not just scalar functions of two variables. The method is shown to be successful for edge interval segmentation in [8]. If one-pixel wide edges are available, x and y can be parameterized as a function of arc length yielding a 2-vector function of a single variable. Only three sign-of-curvature labels are needed for each 1-D function: concave up, concave down, and flat. In the future, we hope to be able to apply the algorithm to signals representing scalar functions of three variables, such as dynamic scenes and 3-D images from CAT scanners. In that case, 27 sign-of-curvature labels are needed, and approximating functions require many coefficients (20 for a tricubic).

Perception of surfaces plays a key role in image understanding. We have shown experimentally that the segmentation of range images into scene surfaces can be data-driven and need not involve higher level knowledge of objects. The perceptual organization capabilities of the surface-based image segmentation algorithm appear to also be worthwhile capabilities for intensity image segmentation as is shown via experimental results. More research is needed to determine how higher level knowledge should be used in relating intensity-image surface primitives to the real scene surfaces.

Better methods of noise estimation are needed to improve the automatic threshold selection process. For non-stationary noise, it may be necessary to store an estimate of the noise variance for each pixel or region in the image. The noise variance estimates must then be consulted during each iteration. We are also looking into various types of adaptive smoothing, such as in [98], to improve the quality of the partial derivative estimates used to compute surface curvature and check surface normal compatibility. Also, various applications will require different types of surface models. The shape description needs for NC machining may be quite different than those for surface shape matching in 3-D object recognition, and neither application may be able to use the extracted polynomial surface primitives in the current form described here. Thus, the conversion of the shape information into more useful forms for given applications is another key issue that must addressed.

Appendix
Equal Angle Increment Sampling

Formulas for calculating the geometric distortion introduced by equal angle increment sampling for range sensors with two orthogonal axis rotating mirrors and with spherical azimuth/elevation scanning mechanisms are included here. Let z, x, y be 3-D Cartesian coordinates with z representing depth from the x, y focal plane. Let r, θ, ϕ be 3-D orthogonal axis angular coordinates used for range sensors with two orthogonal axis mirrors. Let r, θ, ψ be 3-D spherical coordinates used for azimuth-elevation range sensors.

The transformations from orthogonal axis angular coordinates to Cartesian coordinates are given by the following:

$$x(r, \theta, \phi) = \frac{r \tan \theta}{\sqrt{1 + \tan^2 \theta + \tan^2 \phi}} \tag{36}$$

$$y(r, \theta, \phi) = \frac{r \tan \phi}{\sqrt{1 + \tan^2 \theta + \tan^2 \phi}} \tag{37}$$

$$z(r, \theta, \phi) = \frac{r}{\sqrt{1 + \tan^2 \theta + \tan^2 \phi}}. \tag{38}$$

Note the symmetry between the horizontal and vertical angles. The inverse transformations are given by

$$r(x, y, z) = \sqrt{x^2 + y^2 + z^2} \tag{39}$$

$$\theta(x, z) = \tan^{-1}\left(\frac{x}{z}\right) \tag{40}$$

$$\phi(y, z) = \tan^{-1}\left(\frac{y}{z}\right). \tag{41}$$

Note that horizontal angle θ is not a function of the vertical Cartesian coordinate y and that the vertical angle ϕ is not a function of the horizontal Cartesian coordinate x.

The spherical coordinate transformation from (x, y, z) to (r, θ, ϕ) coordinates is given by the following equations where ψ is the elevation angle and θ is the azimuth angle:

$$x(r, \theta, \psi) = r \cos \psi \sin \theta \quad (42)$$

$$y(r, \psi) = r \sin \psi \quad (43)$$

$$z(r, \theta, \psi) = r \cos \psi \cos \theta. \quad (44)$$

The inverse transformations for r and θ are identical to the orthogonal axis case, but the expression for the elevation angle is given by

$$\psi(x, y, z) = \tan^{-1}\left(\frac{y}{\sqrt{x^2 + z^2}}\right). \quad (45)$$

Note that ψ depends also on x in addition to y and z. Hence, the only difference between orthogonal axis angular coordinates and spherical coordinates is in the vertical angles ϕ and ψ.

The "warping" of surfaces in image coordinates by equal angle increment sampling in θ and ϕ or ψ, which was mentioned in the text, can be understood by comparing the depth expression for Cartesian coordinates to the depth expressions for orthogonal axis angular coordinates and spherical coordinates:

$$z(x, y) = \sqrt{\left(r(x, y)\right)^2 - (x^2 + y^2)} \quad (46)$$

$$z(\theta, \phi) = \frac{r(\theta, \phi)}{\sqrt{1 + \tan^2 \theta + \tan^2 \phi}} \quad (47)$$

$$z(\theta, \psi) = r(\theta, \psi) \cos \theta \cos \psi. \quad (48)$$

Flat surfaces in $z(x, y)$ data will appear curved in $z(\theta, \phi)$ data or $z(\theta, \psi)$ data because of the differences in surface parameterization. If given range images from an orthogonal axis coordinate $z(\theta, \phi)$ range sensor or spherical coordinate $z(\theta, \psi)$ range sensor, the Cartesian x and y coordinates can be computed for each angle pair $((\theta, \phi)$ for orthogonal axis angular coordinates or (θ, ψ) for spherical coordinates):

$$\begin{aligned} x_{\text{ortho}}(\theta, \phi) &= z(\theta, \phi) \tan \theta \\ x_{\text{spher}}(\theta, \psi) &= z(\theta, \psi) \tan \theta \end{aligned} \quad (49)$$

$$y_{\text{ortho}}(\theta, \phi) = z(\theta, \phi) \tan \phi$$

$$y_{\text{spher}}(\theta, \psi) = \sqrt{z^2(\theta, \psi) + x^2_{\text{spher}}(\theta, \psi)} \tan \psi. \quad (50)$$

The "difficulty" with these x, y coordinates, from an image processing viewpoint, is that they do not lie on an equally spaced grid of image pixels. If desired, interpolation can be used to resample the surface data to obtain an equally spaced sampled Cartesian orthographic projection $z(x, y)$ range image, but this is not necessary in many cases. Since most of the range images in this paper use a relatively small field of view, the range images can be segmented and approximate surface shape can be reconstructed directly without resampling. Once the appropriate image regions have been segmented, accurate physical surface shape in Cartesian coordinates can be computed (if all range sensor parameters are known) by computing the Cartesian x, y, z coordinates from the angular coordinates at each pixel in the segmented image regions and then fitting new graph surfaces to the Cartesian data.

Acknowledgment

The authors would like to thank L. Watson, L. Maloney, B. Haralick, D. Chen, R. Sarraga, and the reviewers for their helpful observations and suggestions. We also thank the Environmental Research Institute of Michigan (ERIM), Structural Dynamics Research Corporation, and General Motors Research Labs.

References

[1] G. J. Agin and T. O. Binford, "Computer description of curved objects," in *Proc. 3rd Int. Joint Conf. Artificial Intelligence*, Stanford, CA, Aug. 20–23, 1973, pp. 629–640.

[2] R. L. Anderson and E. E. Houseman, *Tables of Orthogonal Polynomial Values Extended to N = 104*, Iowa State College Agricultural and Mechanic Arts, Ames, IA, Res. Bull. 297, Apr. 1942.

[3] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[4] S. Barnard, "A stochastic approach to stereo vision," in *Proc. 5th Nat. Conf. Artificial Intelligence*, AAAI, Philadelphia, PA, August 11–15, 1986, pp. 676–680.

[5] R. H. Bartels and J. J. Jezioranski, "Least-squares fitting using orthogonal multinomials," *ACM Trans. Math. Software*, vol. 11, no. 3, pp. 201–217, Sept. 1985.

[6] P. R. Beaudet, "Rotationally invariant image operators," in *Proc. 4th Int. Conf. Pattern Recognition*, Kyoto, Japan, Nov. 7–10, 1978, pp. 579–583.

[7] G. Beheim and K. Fritsch, "Range finding using frequency-modulated laser diode," *Appl. Opt.*, vol. 25, no. 9, pp. 1439–1442, May 1986.

[8] P. J. Besl, "Surfaces in early range image understanding," Ph.D. dissertation, Dep. Elec. Eng. Comput. Sci., Univ. Michigan, Ann Arbor, Rep. RSD-TR-10-86, Mar. 1986; see also *Surfaces in Range Image Understanding*. New York: Springer-Verlag, 1988.

[9] P. J. Besl and R. C. Jain, "Three dimensional object recognition," *ACM Comput. Surveys*, vol. 17, no. 1, pp. 75–145, Mar. 1985.

[10] ——, "Invariant surface characteristics for three dimensional object recognition in range images," *Comput. Vision, Graphics, Image Processing*, vol. 33, no. 1, pp. 33–80, Jan. 1986.

[11] P. J. Besl, E. J. Delp, and R. C. Jain, "Automatic visual solder joint inspection," *IEEE J. Robotics Automation*, vol. RA-1, no. 1, pp. 42–56, May 1985.

[12] B. Bhanu, "Representation and shape matching of 3-D objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 3, pp. 340–350, May 1984.

[13] B. Bhanu, S. Lee, C. C. Ho, and T. Henderson, "Range data processing: Representation of surfaces by edges," in *Proc. Int. Pattern Recognition Conf.*, IAPR-IEEE, Oct. 1986, pp. 236–238.

[14] R. M. Bolle and D. B. Cooper, "Bayesian recognition of local 3-D shape by approximating image intensity functions with quadric polynomials," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 4, pp. 418–429, July 1984.

[15] R. C. Bolles and M. A. Fischler, "A RANSAC-based approach to model fitting and its application to finding cylinders in range data," in *Proc. 7th Int. Joint Conf. Artificial Intelligence*, Vancouver, B.C., Canada, Aug. 24–28, 1981, pp. 637–643.

[16] R. C. Bolles and P. Horaud, "3DPO: A three-dimensional part orientation system," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 3–26, Fall 1986.

[17] B. A. Boyter, "Three-dimensional matching using range data," in *Proc. 1st Conf. Artificial Intelligence Applications*, IEEE Comput. Soc., 1984, pp. 211–216.

[18] M. Brady, "Computational approaches to image understanding," *ACM Comput. Surveys*, vol. 14, no. 1, pp. 3–71, Mar. 1982.

[19] M. Brady, J. Ponce, A. Yuille, and H. Asada, "Describing surfaces," *Comput. Vision, Graphics, Image Processing*, vol. 32, pp. 1–28, 1985.

[20] C. Brice and C. Fennema, "Scene analysis using regions," *Artificial Intell.*, vol. 1, pp. 205–226, 1970.

[21] B. Carrihill and R. Hummel, "Experiments with the intensity ratio depth sensor," *Comput. Vision, Graphics, Image Processing*, vol. 32, pp. 337–358, 1985.

[22] D. Chen, "A regression updating approach for detecting multiple curves," in *Proc. 2nd World Conf. Robotics Research*, Scottsdale,

AZ, Aug. 18–21, 1986, Paper RI/SME, MS86-764; also *IEEE Trans. Pattern Anal. Machine Intell.*, to be published.
[23] S. S. Chern, "A proof of the uniqueness of Minkowski's problem for convex surfaces," *Amer. J. Math.*, vol. 79, pp. 949–950, 1957.
[24] F. S. Cohen and D. B. Cooper, "Simple parallel hierarchical and relaxation algorithms for segmenting noncausal markovian random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 2, pp. 195–219, Mar. 1987.
[25] E. N. Coleman and R. Jain, "Obtaining shape of textured and specular surfaces using four-source photometry," *Comput. Graphics Image Processing*, vol. 18, no. 4, pp. 309–328, Apr. 1982.
[26] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 25–39, 1983.
[27] C. Dane, "An object-centered three-dimensional model builder," Ph.D. dissertation, Dep. Comput. Inform. Sci., Moore School Elec. Eng., Univ. Pennsylvania, Philadelphia, 1982.
[28] W. W. Daniel, *Applied Nonparametric Statistics*. Boston, MA: Houghton-Mifflin, 1978.
[29] L. S. Davis, "A survey of edge detection techniques," *Comput. Graphics Image Processing*, vol. 4, pp. 248–270, 1975.
[30] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 1, pp. 39–55, Jan. 1987.
[31] S. Dizenzo, "Advances in image segmentation," *Image and Vision Comput.*, vol. 1, no. 4, pp. 196–210, Nov. 1983.
[32] T. G. Fan, G. Medioni, and R. Nevatia, "Description of surfaces from range data using curvature properties," in *Proc. Computer Vision and Pattern Recognition Conf.*, IEEE Comput. Soc., Miami, FL, June 22–26, 1986, pp. 86–91.
[33] O. D. Faugeras and M. Hebert, "The representation, recognition, and locating of 3-D objects," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 27–52, Fall 1986.
[34] O. D. Faugeras, M. Hebert, and E. Pauchon, "Segmentation of range data into planar and quadric patches," in *Proc. 3rd Computer Vision and Pattern Recognition Conf.*, Arlington, VA, 1983, pp. 8–13.
[35] I. D. Faux and M. J. Pratt, *Computational Geometry for Design and Manufacture*. UK: Ellis Horwood, Chichester, 1979.
[36] F. P. Ferrie and M. D. Levine, "Piecing together 3D shape of moving objects: An overview," in *Proc. Computer Vision and Pattern Recognition Conf.*, IEEE Comput. Soc., San Francisco, CA, June 9–13, 1985, pp. 574–584.
[37] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, pp. 3–16, 1981.
[38] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.
[39] B. Gil, A. Mitiche, and J. K. Aggarwal, "Experiments in combining intensity and range edge maps," *Comput. Vision, Graphics, Image Processing*, vol. 21, pp. 395–411, Mar. 1983.
[40] D. Gilbarg and N. Trudinger, *Elliptic Partial Differential Equations of Second Order*. Berlin: Springer-Verlag, 1983.
[41] C. Goad, "Special purpose automatic programming for 3D model-based vision," in *Proc. Image Understanding Workshop*, DARPA, Arlington, VA, June 23, 1983, pp. 94–104.
[42] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins University Press, 1983.
[43] W. E. L. Grimson, "A computer implementation of a theory of human stereo vision," M.I.T. Artificial Intelligence Lab., Cambridge, MA, Memo. 565, 1980.
[44] ——, *From Images to Surfaces*. Cambridge, MA: M.I.T. Press, 1981.
[45] W. E. L. Grimson and T. Pavlidis, "Discontinuity detection for visual surface reconstruction," *Comput. Vision, Graphics, Image Processing*, vol. 30, pp. 316–330, 1985.
[46] E. L. Hall, J. B. K. Tio, C. A. McPherson, and F. A. Sadjadi, "Measuring curved surfaces for robot vision," *Computer*, vol. 15, no. 12, pp. 42–54, Dec. 1982.
[47] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Comput. Vision, Graphics, Image Processing*, vol. 29, pp. 100–132, 1985.
[48] R. M. Haralick and L. Watson, "A facet model for image data," *Comput. Graphics Image Processing*, vol. 15, pp. 113–129, 1981.
[49] R. M. Haralick, L. T. Watson, and T. J. Laffey, "The topographic primal sketch," *Int. J. Robotics Res.*, vol. 2, no. 1, pp. 50–72, Spring 1983.
[50] M. Hebert and T. Kanade, "The 3-D profile method for object recognition," in *Proc. Computer Vision and Pattern Recognition Conf.*, IEEE Comput. Soc., San Francisco, CA, June 9–13, 1985, pp. 458–463.
[51] M. Hebert and J. Ponce, "A new method for segmenting 3-D scenes into primitives," in *Proc. 6th Int. Conf. Pattern Recognition*, Munich, West Germany, Oct. 19–22, 1982, pp. 836–838.
[52] T. C. Henderson, "Efficient 3-D object representations for industrial vision systems," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, no. 6, pp. 609–617, Nov. 1983.
[53] T. C. Henderson and B. Bhanu, "Three-port seed method for the extraction of planar faces from range data," in *Proc. Workshop Industrial Applications of Machine Vision*, Research Triangle Park, NC, May 1982, pp. 181–186.
[54] M. Herman, "Generating detailed scene descriptions from range images," in *Proc. Int. Conf. Robotics and Automation*, St. Louis, MO, Mar. 25–28, 1985, pp. 426–431.
[55] B. K. P. Horn, "Extended Gaussian images," *Proc. IEEE*, vol. 72, no. 12, pp. 1656–1678, Dec. 1984.
[56] K. Ikeuchi and B. K. P. Horn, "Numerical shape from shading and occluding boundaries," *Artificial Intell.*, vol. 17, pp. 141–184, Aug. 1981.
[57] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a directed split-and-merge procedure," in *Proc. 2nd Int. Joint Conf. Pattern Recognition*, 1974, pp. 424–433.
[58] K. Ikeuchi and B. K. P. Horn, "Numerical shape from shading and occluding boundaries," *Artificial Intell.*, vol. 17, pp. 141–184, Aug. 1981.
[59] S. Inokuchi and R. Nevatia, "Boundary detection in range pictures," in *Proc. 5th Int. Conf. Pattern Recognition*, Miami, FL, Dec. 1–4, 1980, pp. 1031–1035.
[60] S. Inokuchi, T. Nita, F. Matsuday, and Y. Sakurai, "A three-dimensional edge-region operator for range pictures," in *Proc. 6th Int. Conf. Pattern Recognition*, Munich, West Germany, Oct. 19–22, 1982, pp. 918–920.
[61] S. Inokuchi, K. Sato, and F. Matsuda, "Range imaging system for 3-D object recognition," in *Proc. 7th Int. Conf. Pattern Recognition*, Montreal, P.Q., Canada, July 30–Aug. 2, 1984, pp. 806–808.
[62] R. Hoffman and A. K. Jain, "Segmentation and classification of range images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 5, pp. 608–620, Sept. 1987.
[63] R. Jain, "Dynamic scene analysis," in *Progress in Pattern Recognition*, vol. 2, A. Rosenfeld and L. Kanal, Eds. Amsterdam, The Netherlands: North-Holland, 1983.
[64] R. A. Jarvis, "A perspective on range finding techniques for computer vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, no. 2, pp. 122–139, Mar. 1983.
[65] T. Kanade, "Survey: Region segmentation: Signal vs. semantics," *Comput. Graphics Image Processing*, vol. 13, pp. 279–297, 1980.
[66] ——, "Recovery of the three-dimensional shape of an object from a single view," *Artificial Intell.*, vol. 17, pp. 409–460, Aug. 1981.
[67] J. R. Kender and E. M. Smith, "Shape from darkness: Deriving surface information from dynamic shadows," in *Proc. 5th Nat. Conf. Artifical Intelligence*, AAAI, Philadelphia, PA, Aug. 11–15, 1986, pp. 664–669.
[68] G. Kinoshita, M. Idesawa, and S. Naomi, "Robotic range sensor with projection of bright ring pattern," *J. Robotic Syst.*, vol. 3, no. 3, pp. 249–257, 1986.
[69] D. T. Kuan and R. J. Drazovich, "Model-based interpretation of range imagery," in *Proc. Nat. Conf. Artificial Intelligence*, Austin, TX, Aug. 6–10, 1984, pp. 210–215.
[70] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Englewood Cliffs, NJ: Prentice-Hall, 1974.
[71] R. A. Lewis and A. R. Johnston, "A scanning laser rangefinder for a robotic vehicle," in *Proc. 5th Int. Joint Conf. Artificial Intelligence*, Cambridge, MA, Aug. 22–25, 1977, pp. 762–768.
[72] C. Lin and M. J. Perry, "Shape description using surface triangularization," in *Proc. Workshop Computer Vision: Representation and Control*, IEEE Comput. Soc., Rindge, NH, Aug. 23–25, 1982, pp. 38–43.
[73] D. Marr, *Vision*. New York: Freeman, 1982.
[74] G. Medioni and R. Nevatia, "Description of 3-D surfaces using curvature properties," in *Proc. Image Understanding Workshop*, DARPA, New Orleans, LA, Oct. 3–4, 1984, pp. 291–299.
[75] D. L. Milgrim and C. M. Bjorklund, "Range image processing: Planar surface extraction," in *Proc. 5th Int. Conf. Pattern Recognition*, Miami, FL, Dec. 1–4, 1980, pp. 912–919.

[76] B. Gil, A. Mitiche, and J. K. Aggarwal, "Experiments in combining intensity and range edge maps," *Comput. Vision, Graphics, Image Processing*, vol. 21, pp. 395–411, Mar. 1983.
[77] R. Nevatia and T. O. Binford, "Structured descriptions of complex objects," in *Proc. 3rd Int. Joint Conf. Artificial Intelligence*, Stanford, CA, Aug. 20–23, 1973, pp. 641–647.
[78] W. M. Newman and R. F. Sproull, *Principles of Interactive Computer Graphics*, 2nd ed. New York: McGraw-Hill, 1979.
[79] M. Oshima and Y. Shirai, "Object recognition using three-dimensional information," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, no. 4, pp. 353–361, July 1983.
[80] T. Pavlidis, "Segmentation of pictures and maps through functional approximation," *Comput. Graphics Image Processing*, vol. 1, pp. 360–372, 1972.
[81] F. G. Peet and T. S. Sahota, "Surface curvature as a measure of image texture," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 6, pp. 734–738, Nov. 1985.
[82] T. C. Pong, L. G. Shapiro, L. T. Watson, and R. M. Haralick, "Experiments in segmentation using a facet model region grower," *Comput. Vision, Graphics, Image Processing*, vol. 25, pp. 1–23,, 1984.
[83] R. J. Popplestone, C. M. Brown, A. P. Ambler, and G. F. Crawford, "Forming models of plane-and-cylinder faceted bodies from light stripes," in *Proc. 4th Int. Joint Conf. Artificial Intelligence*, Tbilisi, Georgia, USSR, Sept. 1975, pp. 664–668.
[84] M. Potmesil, "Generating models of solid objects by matching 3D surface segments," in *Proc. 8th Int. Joint Conf. Artificial Intelligence*, Karlsruhe, West Germany, Aug. 8–12, 1983, pp. 1089–1093.
[85] J. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictorics*, B. Lipkin and A. Rosenfeld, Eds. New York: Academic, 1979, pp. 75–149.
[86] G. T. Reid, "Automatic fringe pattern analysis: a review," *Opt. Lasers Eng.*, vol. 7, pp. 37–68, 1986.
[87] W. Richards and D. D. Hoffman, "Codon constraints on closed 2D shapes," *Comput. Vision, Graphics, Image Processing*, vol. 31, pp. 265–281, 1985.
[88] E. M. Riseman and M. A. Arbib, "Computational techniques in the visual segmentation of static scenes," *Comput. Graphics Image Processing*, vol. 6, pp. 221–276, 1977.
[89] I. Rock, *The Logic of Perception*. Cambridge, MA: M.I.T. Press, 1983.
[90] A. Rosenfeld and L. S. Davis, "Image segmentation and image models," *Proc. IEEE*, vol. 67, no. 5, pp. 764–772, May 1979.
[91] A. Rosenfeld and A. Kak, *Digital Picture Processing*, vols. 1 and 2. New York: Academic, 1982.
[92] I. K. Sethi and S. N. Jayaramamurthy, "Surface classification using characteristic contours," in *Proc. 7th Int. Conf. Pattern Recognition*, Montreal, P.Q., Canada, July 30–Aug. 2, 1984, pp. 438–440.
[93] Y. Shirai, "Recognition of polyhedrons with a range finder," *Pattern Recognition*, vol. 4, pp. 243–250, 1972.
[94] Y. Shirai and M. Suwa, "Recognition of polyhedra with a range finder," in *Proc. 2nd Int. Joint Conf. Artificial Intelligence*, London, UK, Aug. 1971, pp. 80–87.
[95] D. R. Smith and T. Kanade, "Autonomous scene description with range imagery," *Comput. Vision, Graphics, Image Processing*, vol. 31, pp. 322–334, 1985.
[96] W. Snyder and G. Bilbro, "Segmentation of three-dimensional images," in *Proc. Int. Conf. Robotics and Automation*, IEEE Comput. Soc., St. Louis, MO, Mar. 25–28, 1985, pp. 396–403.
[97] K. Sugihara, "Range-data analysis guided by junction dictionary," *Artificial Intell.*, vol. 12, pp. 41–69, 1979.
[98] D. Terzopoulos, "Computing visible surface representations," Artificial Intell. Lab., M.I.T., Cambridge, MA, AI Memo 800, Mar. 1985.
[99] W. Tiller, "Rational B-splines for curve and surface representation," *IEEE Comput. Graphics Applications*, vol. 3, no. 6, pp. 61–69, 1983.
[100] F. Tomita and T. Kanade, "A 3D vision system: Generating and matching shape descriptions in range images," in *Proc. Int. Conf. Robotics*, IEEE Comput. Soc., Atlanta, GA, Mar. 13–15, 1984, pp. 186–191.
[101] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, MA: M.I.T. Press, 1979.
[102] B. C. Vemuri, A. Mitiche, and J. K. Aggarwal, "Curvature-based representation of objects from range data," *Image and Vision Comput.*, vol. 4, no. 2, pp. 107–114, May 1986.
[103] A. P. Witkin, "Recovering surface shape and orientation from texture," *Artificial Intell.*, vol. 17, pp. 17–45, Aug. 1981.
[104] A. P. Witkin and J. Tenenbaum, "The role of structure in vision," in *Human and Machine Vision*, Beck et al., Eds. New York: Academic, 1983, pp. 481–543.
[105] R. J. Woodham, "Analysing images of curved surfaces," *Artificial Intell.*, vol. 17, pp. 117–140, Aug. 1981.
[106] J. W. Woods, "Two-dimensional discrete Markov fields," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 232–240, 1972.
[107] S. W. Zucker, "Region growing: Childhood and adolescence," *Comput. Graphics Image Processing*, vol. 5, pp. 382–399, 1976.
[108] D. M. Zuk and M. L. Delleva, "Three-dimensional vision system for the adaptive suspension vehicle," Defense Supply Service, Washington, Final Rep. 170400-3-F, ERIM, DARPA 4468, 1983.

Paul J. Besl (M'81–S'84–M'87) graduated *summa cum laude* in physics from Princeton University, Princeton, NJ, in 1978 and received the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of Michigan, Ann Arbor, in 1981 and 1986, respectively. In 1987, he received a Rackham Distinguished Dissertation Award for his thesis on range image understanding from the University of Michigan.

From 1979 to 1981, he did computer simulations for Bendix Aerospace Systems in Ann Arbor, MI, and from 1981 to 1983, worked on the GEOMOD solid modeling system at Structural Dynamics Research Corp. in Cincinnati, OH. Currently, he is a Research Scientist at General Motors Research Laboratories in Warren, MI, where his primary research interest is computer vision, especially range image analysis and geometric modeling for image understanding.

Dr. Besl is a member of the Association for Computing Machinery, the American Association for Artificial Intelligence, and the Machine Vision Association of the Society of Manufacturing Engineers.

Ramesh C. Jain (M'79–SM'83) received the B.E. degree from Nagpur University in 1969 and the Ph.D. degree from the Indian Institute of Technology, Kharagpur, India, in 1975.

He is a Professor of Electrical Engineering and Computer Science, and Director of the Computer Vision Research Laboratory at the University of Michigan, Ann Arbor. Formerly he worked at General Motors Research Labs, Wayne State University, University of Texas at Austin, University of Hamburg, West Germany, and Indian Institute of Technology, Kharagpur, India. His current research interests are in computer vision, and artificial intelligence. He has been active in dynamic scene analysis, range image understanding, industrial inspection, object recognition, knowledge-based systems, and related areas. He has published research papers addressing several aspects of the above areas. He is a consultant to many companies in the areas of computer vision, artificial intelligence, and computer graphics.

Dr. Jain is a member of the Association for Computing Machinery, the American Association for Artificial Intelligence, the Pattern Recognition Society, the Cognitive Science Society, the Optical Society of America, the Society of Photo-Optical Instrumentation Engineers, and Society of Manufacturing Engineers. He has been involved in organization of several professional conferences and workshops. Currently, he is on the Editorial Boards of *IEEE Expert*, *Machine Vision and Applications*, *Computer Vision Graphics and Image Processing*, the *Bulletin of Approximate Reasoning*, and *Image and Vision Computing*.

Chapter 3: Feature Extraction and Matching

Segmented images derived using methods described in Chapter 2 are represented in a compact form to facilitate further abstraction. Often, shape and region features such as curvature and topology are extracted from the segmented images. Representation schemes are chosen to match the methods used for object recognition and description. In the following sections, schemes for representation and description that are popular in computer vision are first discussed. Then, techniques for feature extraction are described. The third section describes various matching techniques. Object recognition requires matching an object description in an image to models of known objects. The models, in turn, use certain descriptive features and their relations. Matching also plays an important role in other aspects of information recovery from images. For example, stereo and structure from motion depend upon matching selected points or features in two or more images. This problem of matching selected points is known as the "correspondence problem."

Representation and description

Symbolic representation of iconic information in a segmented image is the goal of representation schemes. Iconic information is usually very rich and redundant. Symbolic information, which is an abstraction of iconic information, contains the main features of the iconic information at the cost of minute details. While this loss may be acceptable or even desirable in some applications, it should — in general — be minimized.

Representation and description of symbolic information can be approached in many ways. One approach is to represent the object in terms of its bounding curve. Popular among several methods developed for boundary representation are chain codes, polygonalization, one-dimensional signatures, and representation using dominant points. If recognition is the only aim, then one may want to use some features, such as corners and inflection points, for representing an object. Another approach is to obtain region-based shape descriptors, such as topological or texture descriptors. Representation and shape description schemes are usually chosen such that the descriptors are invariant to rotation, translation, and scale change. In this section, we will discuss some common methods for representation of boundaries and regions, as well as characterization of two-dimensional shape.

Chain codes. One of the earliest methods for representing a boundary uses directional codes called "chain codes." The object boundary is resampled at appropriate scale, and an ordered list of points on the boundary is represented by a string of directional codes. Typical directional codes and their applications to an object boundary are illustrated in Figure 3.1.[1] Often, to retain all the information in the boundary, the resampling step is bypassed; however, resampling eliminates minor fluctuations that typically are due to noise. The use of chain codes has some attractive features. For example,

- Rotation of an object by 45° can be easily implemented.
- The derivative of the chain code, obtained by using first difference, is rotation invariant.
- Other characteristics of a region, such as area and corners, can be computed directly using the chain code.

The limitation of this representation method is attributable to the limited directions used to represent the tangent at a point. Although codes with larger number of directions[67] are occasionally used, eight-directional chain code is the most commonly used code.

Polygonalization. Polygonal approximation of boundaries of objects has been studied extensively and numerous methods have been developed. The fit is made to reduce a chosen error criterion between the approximation and the original curve. In the iterative endpoint-fit algorithm,[68] the first step is to connect a straight-line segment between the two farthest points on the boundary. The perpendicular distances from the segment to each point on the curve are measured. If any distance is greater than a chosen threshold, the segment is replaced by two segments; one each from a segment endpoint to the curve point where the distance to the segment is greatest. This process is iterated until all segments are within the threshold. In papers by Tomek,[69] Williams,[70,71] Sklansky and Gonzalez,[72] and Pavlidis,[73] a straight-line fit is constrained to pass within a radius around each data point. The line segment is grown from the first point, and when further extension of the line segment causes it to fall outside the radius of a point, a new line is started. Kurozumi and Davis[74] employed a minimax approach, in which the line segment approximations are chosen to minimize the maximum distance between the data points and the approximating line segment.

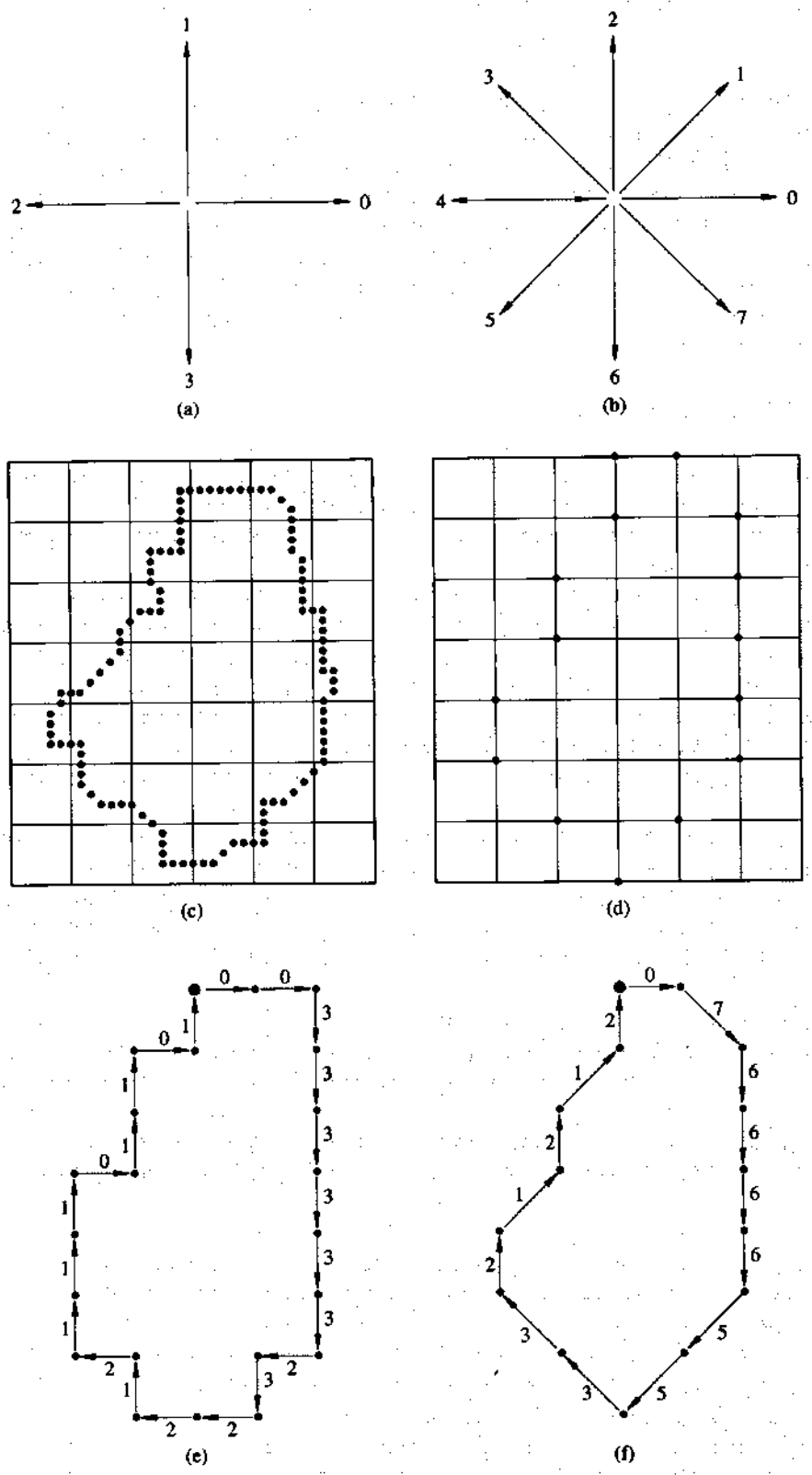


Figure 3.1: (a) Four-directional chain code; (b) eight-directional chain code; (c) digital boundary with resampling grid superimposed; (d) result of resampling; (e)-(f) corresponding four and eight directional codes. (from R.C. Gonzalez and P. Wintz, *Digital Image Processing*, © 1977 by Addison-Wesley Publishing Company. Reprinted with permission of the publisher.)[1]

In another class of techniques, area versus distance is used as a measure of "goodness" of fit. Wall and Danielsson[75] used a scan-along technique in which a new line segment is generated if the area deviation for each line segment exceeds a preset value. Wall[76] used this polygonal approximation for generating a smooth cubic curve as an approximation to the original data. Images of the coastline of Great Britain and its approximations are shown in Figure 3.2.[76]

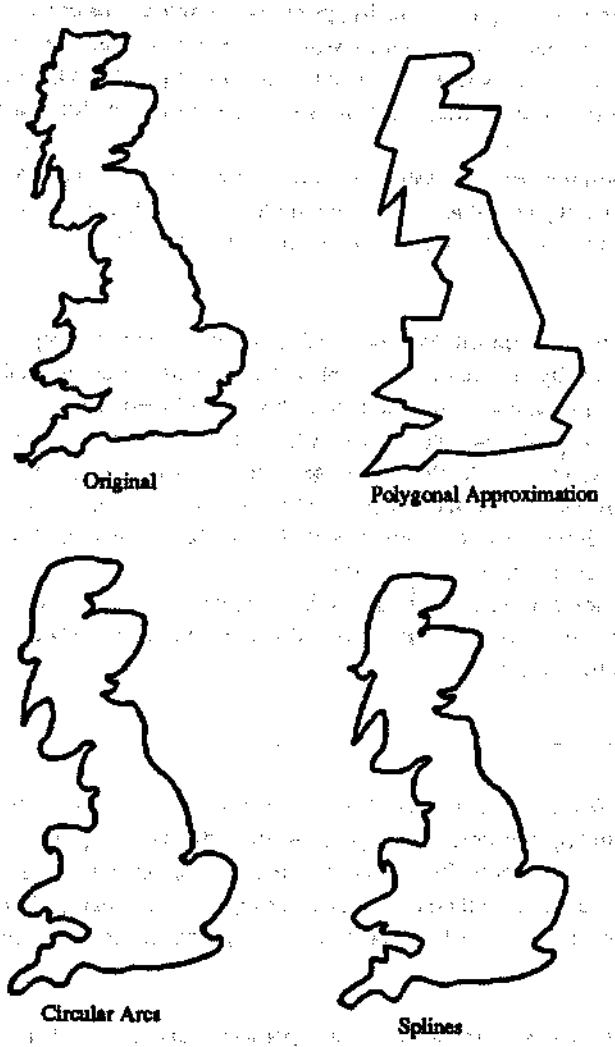


Figure 3.2: Illustration of polygonal, circular arc, and spline approximations (from Wall)[76]

Leu and Chen[77] focus on uniqueness and accuracy of representation, two important issues of polygonal approximation. Uniqueness is achieved by starting the approximation simultaneously at places along the shape boundary where the arcs are closer to straight lines than their neighboring arcs. Polygonal approximation is performed by fitting lines to those selected arcs whose maximum arc-to-chord distance is less than a given tolerance.

Besides polygonal approximation methods, higher order curve- and spline-fitting methods are used where more precise approximations are required. These are more computationally expensive than most polygonalization methods, and they can be more difficult to apply. Some of these methods are described in Pavlidis[73,78,79] and Davis.[80]

One of the drawbacks of most of these polygonal-fit techniques is that the operations are not performed symmetrically with respect to curve features such as corners and the centers of curves. The result is that the computed breakpoints between segments may be at different locations depending on the starting and ending locations or on the direction of operation of the line fit. Extensions can be made to some of the methods to produce fewer and more consistent breakpoint locations, but these procedures are usually iterative and can be computationally expensive.

One-dimensional signatures. The slope of the tangent to the curve denoted by the angle θ as a function of position s from an arbitrary starting point is used to represent curves in many applications. Horizontal lines in s-θ curves represent straight lines, and straight lines represent circular arcs whose radii are proportional to the slope of the straight line. An s-θ curve can be treated

as a periodic function with a period given by the perimeter. Other functions — such as distance to the curve from an arbitrary point inside the curve plotted as a function of angle with reference to the horizontal — are also used as shape signatures.

Boundary description. Descriptors for objects represented by their boundaries may be generated using the representations described in the previous sections. Simple descriptors — such as perimeter, length and orientation of the major axis, shape number, and eccentricity — may be readily computed from boundary data. The ordered set of points on the boundary having two-dimensional coordinates (x_k,y_k), where $k = 1,...N$ and N is the total number of points on the boundary, can be treated as a one-dimensional complex function: $(x_k + iy_k)$. The coefficients of the discrete Fourier transform applied to this function can be used as a shape descriptor.

Other descriptors that use the one-dimensional signature functions may also be obtained. A major drawback of many of these descriptors is that complete data for the object boundaries are required. However, because of either problems in segmentation or occlusions in the image, complete data for object boundaries are often not available. In such instances, recognition strategies based on partial information are needed.

Region-based representation and description. Methods analogous to those used to represent object boundaries have been developed to represent regions enclosed by boundaries. Examples of such methods are medial-axis transformations, skeletons, and convex hulls and deficiencies.[1,4] Morphological operators have been developed to extract shape features and generate useful descriptions of object shape.[81-84] Topological descriptors, such as Euler number, are also useful to characterize shape. Haralick et al.[85] used a local facet model — described in Chapter 2 — to generate a topographic primal sketch of images. In their model, each pixel is uniquely described using a set of seven descriptive labels, including peak, pit, and ridge. Such a description is very useful for object matching. Shapiro[86] describes a structural model of shape based on a set of primitives and their properties; she also describes a shape-matching procedure that uses this model.

Region-based representation and description are particularly useful to describe objects for which properties within the region — for example, texture — are significant for object recognition. Many techniques have been developed to model texture using statistical,[58,59] structural,[60,61] and spectral[62] methods.

Feature extraction

Shape descriptors — such as topological descriptors or Fourier descriptors — are useful for object recognition. If the object to be recognized has unique discriminating features, special-purpose algorithms are employed to extract such features. Important for object matching and recognition are corners, high-curvature regions, or other regions along curves at which curvature discontinuities exist. In region-based matching methods, identifying groups of pixels that can be easily distinguished and identified is important. Many methods have been proposed to detect both dominant points in curves and interesting points in regions. These methods are discussed subsequently.

Detection of dominant points in curves. Detection of critical points in curves — such as corners and inflection points — is important for subsequent object matching and recognition. Most algorithms for detecting critical points follow the idea of Attneave,[87] marking the local curvature maxima points as dominant. Several feature detection algorithms using this approach were compared and evaluated by Teh and Chin.[88] Fischler and Bolles[89] analyzed the deviations of a curve from a chord to detect dominant points along a curve. In their approach, points are marked as being critical or as belonging to a smooth or a noisy interval; these markings depend on whether the curve makes a single excursion away from the chord, stays close to the chord, or makes two or more excursions away from the chord, respectively. A drawback of many such methods based on critical points of high curvature is that inflection points due to smooth changes between segments — such as transitions from a circular arc to a tangential line — are not detected.

Most of the algorithms employing critical-point detection require parameters related to the separation of minimum resolvable features. However, features of varying size and separation are usually present in a given image. Parameter values determined by the minimum-size feature may not be adequate to smooth large features; as a result, too many points may be detected as dominant points. One approach to solving this problem is to adaptively determine parameters using local feature data — with no required user parameters. Teh and Chin[88] describe an algorithm that uses such an approach. First, the region of support is adaptively determined using local properties; and local curvature is measured within this region. Then, dominant points are detected by nonmaxima suppression of local curvature. Phillips and Rosenfeld[90] discuss determination of region of support using an arc-chord distance property.

The problem of feature point detection in digital curves may also be approached as a scale-space problem.[91,92] By defining a set of primitive parameterized curvature discontinuities, Asada and Brady[91] introduced curvature primal sketch. In this approach,

curvature as a function of position is computed at multiple scales; these are convolved with a Gaussian function, and the second derivative of the result is computed. Curvature primal sketch is obtained by analyzing this output.

Saint-Marc et al.[93] describe an adaptive-filtering algorithm for smoothing noisy curves. The closed curve shown in Figure 3.3(a) illustrates the performance of this algorithm. The objective is to locate all the vertices and other critical points, such as the inflection point along the curved segment, as well as points of transition from curve to straight-line segments. To obtain the curve shown in Figure 3.3(b), the curvature at each point along the line is determined using a small region of support. Because of artifacts introduced during digitization and thinning, this curve is not smooth. Smoothing is necessary to identify features such as vertices (peaks and valleys in the curvature plot), inflection points (zero-crossings in curvature), and smooth joins (points of transition from zero curvature to a significant value). A common filter used to smooth noisy signals is the Gaussian filter. However, Gaussian filtering smooths both noise and data points. Alternatively, an adaptive filter that emphasizes intraregion smoothing over interregion smoothing is useful in this situation. This filtering approach is now described.

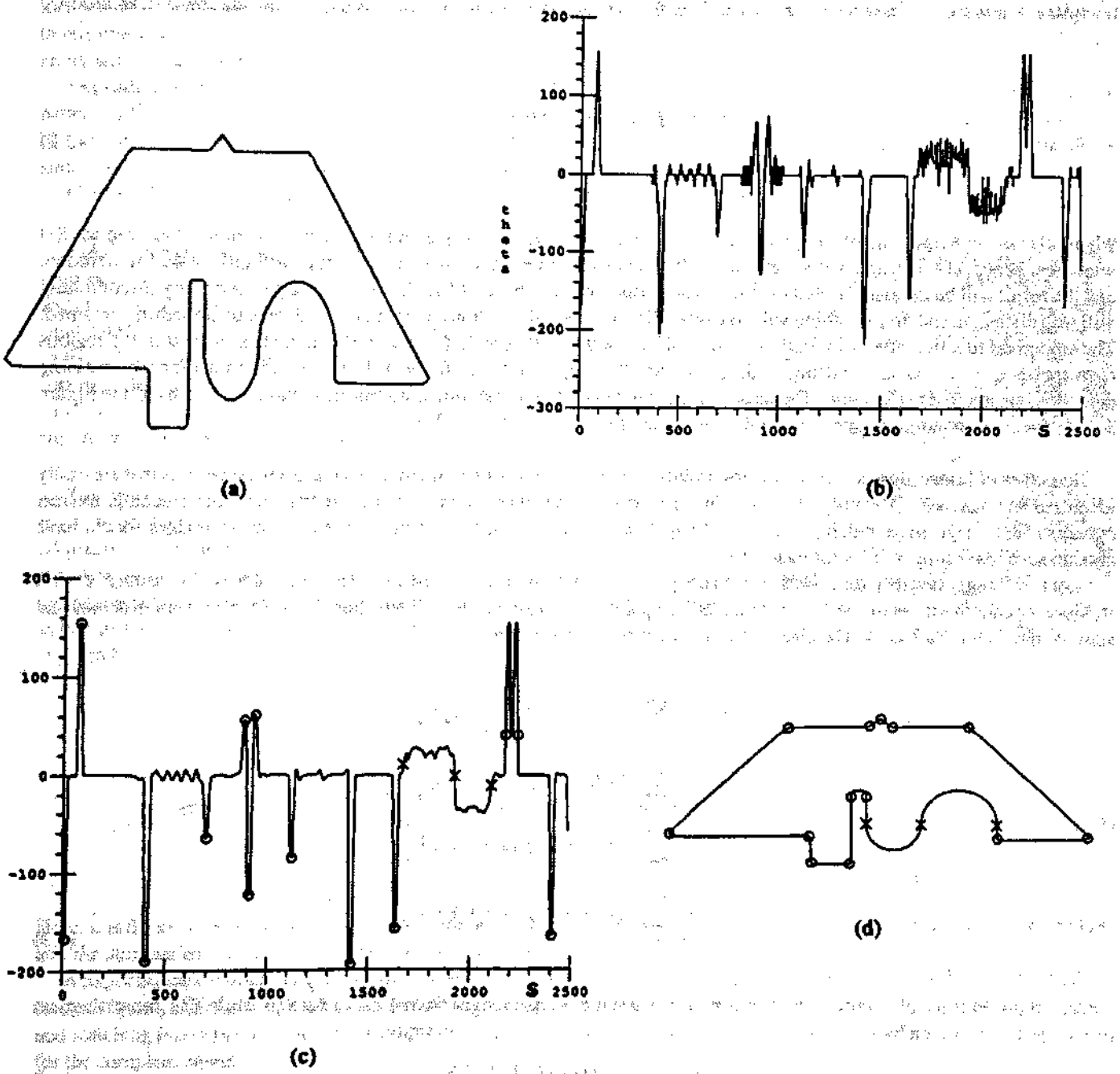


Figure 3.3: Adaptive smoothing for detection of critical points; (a) a closed curve; (b) plot of estimated curvature as a function of position; (c) curvature after adaptive smoothing; (d) critical points.

Let $I_0(s)$ be the signal before smoothing and $I_t(s)$ the signal at the t^{th} iteration. The smoothed version of $I_t(s)$ is then defined at each point by

$$I_{t+1}(s) = \left(\frac{1}{\sum_{i=-1}^{+1} c_t(s+i)} \right) \sum_{i=-1}^{+1} I_t(s+i)\, c_t(s+i) \tag{3.1}$$

where $c(s)$ is a coefficient array of the same size as $I(s)$. Values of $c(s)$ are close to zero at region boundary points and one at region interior points. Thus, two points belonging to different regions are not averaged. Because locations of the region boundary points are not known a priori, an estimate thereof, based on local curvature, is used in calculating $c(s)$, as follows:

$$c(s) = f(d(s)) = \exp\left(\frac{-d^2(s)}{2k^2} \right) \tag{3.2}$$

where $d(s)$ is the magnitude of the gradient at s. The value of $f(0)$ equals one and that of $f(d(s))$ approaches zero as $d(s)$ increases; hence, $c(s)$ is small at discontinuities. If k is chosen to be small, then every feature will diffuse during iteration, and the result will be the same as that obtained with Gaussian smoothing. If k is chosen to be large, then every discontinuity will stop diffusion, and no smoothing will take place. The number of iterations determines the degree of smoothing obtained. The smoothed function after 75 iterations with $k=40$ is shown in Figure 3.3(c). Some fluctuations still exist in the regions corresponding to circular arcs and the two diagonal lines. Two thresholds, one for peak detection and another for zero-crossing detection, are applied to this curve. Detected peaks are noted by $O's$ and significant zero-crossings are noted by $X's$ in Figure 3.3(c); the corresponding points in the object are noted likewise in Figure 3.3(d).

Detection of interesting points in regions. Points used in matching of points from two images must be ones that are easily identified and matched. Obviously, the points in a uniform region and on edges are not good candidates for matching. Interest operators find image areas with high variance. In applications such as stereo and structure from motion, images should have enough such interesting regions for matching.

Moravec[94] suggested that, for a window in an image, directional variances may be a good measure of how "interesting" a point is; Moravec calls this measure the "interestingness" of a point. A point is considered interesting if it has local maximum of minimal sums of directional variances. The directional variances for a window are

$$\begin{aligned}
I_1 &= \sum_{(x,y)\in s} \left(f(x,y) - f(x,y+1) \right)^2 \\
I_2 &= \sum_{(x,y)\in s} \left(f(x,y) - f(x+1,y) \right)^2 \\
I_3 &= \sum_{(x,y)\in s} \left(f(x,y) - f(x+1,y+1) \right)^2 \\
I_4 &= \sum_{(x,y)\in s} \left(f(x,y) - f(x+1,y-1) \right)^2
\end{aligned} \tag{3.3}$$

where s represents the elements in the window. Typical window size ranges from 4 x 4 to 8 x 8 pixels.[94] The interestingness of a point is then given by

$$I(x,y) = Min\,(I_1, I_2, I_3, I_4) \tag{3.4}$$

Choosing points as just described eliminates simple edge points since they have no variance in the direction of the edge.

Feature points are chosen where the interestingness has local maximum. A point is considered "good" as an interesting point if, in addition, local maximum is more than a preset threshold. The Moravec interest operator has found extensive use in stereo-matching applications.

Matching

Matching plays a very important role in many phases of computer vision systems. Object recognition requires matching a description of an object in an image with models of known objects. The goal of matching is to either (1) detect the presence of a known entity, object, or feature or (2) find what an unknown image component is. The difficulty in achieving these matching goals is first encountered with goal-directed matching, wherein the goal is to find a very specific entity in an image. Usually, the location of all instances of such an entity must be found. In stereo and structure-from-motion applications, entities are obtained in one image, and their locations are then determined in the second image. The second problem requires matching an unknown entity with several models to determine which model matches the best.

Depending on the application, matching may be required at different levels. Common entities involved in matching are (1) point patterns, (2) features such as corners and line segments, (3) regions, and (4) objects. Ideally, matched entities should be identical. In computer vision applications, exactly matched entities are rare; rather, matching is usually a maximization of a measure of similarity.

Commonly used matching techniques are discussed in this section.

Point pattern matching. In matching points in two slightly different images of the same scene (e.g., in a stereo pair or a motion sequence), interesting points are detected by applying an operator such as the Moravec interest operator discussed in the previous section. The correspondence process considers local structure of a selected point in one image in assigning initial matchable candidate points from the second image. For example, in stereo-matching applications, the displacement of a point from one image to the other is usually small; thus, only points within a local neighborhood are considered for matching. To obtain final correspondence, the set of initial matches is refined by computing the measure of similarity in global structure around each candidate point. For example, in dynamic-scene analysis, one may assume that motions of neighboring points do not differ significantly. To obtain final matching of points in the two images, relaxation techniques are often employed .

Template matching. In some applications, a particular pictorial or iconic structure, called a "template," should be detected in an image. Templates are usually represented by two-dimensional-intensity functions of small extent (typically, less than 64 x 64 pixels). Template matching is the process of moving the template over the entire image and detecting locations at which the template best fits the image. The commonly used measure of similarity to determine match is the normalized correlation. The arrangement for finding the correlation between the image $f(x,y)$ and the template $w(x,y)$ at a point (m,n) in the image is shown in Figure 3.4.[1] The correlation coefficient $r(m,n)$, which is independent of multiplicative changes in the image intensity function, is given by

$$r(m,n) = \frac{\sum_x \sum_y \left[f(x,y) - \bar{f}(x,y)\right]\left[w(x-m,y-n) - \bar{w}\right]}{\left\{\sum_x \sum_y \left[f(x,y) - \bar{f}(x,y)\right]^2 \sum_x \sum_y \left[w(x-m,y-n) - \bar{w}\right]^2\right\}^{0.5}} \tag{3.5}$$

Here, $\bar{w}$ is the average intensity of the pixels in the template, $\bar{f}(x,y)$ is the average intensity of image pixels within the window, and the summations are carried out over all pixels within the window.

A major limitation of the template-matching technique is its sensitivity to scaling and rotation of objects. To match scaled and rotated objects, separate templates should be constructed. In some approaches, a template is partitioned into several subtemplates, and matching is computed for these subtemplates. The relationships among subtemplates are verified in the final matching step for the complete object.

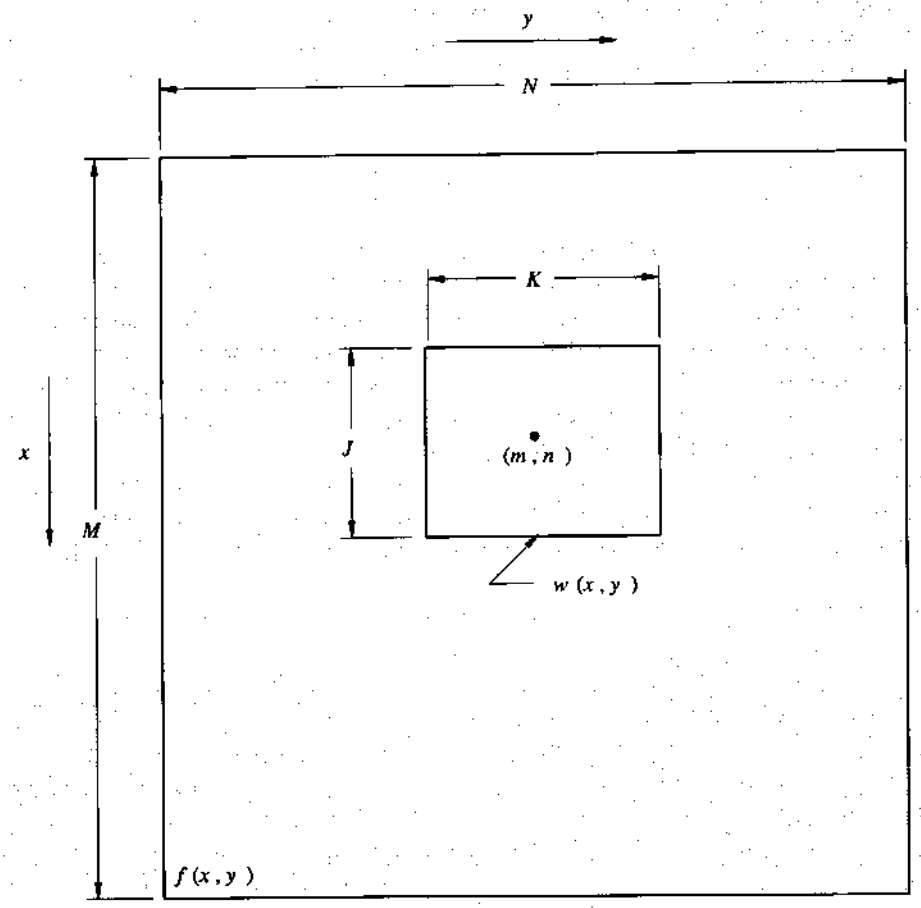


Figure 3.4: Arrangement for calculating the correlation of f(x,y) with w (x,y) at image point (m,n) (from R.C. Gonzalez and P. Wintz, *Digital Image Processing*, © 1977 by Addison-Wesley Publishing Company. Reprinted with permission of the publisher.)[1]

Hough transform. Hough transform is a useful method for recognition of straight lines and curves. For a straight line, the transformation is defined by the parametric representation of a straight line given by

$$\rho = x\cos\theta + y\sin\theta \tag{3.6}$$

where (ρ,θ) are the variables in the parameter space, represent the length and orientation, respectively, of the normal to the line from the origin. Each point (x,y) in the image plane transforms into a sinusoid in the (ρ,θ) domain, as given by the preceding equation. However, all the sinusoids corresponding to points that are collinear in the image plane intersect at a single point in the Hough domain. Thus, pixels belonging to a straight line can be easily identified in the Hough domain. Hough transform is typically implemented as a voting scheme. Each pixel in the image "votes" for several cells in the parameter space. The cell in the parameter space with the most votes characterizes the corresponding line in the spatial domain.

Hough transform can be defined to recognize other types of curves. For example, points on a circle can be detected by searching through a three-dimensional parameter space of (X_c,Y_c,R), where the first two parameters, X_c and Y_c, define the location of the center and R represents radius. The Hough transform technique has also been generalized to detect arbitrary shapes.[95] A problem with the Hough transform is the large parameter search space required to detect even simple curves. This problem can be alleviated somewhat by the use of additional information that may be available in the spatial domain. For example, in detecting circles by the brute-force method, searching must be done in three-dimensional-parameter space; however, when the direction of the curve gradient is known, the search is restricted to just one dimension.

Feature matching. An image of an object has as its components many regions. In addition, an object may appear in various orientations, locations, and scales. Identifying an object in an image requires that all of its components first be identified and then

that their properties or feature values and relationships be computed. Subsequently, the objective is to match these components into the corresponding components in the object model. This matching may be treated as a graph-matching problem in which nodes represent regions and arcs represent relations between regions. Nodes in such a graph are matched by considering region-based properties and the entire graph is matched by considering the structural or spatial relationships among various regions. Object recognition using this method is equivalent to the graph isomorphism problem. When the object is only partially visible in the image, subgraph isomorphism can be used.

Pattern classification. Features extracted from images may be represented in a feature space. A classifier — such as the minimum-distance classifier, which is used extensively in statistical pattern recognition — may be used to identify the object. This pattern classification method is especially useful in applications wherein the objective is to assign a label, from among many possible labels, to a region in the image. The classifier may be taught in a supervised-learning mode, by using prototypes of known object classes, or in an unsupervised mode, in which the learning is automatic. Proper selection of features is critical for the success of this approach. Several methods of pattern classification using structural relationships have also been developed.

Hierarchical decomposition and matching. Techniques such as polygonal approximation and signature analysis are useful for shape recognition only when the complete outline of the object is available. Because of occlusions or other image segmentation problems, recognizing shapes when only parts of the object are visible becomes necessary in computer vision. In some applications, the number of possible different shapes to be recognized may be large. Then, complex parts may be represented as a combination of already known simple shapes and their spatial relationships, and similar decomposition steps may be followed during object recognition.

An object recognition system that creates a library of parts by hierarchical decomposition is described by Ettinger.[96] The library organization and indexing are designed to avoid linear search of all model objects. The system has hierarchical organization for both structure (whole object-to-component subparts) and scale (gross-to-fine features). Object representation is based on the curvature primal sketch of Asada and Brady.[91] Features used are corner, end, crank, smooth-join, inflection, and bump, which are derived from discontinuities in contour orientation and curvature. Subparts consist of subsets of these features; these subsets partition the object into components. The model libraries are automatically built using the hierarchical nature of the model representations. The recognition engine is structured as an interpretation tree.[97] A constrained search scheme is used for matching scene features to model features. Early in the search process, many configurations in the search space are pruned using simple geometric constraints, such as — for pairs of features — orientation difference, direction, and distance.

Research trends

Discontinuities in detected edges and object boundaries occur frequently during low-level image analysis. Hough transform has been used extensively as an efficient method for detecting such broken edges and boundaries. Nine papers were selected for inclusion in this chapter; five are in this book and the remaining four in its companion book, *Computer Vision: Advances and Applications*. We begin this section of *Principles* with the paper entitled "Generalizing the Hough Transform to Detect Arbitrary Shapes," by Ballard. In *Advances and Applications*, "Object Recognition and Localization via Pose Clustering," by Stockman applies the Hough transform technique to cluster poses of objects in two-dimensional and three-dimensional problems. This paper demonstrates the elegance and efficiency of these clustering approaches for solving object recognition problems. Also in *Advances and Applications*, Asada and Brady in "The Curvature Primal Sketch," describe a representation scheme based on significant changes in curvature along the boundary of a planar shape. The methods introduced in this paper have been used in other applications, including the object recognition system described by Ettinger in his paper in *Advances and Applications* entitled "Large Hierarchical Object Recognition Using Libraries of Parameterized Model Sub-parts." In the final paper in *Advances and Applications*, "Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes," Mokhtarian and Mackworth describe a technique for finding a description of planar curves at varying levels of detail. The generalized scale-space image of a planar curve that they obtain is invariant under rotation, uniform scaling, and translation of the curve; this invariance makes matching two curves convenient. Analogous to the concept of boundary representation by curvature primal sketch is the concept for surface description that is presented in "The Topographic Primal Sketch," by Haralick et al. in *Principles*. The topographic primal sketch is derived by the classifying and grouping of underlying image intensity surface patches. The paper found in *Principles*, entitled "Image Analysis Using Mathematical Morphology," by Haralick et al., is a tutorial on morphological techniques. Also in *Principles*, "A Structural Model of Shape," by Shapiro describes a shape as consisting of a set of primitives, their properties, and their interrelationships. Her model is used in a shape-matching procedure to find mappings from a prototype shape to a candidate shape. The final paper in *Principles* by Fischler and Bolles entitled "Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography" describes a technique for solving the

problem of location determination. Their technique makes possible interpreting/smoothing data that contain a significant percentage of gross errors.

Clearly, features play the most important role in recognition. In the early stages of computer vision, global features — such as moments and Fourier descriptors — were used. Since these features were inadequate for occluded objects, local features — such as corners and interest points — started attracting attention. Recently, more emphasis has been placed on domain-dependent features. Now, features are being designed considering the models of objects being recognized.[98-100] Features designed using models are not dictated by their generality and mathematical representability, but by their utility in recognizing objects. The idea of object-dependent features is becoming very popular.[101] An interesting direction in feature detection is the use of robust statistics. In most applications, features are detected using a form of least squares fitting. Least squares fitting presents problems in the presence of outliers. Hough transform techniques use voting to eliminate the undesirable influence of outliers. Recently, robust statistics have been increasingly used to detect features or to estimate parameters in the presence of outliers.[102-104] Current robust techniques are computationally very demanding. The Hough transform is a fast implementation of a simple robust technique. Fast robust techniques will likely play a very important role in feature detection. Neural nets appear to offer parallel methods for fast robust detection of features,[105,106] but to date their success has been limited to very simple images. Another interesting development in early vision, especially in relation to feature detection, is the increasing attention being given to qualitative vision.[107-109] In the detection of certain features, only signs of certain variables can be used to give a gross, but robust, classification that yields qualitative feature-related information. Such detection approaches are likely to be more successful in complex vision systems than very precise, but noise-sensitive, approaches.

References Cited
Chapter 3

67. H. Freeman, "Applications of Generalized Chain Coding Scheme to Map Data Processing," *Proc. IEEE Conf. Pattern Recognition and Image Processing*, IEEE CS Press, Los Alamitos, Calif., 1978, pp. 220-226.
68. U.E. Ramer, "An Iterative Procedure for the Polygonal Approximation of Plane Curves," *Computer Graphics and Image Processing*, Vol. 1, 1972, pp. 244-256.
69. I. Tomek, "Two Algorithms for Piecewise-Linear Continuous Fit of Functions of One Variable," *IEEE Trans. on Computers*, Vol. 23, No. 4, 1974, pp. 445-448.
70. C.M. Williams, "An Efficient Algorithm for the Piecewise Linear Approximation of Planar Curves," *Computer Graphics and Image Processing*, Vol. 8, 1978, pp. 286-293.
71. C.M. Williams, "Bounded Straight-Line Approximation of Digitized Planar Curves and Lines," *Computer Graphics and Image Processing*, Vol. 16, 1981, pp. 370-381.
72. J. Sklansky and V. Gonzalez, "Fast Polygonal Approximation of Digitized Curves," *Pattern Recognition*, Vol. 12, 1980, pp. 327-331.
73. T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, Maryland, 1982.
74. Y. Kurozumi and W.A. Davis, "Polygonal Approximation by Minimax Method," *Computer Graphics and Image Processing*, Vol. 19, 1982, pp. 248-264.
75. K. Wall and P.E. Danielsson, "A Fast Sequential Method for Polygonal Approximation of Digitized Curves," *Computer Graphics and Image Processing*, Vol. 28, 1984, pp. 220-227.
76. K. Wall, "Curve Fitting Based on Polygonal Approximation," *Proc. Eighth Int'l Conf. Pattern Recognition*, IEEE CS Press, Los Alamitos, Calif., 1986, pp. 1273-1275.
77. J.G. Leu and L. Chen, "Polygonal Approximation of 2-D Shapes through Boundary Merging," *Pattern Recognition Letters*, Vol. 8, 1988, pp. 231-238.
78. T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, New York, N.Y., 1977.
79. T. Pavlidis, "Survey: A Review of Algorithms for Shape Analysis," *Computer Graphics and Image Processing*, Vol. 7, 1978, pp. 243-258.
80. L.S. Davis, "Two Dimensional Shape Representation," *Handbook of Pattern Recognition and Image Processing*, eds. T.Y. Young and K.S. Fu, Academic Press, Orlando, Fla., 1986, pp. 233-245.
81. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, New York, N.Y., 1982.
82. J. Serra, "Introduction to Mathematical Morphology," *Computer Vision, Graphics, and Image Processing*, Vol. 35, No. 3, 1986, pp. 283-325.
83. R.M. Haralick, S.R. Sternberg, and X. Zhuang, "Image Analysis Using Mathematical Morphology," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 9, No. 5, 1987, pp. 532-550.
84. E.R. Dougherty and C.R. Giardina, *Morphological Methods in Image and Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J., 1987.
85. R.M. Haralick, L.T. Watson, and T.J. Laffey, "The Topographic Primal Sketch," *Int'l J. Robotics Research*, Vol. 2, No. 1, 1983, pp. 50-72.
86. L.G. Shapiro, "A Structural Model of Shape," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 2, No. 1, 1980, pp. 111-126.
87. F. Attneave, "Some Informational Aspects of Visual Perception," *Psycho. Review*, Vol. 61, 1954, pp. 183-193.
88. C.H. Teh and R.T. Chin, "On the Detection of Dominant Points on Digital Curves," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 8, 1989, pp. 859-872.
89. M.A. Fischler and R.C. Bolles, "Perceptual Organization and Curve Partitioning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 8, No. 1, 1986, pp. 100-105.
90. T.Y. Phillips and A. Rosenfeld, "A Method of Curve Partitioning Using Arc-Chord Distance," *Pattern Recognition Letters*, Vol. 5, 1987, pp. 245-249.
91. H. Asada and M. Brady, "The Curvature Primal Sketch," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 8, No. 1, 1986, pp. 26-33.
92. K. Deguchi, "Multi-Scale Curvatures for Contour Feature Extraction," *Proc. Ninth Int'l Conf. Pattern Recognition*, IEEE CS Press, Los Alamitos, Calif., 1988, pp. 1113-1115.
93. P. Saint-Marc, J.S. Chen, and G. Medioni, "Adaptive Smoothing: A General Tool for Early Vision," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, IEEE CS Press, Los Alamitos, Calif., 1989, pp. 618-624.
94. H.P. Moravec, "Toward Automatic Visual Obstacle Avoidance," *Proc. Int'l Joint Conf. Artificial Intelligence*, Morgan Kaufmann Publishers, Inc., San Mateo, Calif., 1977, p. 584.
95. D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, Vol. 13, No. 2, 1981, pp. 111-122.
96. G.J. Ettinger, "Large Hierarchical Object Recognition in Robot Vision," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, IEEE CS Press, Los Alamitos, Calif., 1988, pp. 32-41.
97. W.E.L. Grimson and T. Lozano-Perez, "Model-Based Recognition and Localization from Sparse Range or Tactile Data," *Int'l J. Robotics Research*, Vol. 3, No. 3, 1984, pp. 3-35.
98. R.C. Bolles and R.A. Cain, "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method," *Int'l J. Robotics Research*, Vol. 1, No. 3, 1982, pp. 57-82.
99. J.L. Turney, T.N. Mudge, and R.A. Volz, "Recognizing Partly Occluded Parts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 7, No. 4, 1985, pp. 410-421.

100. T. Knoll and R.C. Jain, "Recognizing Partially Visible Objects Using Feature Indexed Hypotheses," *IEEE Trans. Robotics and Automation*, Vol. 2, No. 1, 1986, pp. 3-13.
101. B. Bhanu and C. Ho, "CAD-Based 3-D Object Representation for Robot Vision," *Computer*, Vol. 20, No. 8, 1987, pp. 19-36.
102. P.J. Besl, J.B. Birch, and L.T. Watson, "Robust Window Operators," *Machine Vision and Applications*, Vol. 2, 1989, pp. 179-191.
103. R.M. Haralick et al, "Pose Estimation from Corresponding Point Data," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 19, No. 6, 1989, pp. 1426-1446.
104. L. Liu et al, "Application of Robust Sequential Edge Detection and Linking to Boundaries of Low Contrast Lesions in Medical Images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, IEEE CS Press, Los Alamitos, Calif., 1989, pp. 582-587.
105. K. Fukushima, "Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition," *Neural Networks*, Vol. 1, 1988, pp. 119-130.
106. N. Grossberg, *Neural Networks and Natural Intelligence*, MIT Press, Cambridge, Mass., 1988.
107. S. Haynes and R.C. Jain, "A Qualitative Approach for Recovering Depths in Dynamic Scenes," *IEEE Workshop on Computer Vision*, IEEE CS Press, Los Alamitos, Calif., 1987, pp. 66-71.
108. J. Aloimonos, "Visual Shape Computation," *Proc. IEEE*, Vol. 76, No. 8, IEEE Press, New York, N.Y., 1988, pp. 899-916.
109. A. Verri and T Poggio, "Motion Field and Optical Flow: Qualitative Properties," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 5, 1989, pp. 490-498.

"Generalizing the Hough Transform to Detect Arbitrary Shapes" by D.H. Ballard from *Pattern Recognition*, Volume 13, Number 2, 1981, pages 111-122.

GENERALIZING THE HOUGH TRANSFORM TO DETECT ARBITRARY SHAPES*

D. H. BALLARD

Computer Science Department, University of Rochester, Rochester, NY 14627, U.S.A.

(*Received* 10 *October* 1979; *in revised form* 9 *September* 1980; *received for publication* 23 *September* 1980)

Abstract—The Hough transform is a method for detecting curves by exploiting the duality between points on a curve and parameters of that curve. The initial work showed how to detect both analytic curves[1,2] and non-analytic curves,[3] but these methods were restricted to binary edge images. This work was generalized to the detection of some analytic curves in grey level images, specifically lines,[4] circles[5] and parabolas.[6] The line detection case is the best known of these and has been ingeniously exploited in several applications.[7,8,9]

We show how the boundaries of an *arbitrary* non-analytic shape can be used to construct a mapping between image space and Hough transform space. Such a mapping can be exploited to detect instances of that particular shape in an image. Furthermore, variations in the shape such as rotations, scale changes or figure–ground reversals correspond to straightforward transformations of this mapping. However, the most remarkable property is that such mappings can be composed to build mappings for complex shapes from the mappings of simpler component shapes. This makes the generalized Hough transform a kind of universal transform which can be used to find arbitrarily complex shapes.

Image processing
Parallel algorithms
Hough transform
Shape recognition
Pattern recognition

1. INTRODUCTION

In an image, the pertinent information about an object is very often contained in the shape of its boundary. Some appreciation of the importance of these boundary shapes in human vision can be gained from experiments performed on the human visual system, which have shown that crude encodings of the boundaries are often sufficient for object recognition[10] and that the image may be initially encoded as an 'edge image', i.e. an image of local intensity or color gradients. Marr[11] has termed this edge image a 'primal sketch' and suggested that this may be a necessary first step in image processing. We describe a very general algorithm for detecting objects of a specified shape from an image that has been transformed into such an edge representation. In that representation, sample points in the image no longer contain grey level information, but instead each sample point contains a magnitude and direction representing the severity and orientation of the local grey level change.

Operators that transform the image in such a way are known as edge operators, and many such operators are available, all based on different models of the local grey level changes. Two of the most used are the gradient operator (for example, see Prewitt[12]) and the Hueckel operator,[13] which model local grey level changes as a ramp and a step respectively.

Our generalized Hough algorithm uses edge information to define a mapping from the orientation of an edge point to a reference point of the shape. The reference point may be thought of as the origin of a local co-ordinate system for the shape. Then there is an easy way of computing a measure which rates how well points in the image are likely to be origins of the specified shape. Figure 1 shows a few graphic examples of the information used by the generalized Hough transform. Lines indicate gradient directions. A feature of the transform is that it will work even when the boundary is disconnected due to noise or occlusions. This is generally not true for other strategies which track edge segments.

The original algorithm by Hough[2] did not use

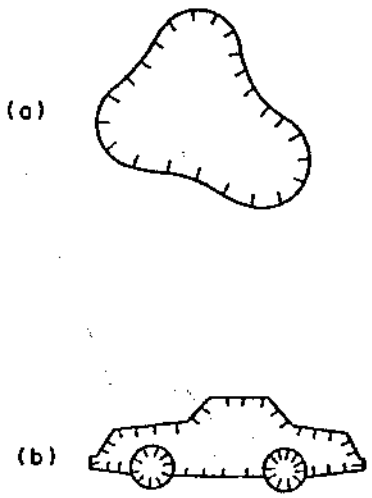


Fig. 1. Kinds of shapes detected with generalized Hough transform. (a) Simple shape; (b) composite shape.

*The research described in this report was supported in part by NIH Grant R23-HL-2153-01 and in part by the Alfred P. Sloan Foundation Grant 78-4-15.

orientation information of the edge, and was considerably inferior to later work using the edge orientation for parametric curves.[5,6,14] Shapiro[15,16,17] has collected a good bibliography of previous work as well as having contributed to the error analysis of the technique.

1.1 *Organization*

Section 2 describes the Hough transform for analytic curves. As an example of the parametric version of the transform, we use the ellipse. This example is very important due to the pervasiveness of circles in images, and the fact that a circle becomes an ellipse when rotated about an axis perpendicular to the viewing angle. Despite the importance of ellipses, not much work has used the Hough transform. The elliptical transform is discussed in detail in Section 3. Section 4 describes the generalized algorithm and its properties. Section 5 describes special strategies for implementing the algorithm and Section 6 summarizes its advantages.

2. THE HOUGH TRANSFORM FOR ANALYTIC CURVES

We consider analytic curves of the form $f(\mathbf{x}, \mathbf{a}) = 0$ where $\mathbf{x}$ is an image point and $\mathbf{a}$ is a parameter vector.

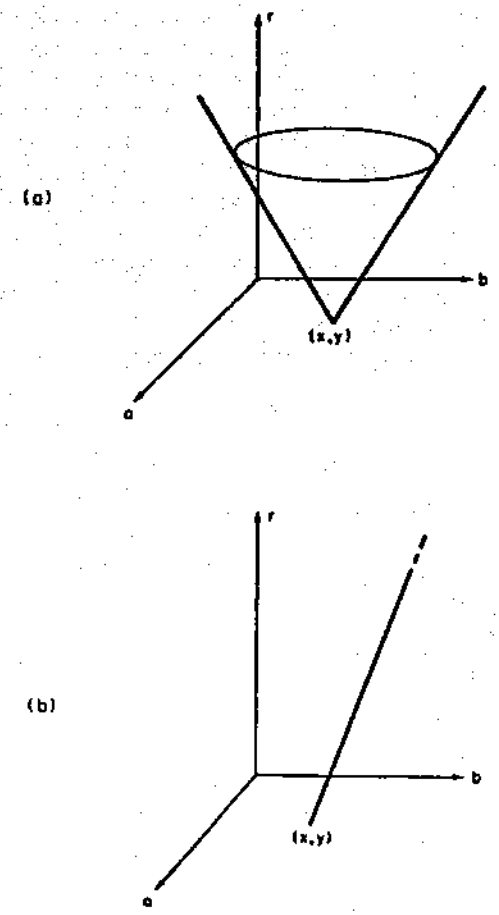


Fig. 2. (a) Locus of parameters with no directional information. (b) Locus of parameters with directional information.

To see how the Hough transform works for such curves, let us suppose we are interested in detecting circular boundaries in an image. In Cartesian coordinates, the equation for a circle is given by

$$(x-a)^2 + (y-b)^2 = r^2. \tag{1}$$

Suppose also that the image has been transformed into an edge representation so that only the magnitude of local intensity changes is known. Pixels whose magnitude exceeds some threshold are termed *edge pixels*. For each edge pixel, we can ask the question: if this pixel is to lie on a circle, what is the locus for the parameters of that circle? The answer is a right circular cone, as shown in Fig. 2(a). This can be seen from equation (1) by treating x and y as fixed and letting, a, b, and r vary.

The interesting result about this locus in parameter space is the following. If a set of edge pixels in an image are arranged on a circle with parameters a_0, b_0, and r_0, the resultant loci of parameters for each such point will pass through the same point (a_0, b_0, r_0) in parameter space. Thus many such right circular cones will intersect at a common point.

2.1 *Directional information*

We see immediately that if we also use the *directional* information associated with the edge, this reduces the parameter locus to a line, as shown in Fig. 2(b). This is because the center of the circle for the point (x, y) must lie r units along the direction of the gradient. Formally, the circle involves 3 parameters. By using the equation for the circle together with its derivative, the number of free parameters is reduced to one. Formally, what happens is the equation

$$\frac{df}{dx}(\mathbf{x}, \mathbf{a}) = 0$$

introduces a term dy/dx which is known since

$$\frac{dy}{dx} = \tan\left[\phi(\mathbf{x}) - \frac{\pi}{2}\right]$$

where $\phi(\mathbf{x})$ is the gradient direction. This suggests the following algorithm.

Hough algorithm for analytic curves in grey level images. For a specific curve $f(\mathbf{x}, \mathbf{a}) = 0$ with parameter vector $\mathbf{a}$, form an array $A(\mathbf{a})$, initially set to zero. This array is termed an accumulator array. Then for each edge pixel $\mathbf{x}$, compute all $\mathbf{a}$ such that $f(\mathbf{x}, a) = 0$ and $df/dx(\mathbf{x}, \mathbf{a}) = 0$ and increment the corresponding accumulator array entries:

$$A(\mathbf{a}) := A(\mathbf{a}) + 1.$$

After each edge pixel $\mathbf{x}$ has been considered, local maxima in the array A correspond to curves of f in the image.

If only the equation $f(\mathbf{x}, \mathbf{a}) = 0$ is used, the cost of the computation is exponential in the number of parameters minus one, that is, where m parameters each have M values, the computation is proportional to

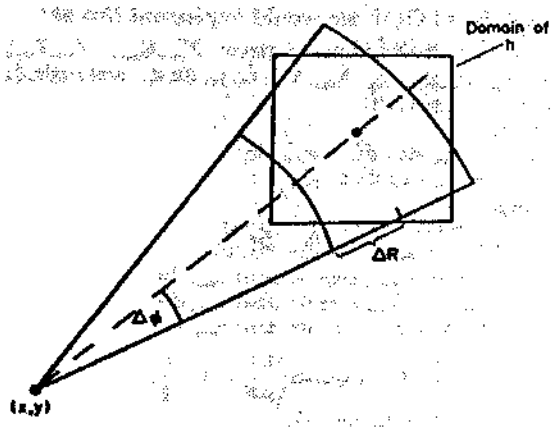


Fig. 3. Using convolution templates to compensate for errors.

M^{m-1}. This is because the equation of the curve can be used to determine the last parameter. The use of gradient directional information saves the cost of another parameter making the total effort proportional to M^{m-2}, for $m \geq 2$.

2.2 *Compensating for errors*

A problem arises in detecting maxima in the array $A(\mathbf{a})$. Many sources of error effect the computation of the parameter vector $\mathbf{a}$ so that in general many array locations in the vicinity of the ideal point $\mathbf{a}$ are incremented instead of the point itself. One way of handling this problem is to use a formal error model on the incrementation step. This model would specify a set of nearby points instead of a single point. Shapiro[15–18] has done extensive work on this subject. Another solution to this problem is to replace uncompensated accumulator values by a function of the values themselves and nearby points after the incrementation step. The effect of this operation is to smooth the accumulator array. We show that, under the assumption of isotropic errors, these methods are equivalent.

Returning to the initial example of detecting circles, the smoothing of the accumulator array is almost equivalent to the change in the incrementing procedure we would use to allow for uncertainties in the gradient direction ϕ and the radius r. If we recognized these uncertainties as:

$$\phi(\mathbf{x}) \pm \Delta\phi$$
$$r \pm \Delta r(r)$$

we would increment all values of $\mathbf{a}$ which fall within the shaded band of Fig. 3. We let Δr increase with r so that uncertainties are counted on a percentage basis. Figure 3 shows the two-dimensional analog of the general three-dimensional case.

Suppose we approximate this procedure by incrementing all values of $\mathbf{a}$ which fall inside the square domain centered about the nominal center shown in Fig. 3, according to some point spread function h. After the first contributing pixel which increments center $\mathbf{a}_0$ has been taken into account, the new accumulator array contents A will be given by

$$A(\mathbf{a}) = h(\mathbf{a}-\mathbf{a}_0) \tag{2}$$

where $\mathbf{a} = (a_1, a_2, r)$ and $\mathbf{a}_0 = (a_{10}, a_{20}, r_0)$. If we include all the contributing pixels for that center, denoted by C, the accumulator is

$$A(\mathbf{a}) = C(\mathbf{a}_0)h(\mathbf{a}-\mathbf{a}_0). \tag{3}$$

Finally for all incremented centers, we sum over $\mathbf{a}_0$:

$$A(\mathbf{a}) = \sum_{\mathbf{a}_0} C(\mathbf{a}_0)h(\mathbf{a}-\mathbf{a}_0). \tag{4}$$

But $C(\mathbf{a}_0) = A(\mathbf{a}_0)$, so that

$$\begin{aligned} A(\mathbf{a}) &= \sum_{\mathbf{a}_0} A(\mathbf{a}_0)h(\mathbf{a}-\mathbf{a}_0) \\ &= A^* h \\ &\equiv A_s(\mathbf{a}). \end{aligned} \tag{5}$$

Thus within the approximation of letting the square represent the shaded band shown in Fig. 3, the smoothing procedure is equivalent to an accommodation for uncertainties in the gradient direction and radius.

3. AN EXAMPLE: ELLIPSES

The description of the algorithm in Section 2.1 is very terse and its implementation often requires considerable algebraic manipulation. We use the example of finding ellipses to show the kinds of calculation which must be done. Ellipses are an important example, as circles, which are a ubiquitous part of many everyday objects, appear as ellipses when viewed from a distant, oblique angle.

We use the center of the ellipse as a reference point and assume that it is centered at x_0, y_0 with major and minor diameters a and b. For the moment, we will assume that the ellipse is oriented with its major axis parallel to the x-axis. Later we will relax this requirement by introducing an additional parameter for arbitrary orientations. For the moment, assume a and b are fixed. Then the equation of the ellipse is:

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1. \tag{6}$$

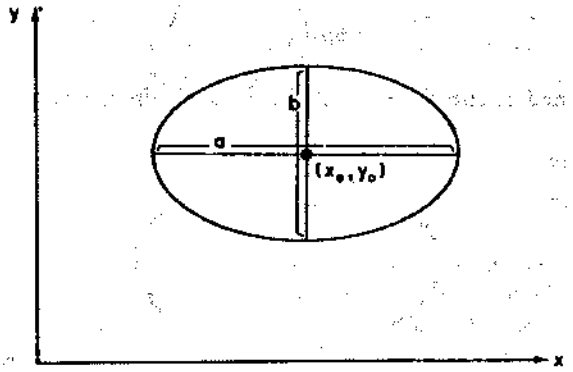


Fig. 4. Parametrization of an ellipse with major axis parallel to x-axis.

Let $X = x - x_0$, $Y = y - y_0$, then

$$\frac{X^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{7}$$

Differentiating with respect to X

$$\frac{2X}{a^2} + \frac{2Y}{b^2}\frac{dY}{dX} = 0. \tag{8}$$

But dY/dX is known from the edge pixel information! Let $dY/dX = \xi$, then from (8)

$$X^2 = \left(\frac{a^2}{b^2}\xi\right)^2 Y^2. \tag{9}$$

Substituting in (7)

$$\frac{Y^2}{b^2}\left(1 + \frac{a^2}{b^2}\xi^2\right) = 1 \tag{10}$$

$$Y = \pm \frac{b^2}{\sqrt{\left(1 + \frac{a^2}{b^2}\xi^2\right)}} \tag{11}$$

so that

$$X = \pm \frac{a^2}{\sqrt{\left(1 + \frac{b^2}{a^2\xi^2}\right)}} \tag{12}$$

and finally, given a, b, x, y and dY/dX, we can determine x_0 and y_0 as:

$$x_0 = x \pm \frac{a^2}{\sqrt{\left(1 + \frac{b^2}{a^2\xi^2}\right)}} \tag{13}$$

$$y_0 = y \pm \frac{b^2}{\sqrt{\left(1 + \frac{a^2\xi^2}{b^2}\right)}}. \tag{14}$$

The four solutions correspond to the four quadrants, as shown in Fig. 5. The appropriate quadrant can be found from the gradient by testing the signed differences dY and dX.

The final step is to handle rotations by introducing a fifth parameter θ. For an arbitrary θ, we calculate (X, Y) using

$$\xi = \tan\left(\phi - \theta - \frac{\pi}{2}\right)$$

and rotate these (X, Y) by θ to obtain the correct (x_0, y_0). In ALGOL we would implement this as:

```
procedure HoughEllipse (integer X_min, X_max, Y_min, Y_max,
θ_min, θ_max, a_min, a_max, b_min, b_max, x, y, x_0, y_0, dx, dy; real angle, ξ;
integer array A, P);
begin;
for x := x_min step dx to x_max do
for y := y_min step dy to y_max do
  begin
    dX := P(x+delta, y) − P(x, y);
    dY := P(x, y+delta) − P(x, y);
      for a := a_min step da until a_max do
      for b := b_min step db until b_max do
      for θ := θ_min step dθ until θ_max do
        begin;
          angle := arctan(dY/dX) − θ − π/2;
          ξ := tan(angle);
          dx := Sign X(dX, dY) a²/√(1 + b²/(a²ξ²));
          dy := Sign Y(dX, dY) b²/√(1 + a²/b²)²;
          Rotate-by-Theta(dx, dy);
          x_0 := x + dx;
          y_0 := y + dy;
          A(x_0, y_0, θ, a, b) := A(x_0, y_0, θ, a, b) + 1;
        end;
end.
```

Notice that to determine the appropriate formulae for an arbitrary orientation angle θ, we need only rotate the gradient angle and the offsets dx and dy. Sign X and Sign Y are functions which return ± 1 depending on the quadrant determined by dX and dY.

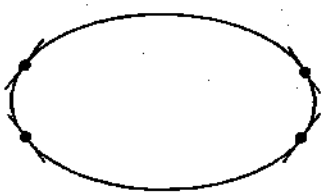


Fig. 5. Four reference point solutions resolvable with gradient quadrant information.

3.1 *Parameter space image space trade-offs*

Tsuji and Matsumoto[19] recognized that a decreased computational effort in parameter space could be traded for an increased effort in edge space. It is our intent to place these ideas on a formal footing. Later we will see that the same kind of trade-off is potentially available for the case of arbitrary shapes, but is impractical to implement.

An ellipse has five parameters. Referring to the basic algorithm in Section 2.1, we use the equation for the ellipse together with its derivative to solve for two of these parameters as a function of the other three. Thus the algorithm examines every edge point and uses a three-dimensional accumulator array so that the computations are of order $0(ed^3)$. Here e is the number of edge pixels and we are assuming d distinct values for each parameters. Suppose we use pairs of edge points in the algorithm. This results in four equations, two involving the equation for an ellipse evaluated at the different points and two for the related derivatives. This leaves one free parameter. Thus the resultant computational effort is now $0(e^2d)$. The detailed derivation of this form of the Hough algorithm is presented in the Appendix.

If parameter space can be highly constrained so that the set of plausible values is small, then the former technique will be more efficient, whereas if there are

Table 1. Analytic curves described in terms of the generalized shape parameters x_r, y_r, S_x, S_y, θ

Analytic form	Parameters	Equation
Line	S, θ	$x\cos\theta + y\sin\theta = S$
Circle	x_r, y_r, S	$(x-x_r)^2 + (y-y_r)^2 = S^2$
Parabola	x_r, y_r, S_x, θ	$(y-y_r)^2 = 4S_x(x-x_r)$*
Ellipse	$x_r, y_r, S_x, S_y, \theta$	$\frac{(y-y_r)^2}{S_y^2} + \frac{(x-x_r)^2}{S_x^2} = 1$*

* Plus rotation by θ.

relatively few edges and large variations in parameters, the latter will be more efficient.

4. GENERALIZING THE HOUGH TRANSFORM

To generalize the Hough algorithm to non-analytic curves we define the following parameters for a generalized shape:

$$\mathbf{a} = \{\mathbf{y}, \mathbf{s}, \theta\},$$

where $\mathbf{y} = (x_r, y_r)$ is a reference origin for the shape, θ is its orientation, and $\mathbf{s} = (s_x, s_y)$ describes two orthogonal scale factors. As before, we will provide an algorithm for computing the best set of parameters $\mathbf{a}$ for a given shape from edge pixel data. These parameters no longer have equal status. The reference origin location, $\mathbf{y}$, is described in terms of a table of possible edge pixel orientations. The computation of the additional parameters $\mathbf{s}$ and θ is then accomplished by straightforward transformations to this table. [To simplify the development slightly, and because of its practical significance, we will work with the four-dimensional sunspace $\mathbf{a} = (y, s, \theta)$, where s is a scalar.]

In a sense this choice of parameters includes the previous analytic forms to which the Hough transform has been applied. Table 1 shows these relationships.

4.1 *Earlier work: arbitrary shapes in binary edge images*

Merlin and Farber[3] showed how to use a Hough algorithm when the desired curves could not be described analytically. Each shape must have a specific reference point. Then we can use the following algorithm for a shape with boundary points B denoted by $\{\mathbf{x}_B\}$ which are relative to some reference origin $\mathbf{y}$.

Merlin–Farber Hough algorithm: non-analytic curves with no gradient direction information $\mathbf{a} = \mathbf{y}$. Form a two-dimensional accumulator array $A(\mathbf{a})$ initialized to zero. For each edge pixel $\mathbf{x}$ and each boundary point $\mathbf{x}_B$, compute $\mathbf{a}$ such that $\mathbf{a} = \mathbf{x} - \mathbf{x}_B$ and increment $A(\mathbf{a})$. Local maxima in $A(\mathbf{a})$ correspond to instances of the shape in the image.

Note that this is merely an efficient implementation of the convolution of the shape template where edge pixels are unity and others are zero with the corresponding image, i.e.,

$$A(\mathbf{x}) = T(\mathbf{x}) * S(\mathbf{x}) \tag{15}$$

where E is the binary edge image defined by

$$E(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{x} \text{ is an edge pixel} \\ 0 \text{ otherwise} \end{cases}$$

and $T(\mathbf{x})$ is the shape template consisting of ones where $\mathbf{x}$ is a boundary point and zeros otherwise, i.e.,

$$T(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{x} \text{ is in } B \\ 0 \text{ otherwise} \end{cases}.$$

This result is due to Sklansky.[20]

The Merlin–Farber algorithm is impractical for real image data. In an image with a multitude of edge pixels, there will be many false instances of the desired shape due to coincidental pixel arrangements. Nevertheless, it is the logical precursor to our generalized algorithm.

4.2 *The generalization to arbitrary shapes*

The key to generalizing the Hough algorithm to arbitrary shapes is the use of directional information. Directional information, besides making the algorithm faster, also greatly improves its accuracy. For example, if the directional information is not used in the circle detector, any significant group of edge points with quite different directions which lie on a circle will be detected. This can be appreciated by comparing Figs 2(a) and 2(b).

Consider for a moment the circular boundary detector with a fixed radius r_0. Now for each gradient point $\mathbf{x}$ with direction ϕ, we need only increment a single point $\mathbf{x}+\mathbf{r}$. For the circle:

$$|\mathbf{r}| = r_0 \tag{16}$$

$$\text{Angle}(\mathbf{r}) = \phi(\mathbf{x}). \tag{17}$$

Now suppose we have an arbitrary shape like the one shown in Fig. 6. Extending the idea of the circle detector with fixed radius to this case, for each point $\mathbf{x}$ on the boundary with gradient direction ϕ, we increment a point $\mathbf{a} = \mathbf{x}+\mathbf{r}$. The difference is that now $\mathbf{r} = \mathbf{a}-\mathbf{x}$ which, in general, will vary in magnitude and direction with different boundary points.

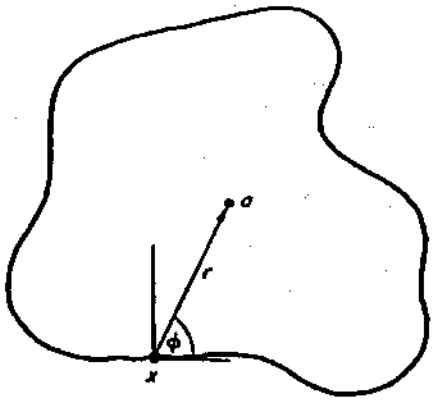


Fig. 6. Geometry for generalized Hough transform.

The fact that r varies in an arbitrary way means that the generalized Hough transform for an arbitrary shape is best represented by a table which we call the *R*-table.

4.3 *The R-table*

From the above discussion, we can see that the *R*-table is easily constructed by examining the boundary points of the shape. The construction of the table is accomplished as follows.

Algorithm for constructing an R*-table.* Choose a reference point y for the shape. For each boundary point x, compute $\phi(\mathbf{x})$ the gradient direction and $\mathbf{r} = \mathbf{y} - \mathbf{x}$. Store r as a function of ϕ.

Notice that the mapping the table represents is vector-valued and, in general, an index ϕ may have many values of r. Table 2 shows the form of the *R*-table diagrammatically.

The *R*-table is used to detect instances of the shape *S* in an image in the following manner.

Generalized Hough algorithm for single shapes. For each edge pixel x in the image, increment all the corresponding points $\mathbf{x}+\mathbf{r}$ in the accumulator array *A* where r is a table entry indexed by ϕ, i.e., $\mathbf{r}(\phi)$. Maxima in *A* correspond to possible instances of the shape *S*.

4.4 *Examples*

Some simple shapes are rotation-invariant, that is, the entries in the incrementation table are invariant functions of the gradient direction ϕ. Figure 7(a) shows an example for washers (or bagels). Here there are exactly two entries for each ϕ, one r units in the gradient direction and one R units in the direction opposite to the gradient direction. In another case the entries may be a simple function of ϕ. Figure 7(b) shows such an example; hexagons. Irrespective of the orientation of the edge, the reference point locus is on a line of length l parallel to the edge pixel and $(3/2)l$ units away from it.

Table 2. *R*-table format

i	ϕ_i	R_{ϕ_i}
0	0	$\{\mathbf{r} \mid \mathbf{a} - \mathbf{r} = \mathbf{x},\ \mathbf{x} \text{ in } B,\ \phi(\mathbf{x}) = 0\}$
1	$\Delta\phi$	$\{\mathbf{r} \mid \mathbf{a} - \mathbf{r} = \mathbf{x},\ \mathbf{x} \text{ in } B,\ \phi(\mathbf{x}) = \Delta\phi\}$
2	$2\Delta\phi$	$\{\mathbf{r} \mid \mathbf{a} - \mathbf{r} = \mathbf{x},\ \mathbf{x} \text{ in } B,\ \phi(\mathbf{x}) = 2\Delta\phi$
...	...	...

Another example is shown in Fig. 8. Here the points on the boundary of the shape are shown in Fig. 8(a). A reference point is selected and used to construct the *R*-table. Figure 8(b) shows a synthetic image of four different shapes and Fig. 8(c) shows the portion of the accumulator array for this image which has the correct values of orientation and scale. It is readily seen that edge points on the correct shape have incremented the same point in the accumulator array, whereas edge points on the other shapes have incremented disparate points.

4.5 *R-table properties and the general notion of a shape*

Up to this point we have considered shapes of fixed orientation and scale. Thus the accumulator array was two-dimensional in the reference point co-ordinates. To search for shapes of arbitrary orientation θ and scale s we add these two parameters to the shape description. The accumulator array now consists of four dimensions corresponding to the parameters $(\mathbf{y}, s, \theta)$. The *R*-table can also be used to increment this larger dimensional space since different orientations and scales correspond to easily-computed transformations of the table. Additionally, simple transformations to the *R*-table can also account for figure–ground reversals and changes of reference point.

We denote a particular *R*-table for a shape *S* by $R(\phi)$. *R* can be viewed as a multiply-vector-valued function. It is easy to see that simple transformations to this table will allow it to detect scaled or rotated instances of the same shape. For example if the shape is scaled by s and this transformation is denoted by T_s, then

$$T_s[R(\phi)] = sR(\phi) \tag{18}$$

i.e., all the vectors are scaled by s. Also, if the object is rotated by θ and this transformation is denoted by T_θ, then

$$T_\theta[R(\phi)] = \mathrm{Rot}\{R[(\phi - \theta) \bmod 2\pi], \theta\} \tag{19}$$

i.e., all the indices are incremented by $-\theta$ modulo 2π, the appropriate vectors r are found, and then they are rotated by θ.

To appreciate that this is true, refer to Fig. 9. In this figure an edge pixel with orientation ϕ may be considered as corresponding to the boundary point $\mathbf{x}_A$, in which case the reference point is $\mathbf{y}_A$. Alternatively, the edge pixel may be considered as $\mathbf{x}_B$ on a rotated instance of the shape, in which case the reference point is at $\mathbf{y}_B$ which can be specified by translating $\mathbf{r}_A$ to $\mathbf{x}_B$ and rotating it through $+\Delta\theta$.

Figure–ground intensity reversals can also be taken into account via a simple *R*-table modification. The indices in the table are changed from ϕ to $(\phi + \pi) \bmod 2\pi$. Of course

$$T_{s_\theta}\{T_{s_\theta}[R(\phi)]\} = R(\phi)$$

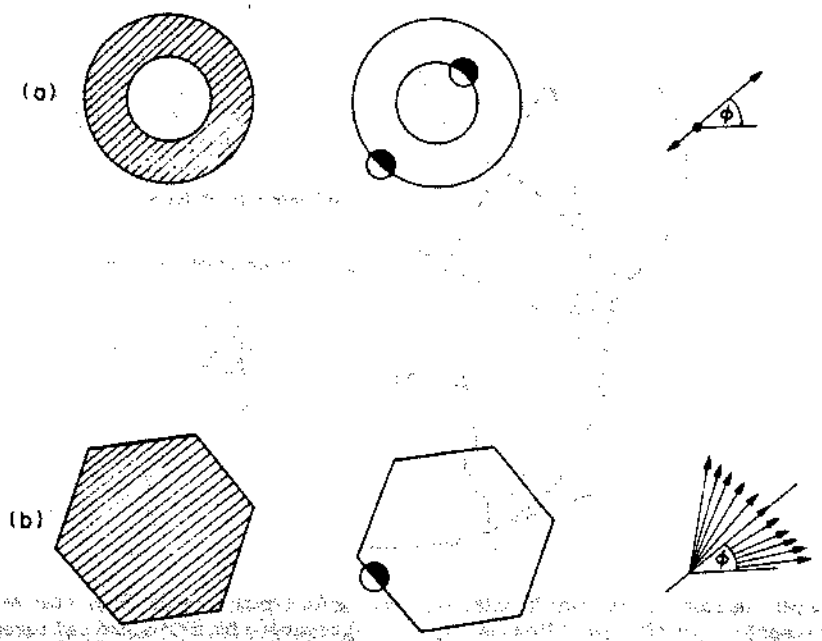


Fig. 7. Simple examples using R-tables; (a) washers; (b) hexagons.

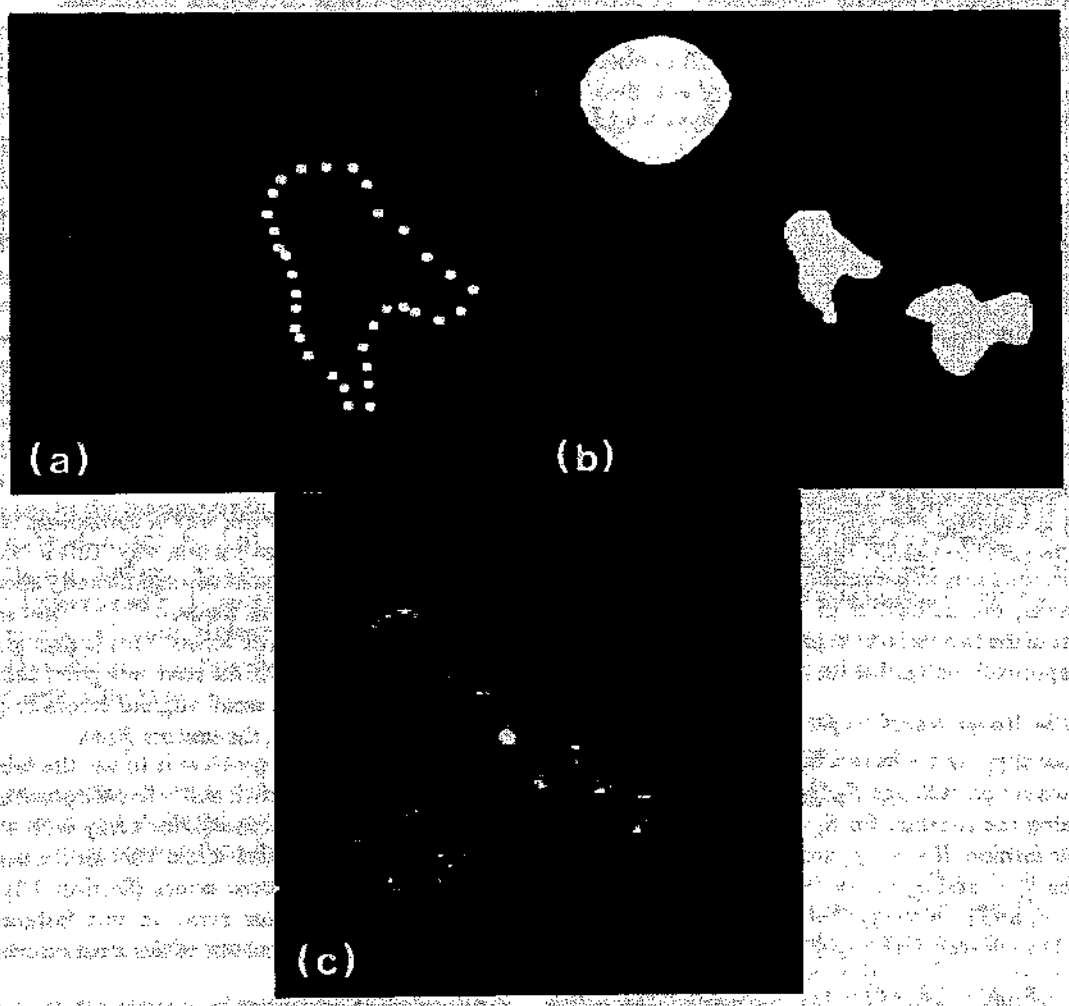


Fig. 8. An example. (a) Points on a shape used to encode R-table. (b) Image containing shape. (c) A plane through the accumulator array $A(x_r, y_r, S_0, \theta_0)$, where S_0 and θ_0 are appropriate for the shape in the image ($S_0 = 64$, $\theta_0 = 0$).

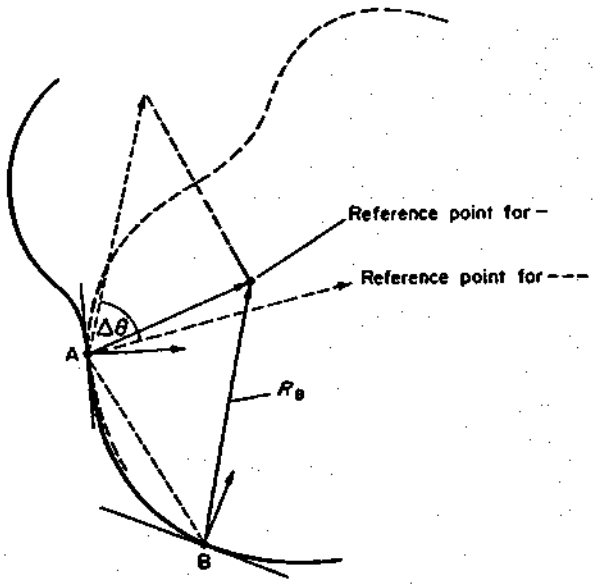


Fig. 9. Construction for visualizing the R-table transformation for a rotation by $\Delta\theta$. Point A can be viewed as: (1) on the shape (——), or (2) as point B on the shape (- - - -), rotated by $\Delta\theta$. If (2) is used then the appropriate R is obtained by translating R_θ to A and rotating it by $\Delta\theta$ as shown.

where T_{f_θ} denotes the figure–ground transformations.

Another property which will be useful in describing the composition of generalized Hough transforms is the change of reference point. If we want to choose a new reference point y' such that $y - y' = r$ then the modification to the R-table is given by $R(\phi) + r$, i.e. r is added to each vector in the table.

4.6 *Using pairs of edges*

We can also entertain the idea of using pairs of edge pixels to reduce the effort in parameter space. Using the R-table and the properties of the previous section, each edge pixel defines a surface in the four-dimensional accumulator space of $a = (y, s, \theta)$. Two edge pixels at different orientations describe the same surface rotated by the same amount with respect to θ. Points where these two surfaces intersect (if any) correspond to possible parameters a for the shape. Thus in a similar manner to Section 3.1, it is theoretically possible to use the two points in image space to reduce the locus in parameter space to a single point. However, the difficulties of finding the intersection points of the two surfaces in parameter space will make this approach unfeasible for most cases.

4.7 *The Hough transform for composite shapes*

Now suppose we have a composite shape S which has two subparts S_1 and S_2. This shape can be detected by using the R-tables for S_1 and S_2 in a remarkably simple fashion. If y, y_1, y_2 are the reference points for shapes S, S_1 and S_2 respectively, we can compute $r_1 = y - y_1$ and $r_2 = y - y_2$. Then the composite generalized Hough transform $R_S(\phi)$ is given by

$$R_S(\phi) = [R_{S_1}(\phi) + r_1] \cup [R_{S_2}(\phi) + r_2] \quad (20)$$

which means that for each index value ϕ, r_1 is added to $R_{S_1}(\phi)$, r_2 is added to $R_{S_2}(\phi)$, and the union of these sets is stored in $R_S(\phi)$. Equation 20 is very important as it represents a way of composing transforms.

In a similar manner we can define shapes as the difference between tables with common entries, i.e.,

$$R_S = R_{S_1} \dot{-} R_{S_2} \quad (21)$$

means the shape S defined by S_1 with the common entries with S_2 deleted. The intersection operation is defined similarly. The primary use of the union operation is to detect shapes which are composites of simpler shapes. However, the difference operation also serves a useful function. Using it, R-tables which explicitly differentiate between two similar kinds of shapes can be constructed. An example would be differentiating between the washers and hexagons discussed earlier.

4.8 *Building convolution templates*

While equation (20) is one way of composing Hough transforms, it may not be the best way. This is because the choice of reference point can significantly affect the accuracy of the transform. Shapiro[15,16,17] has shown this, emphasizing analytic forms. This is also graphically shown in Fig. 10. As the reference point becomes distant from the shape, small angular errors in ϕ can produce large errors in the vectors $R(\phi)$.

One solution to this problem is to use the table for each subshape with its own best reference point and to smooth the resultant accumulator array with a composite smoothing template. Recall that for the case of a single shape and isotropic errors (Section 2.2), convolving the accumulator array in this fashion was equivalent to taking account of the errors during the incrementation.

Where $h_i(y_i)$ denotes the smoothing template for reference point y_i of shape S_i the composite convolution template is given by

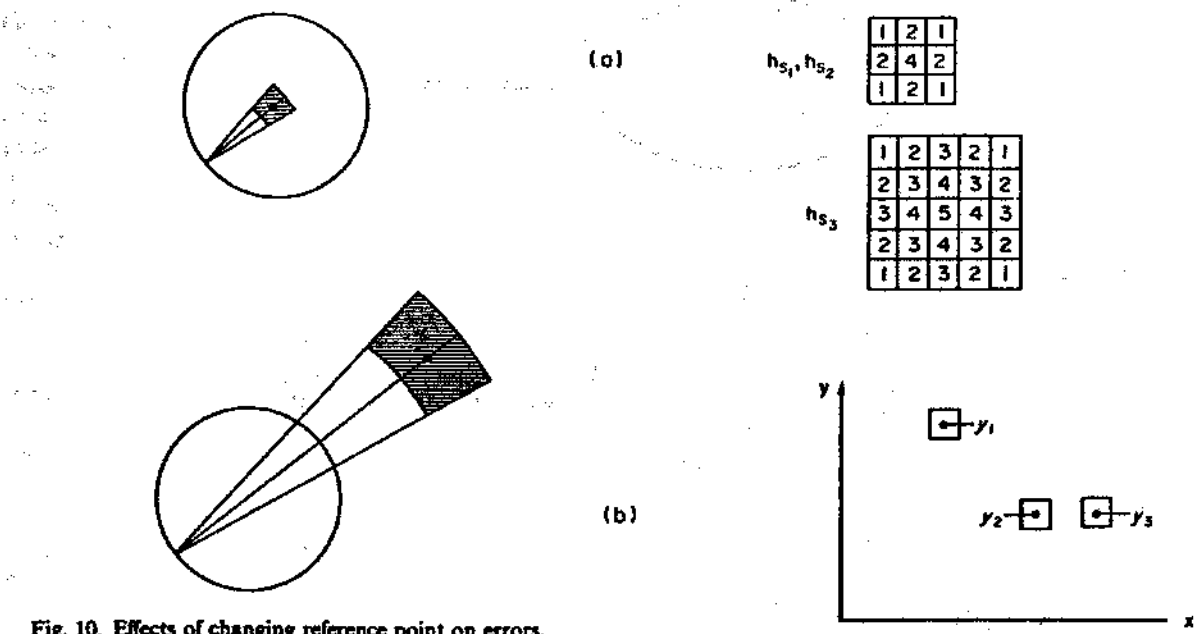


Fig. 10. Effects of changing reference point on errors.

$$H(y) = \sum_{i=1}^{N} h_i(y - y_i). \tag{22}$$

So finally, we have the following algorithm for the detection of a shape S which is the composite of subparts $S_1 \ldots S_N$.

Generalized Hough algorithm for composite shapes. 1. For each edge point with direction ϕ and for each value of scale s and orientation θ, increment the corresponding points $\mathbf{x} + \mathbf{r}$ in A where $\mathbf{r}$ is in

$$R_s(\phi) = T_s\left\{T_\theta\left[\bigcup_{k=1}^{N} R_{S_k}(\phi)\right]\right\}.$$

2. Maxima in $A_s = A^*H$ correspond to possible instances of the shape S. Figure 11 shows a simple example of how templates are combined.

If there are n edge pixels and M points in the error point spread function template, then the number of additions in the incrementation procedure is M. Thus this method might at first seem superior to the convolution method, which requires approximately n^2M additions and multiplications where $M < n^2$, the total number of pixels. However, the following heuristic is available for the convolution since A is typicallly very sparse. Compute

$$A_s(\mathbf{a}) \quad \text{only if } A(\mathbf{a}) > 0. \tag{23}$$

This in practice is very effective, although it may introduce errors if the appropriate index has a zero value and is surrounded by high values.

5. INCREMENTATION STRATEGIES

If we use the strategy of incrementing the accumulator array by unity, then the contents of the accumulator array are approximately proportional to the perimeter of the shape that is detectable in the image.

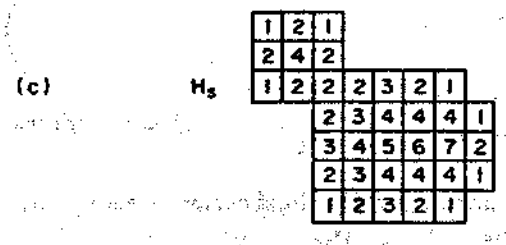


Fig. 11. Example of composite smoothing template construction. (a) Convolution templates for shapes S_1, S_2, S_3. (b) Relationships between reference points y_1, y_2, and y_3 in composite shape S. (c) Combined smoothing template H as a function of h_1, h_2, and h_3 and y_1, y_2, and y_3.

This strategy is biased towards finding shapes where a large portion of the perimeter is detectable. Several different incrementation strategies are available, depending on the different quality of image data. If shorter, very prominent parts of the perimeter are detected, as might be the case in partially occluded objects, then an alternative strategy of incrementing by the gradient modulus value might be more successful, i.e.,

$$A(\mathbf{a}) := A(\mathbf{a}) + g(\mathbf{x}). \tag{24}$$

Of course the two strategies can be combined, e.g.,

$$A(\mathbf{a}) := A(\mathbf{a}) + g(\mathbf{x}) + c \tag{25}$$

where c is a constant.

Another possibility is the use of local curvature information in the incrementation function. Using this strategy, neighboring edge pixels are examined to calculate approximate curvature, K. This requires a more complicated operator than the edge operators we have considered, and complicates the table. Now along with each value of r the corresponding values of curvature must be stored. Then the incrementation

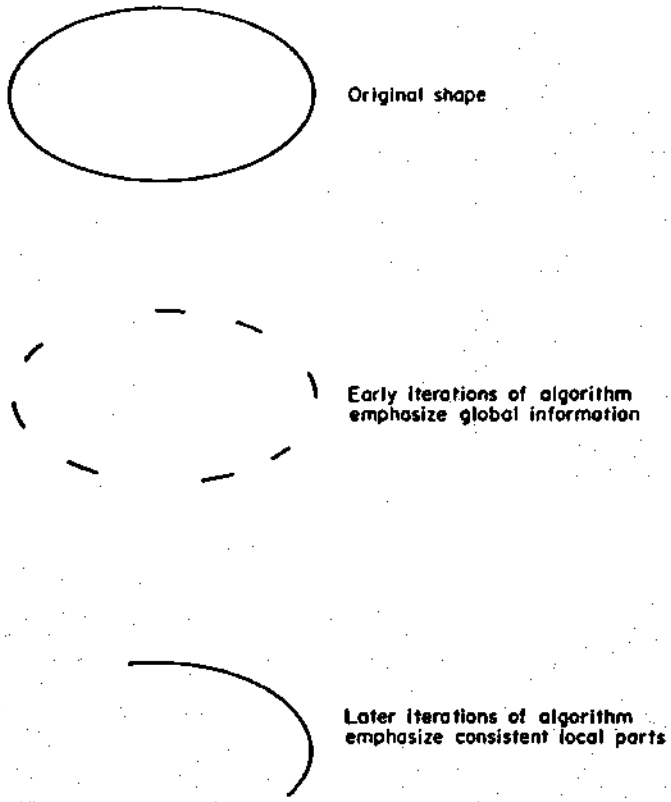


Fig. 12. Dynamic Hough transform.

weights 'informative' high local curvature edge pixels as follows:

$$A(\mathbf{a}) := A(\mathbf{a}) + K. \tag{26}$$

5.1 *Weighting locally consistent information*

Under certain circumstances we may want to weight local information that is consistent. For example, in searching for the boundary of that object, a connected set of edges conforming to the object may be more important than a set of unconnected edges. Figure 12 shows this example. Figure 12(a) might arise in situations with very noisy data. Figure 12(b) is an example where an object is occluded by another object. Wechsler and Sklansky,[6] in the analytic formulation, successfully used the related strategy of increasing the incrementation factor if there were also neighboring edge pixels with the same edge direction. However, we would like to measure local consistency in parameter space.

A simple strategy for handling this case is to explicitly record the reference points for each edge pixel during a first pass. Then on a second pass edge pixels can increment by more than unity if neighboring edge pixels are incrementing the same reference point.

A more complicated strategy is to search for connected curve segments in image space which have compatible parameters. Such an algorithm, based on dynamic programming, is described in Ballard and Sklansky.[14] The appropriate objective function for a curve segment would be

$$h(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \sum_{k=1}^{n} g(\mathbf{x}_k) + \sum_{k=1}^{n-1} q(\mathbf{x}_k, \mathbf{x}_{k+1}) \tag{27}$$

where

$$g(\mathbf{x}_k) = \text{the gradient magnitude} \tag{28}$$

and

$$q(\mathbf{x}_k, \mathbf{x}_{k+1}) = 0 \text{ if } |\phi(\mathbf{x}_n) - \phi(\mathbf{x}_{k-1})|_{\text{mod}\,\pi} \text{ is small and } -\infty \text{ otherwise} \tag{29}$$

In the dynamic programming algorithm, at each iteration step we can build longer compatible curves from all the edge points. Thus the incrementation function for a point x would represent the longest compatible curve from that point. (If a longer curve cannot be built at any iteration, we can easily find this out.)

In a parallel implementation of this algorithm the contents of the accumulator array could be made to vary dynamically. Initially the contents would reflect global information, but with successive iterations the contents would be weighted in favor of consistent, local information.

5.2 *More complex strategies*

When searching for a composite object, different parts may have different importance. This is readily accommodated by associating a weight w_i with each table R_{S_i} so that each entry in R_{S_i} increments by a

factor w_i instead of unity.

The composite object may be searched for in a sequential manner. Applying the table sequentially could greatly improve the efficiency of the computations by limiting areas for subsequent suitable incrementations. Furthermore, standard methods[21,22] could be used to stop the process once the shape had been located to the desired confidence level.

Even more complex strategies are possible wherein the process is integrated into a larger system. Here contextual information can be used to relegate all the previous operations including (a) building composite templates, (b) choosing weights, (c) choosing application sequences, and (d) adjusting weights in new contexts.

6. CONCLUSIONS

We have described a method for detecting instances of a shape S in an image which is a generalization of the Hough transform. This transform is a mapping from edge space to accumulator space such that instances of S produce local maxima in accumulator space. This mapping is conveniently described as a table of edge-orientation reference-point correspondence termed an R-table. This method has the following properties.

1. Scale changes, rotations, figure–ground reversals, and reference point translation of S can be accounted for by straightforward modifications to the R-table.

2. Given the boundary of the shape, its R-table can be easily constructed and requires a number of operations proportional to the number of boundary points.

3. Shapes are stored as canonical forms; instances of shapes are detected by knowing the transformation from the canonical form to the instance. If this transformation is not known then all plausible transformations must be tried.

4. If a shape S is viewed as a composite of several subparts $S_1 \ldots S_n$ then the generalized Hough transform R-table for S can be simply constructed by combining the R-tables for $S_1 \ldots S_n$.

5. A composite shape S may be efficiently detected in a sequential manner by adding the R-tables for the subparts S_i incrementally to the detection algorithm until a desired confidence level is reached.

6. The accumulator table values can be weighted in terms of locally consistent information.

7. The importance of a subshape S_i may be regulated by associating a weight w_i with the R-table.

8. Last but not least, the generalized Hough transform is a parallel algorithm.

Future work will be directed towards characterizing the computational efficiency of the algorithm and exploring its feasibility as a model of biological perception.

Acknowledgements – Portions of this paper benefitted substantially from discussions with Ken Sloan and Jerry Feldman. Special thanks go to R. Peet and P. Meeker for typing this manuscript. The work herein was supported by National Institutes of Health grant R23-HL21253-02.

REFERENCES

1. R. O. Duda and P. E. Hart, Use of the Hough transform to detect lines and curves in pictures, *Communs Ass. comput. Mach.* **15**, 11–15 (1975).
2. P. V. C. Hough, Method and means for recognizing complex patterns, U.S. Patent 3069654 (1962).
3. P. M. Merlin and D. J. Farber, A parallel mechanism for detecting curves in pictures, *IEEE Trans. Comput.* **C24**, 96–98 (1975).
4. F. O'Gorman and M. B. Clowes, Finding picture edges through collinearity of feature points, Proc. 3rd Int. Joint Conf. Artificial Intelligence, pp. 543–555 (1973).
5. C. Kimme, D. H. Ballard and J. Sklansky, Finding circles by an array of accumulators, *Communs Ass. comput. Mach.* **18**, 120–122 (1975).
6. H. Wechsler and J. Sklansky, Automatic detection of ribs in chest radiographs, *Pattern Recognition* **9**, 21–30 (1977).
7. S. A. Dudani and A. L. Luk, Locating straight-line edge segments on outdoor scenes, Proc. IEEE Computer Society on Pattern Recognition and Image Processing, Rensselaer Polytechnic Institute (1977).
8. C. L. Fennema and W. B. Thompson, Velocity determination in scenes containing several moving objects, Technical Report, Central Research Laboratory, Minnesota Mining and Manufacturing Co. St. Paul (1977).
9. J. R. Kender, Shape from texture: a brief overview and a new aggregation transform, Proc., DARPA Image Understanding Workshop, pp. 79–84. Pittsburgh, November (1978).
10. F. Attneave, Some informational aspects of visual perception, *Psychol. Rev.* **61**, 183–193 (1954).
11. D. Marr, Analyzing natural images: a computational theory of texture vision, MIT-AI-Technical Report 334, June (1975).
12. J. M. S. Prewitt, Object enhancement and extraction, *Picture Processing and Psychopictorics*, B. S. Lipkin and A. Rosenfeld, eds. Academic Press, New York (1970).
13. M. Hueckel, A local visual operator which recognizes edges and lines, *J. Ass. comput. Mach.* **20**, 634–646 (1973).
14. D. H. Ballard and J. Sklansky, A ladder-structured decision tree for recognizing tumors in chest radiographs, *IEEE Trans. Comput.* **C25**, 503–513 (1976).
15. S. D. Shapiro, Properties of transforms for the detection of curves in noisy pictures, *Comput. Graphics Image Process.* **8**, 219–236 (1978).
16. S. D. Shapiro, Feature space transforms for curve detection, *Pattern Recognition* **10**, 129–143 (1978).
17. S. D. Shapiro, Generalization of the Hough transform for curve detection in noisy digital images, *Proc. 4th* Int. Joint Conf. *Pattern Recognition*, pp. 710–714. Kyoto, Japan, November (1978).
18. S. D. Shapiro, Transformation for the computer detection of curves in noisy pictures, *Comput. Graphics Image Process.* **4**, 328–338 (1975).
19. S. Tsuji and F. Matsumoto, Detection of elliptic and linear edges by searching two parameter spaces, Proc. 5th Int. Joint Conf. Artificial Intelligence, Vol. 2, pp. 700–705. Cambridge, MA, August (1977).
20. J. Sklansky, On the Hough technique for curve detection, *IEEE Trans. Comput.*, **C27**, 923–926 (1978).
21. K. S. Fu, *Sequential Methods in Pattern Recognition and Machine Learning*. Academic Press, New York (1968).
22. R. Bolles, Verification vision with a programmable assembly system, Stanford AI Memo, AIM-275, December (1975).

APPENDIX. ANALYTIC HOUGH FOR PAIRS OF EDGE POINTS

To develop an explicit version of the Hough algorithm for ellipses using pairs of edge points, we consider the string-tied-at-two-ends parameterization of an ellipse:

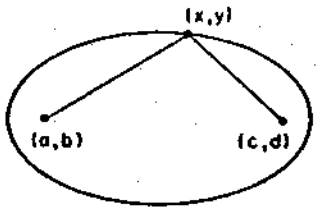


Fig. A1. String-tied-at-both-ends parameterization of an ellipse.

$$(x-a)^2 + (y-b)^2 + (x-c)^2 + (y-d)^2 = l^2$$

where (a, b) and (c, d) are the ends and l is the length of the string, as shown in Fig. A1. Now if we have two edge points (x_1, y_1) and (x_2, y_2) with gradients φ_1 and φ_2, the following equations result:

$$(x_1-a)^2 + (y_1-b)^2 + (x_1-c)^2 + (y_1-d)^2 = l^2 \quad \text{(A1)}$$

$$(x_1-a) + (y_1-b)\varphi_1 + (x_1-c) + (y_1-d)\varphi_1 = 0 \quad \text{(A2)}$$

$$(x_2-a)^2 + (y_2-b)^2 + (x_2-c)^2 + (y_2-d)^2 = l^2 \quad \text{(A3)}$$

$$(x_2-a) + (y_2-b)\varphi_2 + (x_2-c) + (y_2-d)\varphi_2 = 0 \quad \text{(A4)}$$

where in terms of the gradient direction

$$\varphi = \tan\left(\phi - \frac{\pi}{2}\right) = \frac{dy}{dx}.$$

From (A2):

$$a_1 = \varphi_1[(y_1-b) + (y_1-d)] + 2x_1 - c.$$

Substituting in (4):

$$x_2 - \varphi_1[(y_1-b) + (y_1-d)] - 2x_1 + (y_2-b)\varphi_2 + (y_2-d)\varphi_2 + x_2 = 0.$$

Rearranging terms:

$$2x_2 - 2x_1 - 2\varphi_1 y_2 + (\varphi_1-\varphi_2)b + (\varphi_1-\varphi_2)d = 0.$$

Now where:

$$S = \varphi_1 - \varphi_2$$

and

$$K = 2(x_2-x_1-\varphi_1 y_1 + \varphi_2 y_2)$$

and

$$t = \frac{K}{S}$$

we have

$$b = t - d. \quad \text{(A5)}$$

Now we substitute for b in (2)

$$x_1 - a = -(x_1-c) - \varphi_1(2y_1-t)$$

so that we have

$$c = \eta - a \quad \text{(A6)}$$

where $\eta = \varphi_1(2y-t)$

$$(x_1-a)^2 + [y_1-(t-d)]^2 + [x_1-(\eta-a)]^2 + (y_1-d)^2 = l^2 \quad \text{(A7)}$$

$$(x_2-a)^2 + [y_2-(t-d)]^2 + [x_2-(\eta-a)]^2 + (y-d)^2 = l^2 \quad \text{(A8)}$$

Thus our strategy for using two edge points is as follows:

Step 1: choose a.
Step 2: solve equations (5) and (6), a quadratic in d, for d.
Step 3: solve equation (2) for b and equation 3 for c.
Step 4: solve equation (1) for l.

Thus the vector $\mathbf{a} = (a, b, c, d, l)$ has been determined for a pair of edge pixels and can be used to increment the accumulator array.

About the Author—DANA H. BALLARD was born in Holyoke, MA, on October 15, 1946. He received the B.Sc. degree in aeronautics and astronautics from the Massachusetts Institute of Technology, Cambridge, in 1967. He received the M.S.E. degree in information and control engineering from the University of Michigan, Ann Arbor, in 1970 and the Ph.D. degree in information engineering from the University of California, Irvine, in 1974.

From 1968 to 1971 he was a Systems Analyst for Autonetics, Anaheim, Calif. Since 1971 he has been active in computer vision research. During the academic year 1974–1975, he was a Visiting Professor at the Laboratorio Biomediche Technologie, Rome. Italy. He is presently an Assistant Professor of Computer Science and Radiology at the University of Rochester, Rochester, N.Y. His current interests are in artificial intelligence and computer vision, particularly control strategies and geometric models, and their applications to biomedical image processing.

The Topographic Primal Sketch

Robert M. Haralick
Layne T. Watson
Thomas J. Laffey

Departments of Computer Science and
Electrical Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

Reprinted from *International Journal of Robotics Research*, Volume 2, Number 1, 1983, pages 50-72, "The Topographic Primal Sketch" by R.M. Haralick, T.J. Laffey, and L.T. Watson by permission of The MIT Press, Cambridge, Massachusetts.

Abstract

A complete mathematical treatment is given for describing the topographic primal sketch of the underlying gray tone intensity surface of a digital image. Each picture element is independently classified and assigned a unique descriptive label, invariant under monotonically increasing gray tone transformations from the set (peak, pit, ridge, ravine, saddle, flat, and hillside), with hillside having subcategories (inflection point, slope, convex hill, concave hill, and saddle hill). The topographic classification is based on the first and second directional derivatives of the estimated image-intensity surface. A local, facet model, two-dimensional, cubic polynomial fit is done to estimate the image-intensity surface. Zero-crossings of the first directional derivative are identified as locations of interest in the image.

1. Introduction

Representing the fundamental structure of a digital image in a rich and robust way is a primary problem encountered in any general robotics computer-vision system that has to "understand" an image. The richness is needed so that shading, highlighting, and shadow information, which are usually present in real manufacturing assembly line situations, are encoded. Richness permits unambiguous object matching to be accomplished. Robustness is needed so that the representation is invariant with respect to monotonically increasing gray tone transformations.

This research has been supported by National Science Foundation grant MCS-8102872.

The International Journal of Robotics Research
Vol. 2, No. 1, Spring 1983
0278-3649/83/010050-23 $05.00/0

Current representations involving edges or the primal sketch as described by Marr (1976; 1980) are impoverished in the sense that they are insufficient for unambiguous matching. They also do not have the required invariance. Basic research is needed to (1) define an appropriate representation, (2) develop a theory that establishes its relationship to properties that three-dimensional objects manifest on the image, and (3) prove its utility in practice. Until this is done, computer-vision research must inevitably be more ad hoc sophistication than science.

The basis of the topographic primal sketch consists of the classification and grouping of the underlying image-intensity surface patches according to the categories defined by monotonic, gray tone, invariant functions of directional derivatives. Examples of such categories are peak, pit, ridge, ravine, saddle, flat, and hillside. From this initial classification, we can group categories to obtain a rich, hierarchical, and structurally complete representation of the fundamental image structure. We call this representation the *topographic primal sketch*.

Why do we believe that this topographic primal sketch can be the basis for computer vision? We believe it because the light-intensity variations on an image are caused by an object's surface orientation, its reflectance, and characteristics of its lighting source. If any of the three-dimensional intrinsic surface characteristics are to be detected, they will be detected owing to the nature of light-intensity variations. Thus, the first step is to discover a robust representation that can encode the nature of these light-intensity variations, a representation that does not change with strength of lighting or with gain settings on the sensing camera. The topographic classification does just that. The basic research issue is to define a set of categories sufficiently complete to form groupings and structures that have strong

relationships to the reflectances, surface orientations, and surface positions of the three-dimensional objects viewed in the image.

1.1. The Invariance Requirement

A digital image can be obtained with a variety of sensing-camera gain settings. It can be visually enhanced by an appropriate adjustment of the camera's dynamic range. The gain setting or the enhancing, point operator changes the image by some monotonically increasing function that is not necessarily linear. For example, nonlinear, enhancing, point operators of this type include histogram normalization and equal probability quantization.

In visual perception, exactly the same visual interpretation and understanding of a pictured scene occurs whether the camera's gain setting is low or high and whether the image is enhanced or unenhanced. The only difference is that the enhanced image has more contrast, is nicer to look at, and is understood more quickly by the human visual system.

This fact is important because it suggests that many of the current-low-level computer-vision techniques, which are based on edges, cannot ever hope to have the robustness associated with human visual perception. They cannot have the robustness, because they are inherently incapable of invariance under monotonic transformations. For example, edges based on zero-crossings of second derivatives will change in position as the monotonic gray tone transformation changes because convexity of a gray tone intensity surface is not preserved under such transformations. However, the topographic categories peak, pit, ridge, valley, saddle, flat, and hillside do have the required invariance.

1.2. Background

Marr (1976) argues that the first level of visual processing is the computation of a rich description of gray level changes present in an image, and that all subsequent computations are done in terms of this description, which he calls the *primal sketch*. Gray level changes are usually associated with edges, and Marr's primal sketch has, for each area of gray level change, a description that includes type, position, orientation, and fuzziness of edge. Marr (1980) illustrates that from this information it is sometimes possible to reconstruct the image to a reasonable degree. Unfortunately, as mentioned earlier, edge is not invariant with respect to monotonic image transformations; besides, it is not a rich enough structure. For example, difficulty has been experienced in using edges to accomplish unambiguous stereo matching.

The topographic primal sketch we are discussing as a basis for a representation has the required richness and invariance properties and is very much in the spirit of Marr's primal sketch and the thinking behind Ehrich's relational trees (Ehrich and Foith 1978). Instead of concentrating on gray level changes as edges as Marr does, or on one-dimensional extrema as Ehrich and Foith, we concentrate on all types of two-dimensional gray level variations. We consider each area on an image to be a spatial distribution of gray levels that constitutes a surface or facet of gray tone intensities having a specific surface shape. It is likely that, if we could describe the shape of the gray tone intensity surface for each pixel, then by assembling all the shape fragments we could reconstruct, in a relative way, the entire surface of the image's gray tone intensity values. The shapes that we already know about that have the invariance property are peak, pit, ridge, ravine, saddle, flat, and hillside, with hillside having noninvariant subcategories of slope, inflection, saddle hillside, convex hillside, and concave hillside.

Knowing that a pixel's surface has the shape of a peak does not tell us precisely where in the pixel the peak occurs; nor does it tell us the height of the peak or the magnitude of the slope around the peak. The topographic labeling, however, does satisfy Marr's (1976) primal sketch requirement in that it contains a symbolic description of the gray tone intensity changes. Furthermore, upon computing and binding to each topographic label numerical descriptors such as gradient magnitude and direction, directions of the extrema of the second directional derivative along with their values, a reasonable absolute description of each surface shape can be obtained.

1.3. Facet Model

The *facet model* states that all processing of digital image data has its final authoritative interpretation relative to what the processing does to the underlying gray tone intensity surface. The digital image's pixel values are noisy sampled observations of the underlying surface. Thus, in order to do any processing, we at least have to estimate at each pixel position what this underlying surface is. This requires a model that describes what the general form of the surface would be in the neighborhood of any pixel if there were no noise. To estimate the surface from the neighborhood around a pixel then amounts to estimating the free parameters of the general form. It is important to note that if a different general form is assumed, then a different estimate of the surface is produced. Thus the assumption of a particular general form is necessary and has consequences.

The general form we use is a bivariate cubic. We assume that the neighborhood around each pixel is suitably fit by a bivariate cubic (Haralick 1981; 1982). Having estimated this surface around each pixel, the first and second directional derivatives are easily computed by analytic means. The topographic classification of the surface facet is based totally on the first and second directional derivatives. We classify each surface point as peak, pit, ridge, ravine, saddle, flat, or hillside, with hillside being broken down further into the subcategories inflection point, convex hill, concave hill, saddle hill, and slope. Our set of topographic labels is complete in the sense that every combination of values of the first and second directional derivative is uniquely assigned to one of the classes.

1.4. Previous Work

Detection of topographic structures in a digital image is not a new idea. There has been a wide variety of techniques described to detect pits, peaks, ridges, ravines, and the like.

Peuker and Johnston (1972) characterize the surface shape by the sequence of positive and negative differences as successive surrounding points are compared to the central point. Peuker and Douglas (1975) describe several variations of this method for detecting one of the shapes from the set (pit, peak, pass, ridge, ravine, break, slope, flat). They start with the most frequent feature (slope) and proceed to the less frequent, thus making it an order-dependent algorithm.

Johnston and Rosenfeld (1975) attempt to find peaks by finding all points P such that no points in an n-by-n neighborhood surrounding P have greater elevation than P. Pits are found in an analogous manner. To find ridges, they identify points that are either east-west or north-south elevation maxima. This is done using a "smoothed" array in which each point is given the highest elevation in a 2×2 square containing it. East-west and north-south maxima are also found on this array. Ravines are found in a similar manner.

Paton (1975) uses a six-term quadratic expansion in Legendre polynomials fitted to a small disk around each pixel. The most significant coefficients of the second-order polynomial yield a descriptive label chosen from the set (constant, ridge, valley, peak, bowl, saddle, ambiguous). He uses the continuous least-squares-fit formulation in setting up the surface-fit equations as opposed to the discrete least-squares fit used in the facet model. The continuous fit is a more expensive computation than the discrete fit and results in a steplike approximation.

Grender's (1976) algorithm compares the gray level elevation of a central point with surrounding elevations at a given distance around the perimeter of a circular window; the radius of the window may be increased in successive passes through the image. His topographic labeling set consists of slope, ridge, valley, knob, sink, saddle.

Toriwaki and Fukumara (1978) take a totally different approach from all the others. They use two local features of gray level pictures, connectivity number, and coefficient of curvature for classification of the pixel into peak, pit, ridge, ravine, hillside, pass. They then describe how to extract structural information from the image once the labelings have been made. This structural information consists of ridge-lines, ravine-lines, and the like.

Hsu, Mundy, and Beaudet (1978) use a quadratic surface approximation at every point on the image

surface. The principal axes of the quadratic approximation are used as directions in which to segment the image. Lines emanating from the center pixel in these directions thus provide natural boundaries of patches approximating the surface. The authors then selectively generate the principal axes from some critical points distributed over an image and interconnect them into a network to get an approximation of the image data. In this network, which they call the *web representation*, the axes divide the image into regions and show important features such as edges and peaks. They are then able to extract a set of primitive features from the nodes of the network by mask matching. Global features, such as ridge-lines, are obtained by state transition rules.

Lee and Fu (1981) define a set of 3×3 templates that they convolve over the image to give each class except plain a *figure of merit*. Their set of labels includes none, plain, slope, ridge, valley, foot, shoulder. Thresholds are used to determine into which class the pixel will fall. In their scheme, a pixel may satisfy the definition of zero, one, or more than one class. Ambiguity is resolved by choosing the class with the highest figure of merit.

1.5. A Mathematical Approach

From the previous discussion, one can see that a wide variety of methods and labels has been proposed to describe the topographic structure in a digital image. Some of the methods require multiple passes through the image, while others may give ambiguous labels to a pixel. Many of the methods are heuristic in nature. The Hsu, Mundy, and Beudet (1978) approach is the most similar to the one discussed here.

Our classification approach is based on the estimation of the first- and second-order directional derivatives. Thus, we regard the digital-picture function as a sampling of the underlying function f, where some kind of random noise is added to the true function values. To estimate the first and second partials, we must assume some kind of parametric form for the underlying function f. The classifier must use the sampled brightness values of the digital-picture function to estimate the parameters and then make decisions regarding the locations of relative extrema of partial derivatives based on the estimated values of the parameters.

In Section 2, we will discuss the mathematical properties of the topographic structures in terms of the directional derivatives in the continuous surface domain. Because a digital image is a sampled surface and each pixel has an area associated with it, characteristic topographic structures may occur anywhere within a pixel's area. Thus, the implementation of the mathematical topographic definitions is not entirely trivial.

In Section 3 we will discuss the implementation of the classification scheme on a digital image. To identify categories that are local one-dimensional extrema, such as peak, pit, ridge, ravine, and saddle, we search inside the pixel's area for a zero-crossing of the first directional derivative. The directions in which we seek the zero-crossing are along the lines of extreme curvature.

In Section 4, we will discuss the local cubic estimation scheme. In Section 5, we will summarize the algorithm for topographic classification using the local facet model. In Section 6, we will show the results of the classifier on several test images.

2. The Mathematical Classification of Topographic Structures

In this section, we formulate our notion of topographic structures on continuous surfaces and show their invariance under monotonically increasing gray tone transformations. In order to understand the mathematical properties used to define our topographic structures, one must understand the idea of the *directional derivative* discussed in most advanced calculus books. For completeness, we first give the definition of the directional derivative, then the defitions of the topographic labels. Finally, we show the invariance under monotonically increasing gray tone transformations.

2.1. The Directional Derivative

In two dimensions, the rate of change of a function f depends on direction. We denote the directional

derivative of f at the point (r, c) in the direction β by $f'_\beta(r, c)$. It is defined as

$$f'_\beta(r, c) = \lim_{h \to 0} \frac{f(r + h^* \sin \beta, c + h^* \cos \beta) - f(r, c)}{h}.$$

The direction angle β is the clockwise angle from the column axis. It follows directly from this definition that

$$f'_\beta(r, c) = \frac{\partial f}{\partial r}(r, c) * \sin \beta + \frac{\partial f}{\partial c}(r, c) * \cos \beta.$$

We denote the second derivative of f at the point (r, c) in the direction β by $f''_\beta(r, c)$, and it follows that

$$f''_\beta = \frac{\partial^2 f}{\partial r^2} * \sin^2 \beta + 2 * \frac{\partial^2 f}{\partial r\, \partial c} * \sin \beta * \cos \beta + \frac{\partial^2 f}{\partial c^2} * \cos^2 \beta.$$

The *gradient* of f is a vector whose magnitude,

$$\left(\left(\frac{\partial f}{\partial r}\right)^2 + \left(\frac{\partial f}{\partial c}\right)^2\right)^{1/2}$$

at a given point (r, c) is the maximum rate of change of f at that point, and whose direction,

$$\tan^{-1}\left(\frac{\dfrac{\partial f}{\partial r}}{\dfrac{\partial f}{\partial c}}\right)$$

is the direction in which the surface has the greatest rate of change.

2.2. The Mathematical Properties

We will use the following notation to describe the mathematical properties of our various topographic categories for continuous surfaces. Let

∇f = gradient vector of a function f;
$\|\nabla f\|$ = gradient magnitude;
$\omega^{(1)}$ = unit vector in direction in which second directional derivative has greatest magnitude;
$\omega^{(2)}$ = unit vector orthogonal to $\omega^{(1)}$;
λ_1 = value of second directional derivative in the direction of $\omega^{(1)}$;
λ_2 = value of second directional derivative in the direction of $\omega^{(2)}$;
$\nabla f \cdot \omega^{(1)}$ = value of first directional derivative in the direction of $\omega^{(1)}$; and
$\nabla f \cdot \omega^{(2)}$ = value of first directional derivative in the direction of $\omega^{(2)}$.

Without loss of generality, we assume $|\lambda_1| >= |\lambda_2|$.

Each type of topographic structure in our classification scheme is defined in terms of the above quantities. In order to calculate these values, the first- and second-order partials with respect to r and c need to be approximated. These five partials are as follows:

$$\partial f/\partial r,\ \partial f/\partial c,\ \partial^2 f/\partial r^2,\ \partial^2 f/\partial c^2,\ \partial^2 f/\partial r\, \partial c.$$

The gradient vector is simply $(\partial f/\partial r, \partial f/\partial c)$. The second directional derivatives may be calculated by forming the *Hessian* where the Hessian is a 2×2 matrix defined as

$$H = \begin{vmatrix} \partial^2 f/\partial r^2 & \partial^2 f/\partial r\, \partial c \\ \partial^2 f/\partial c\, \partial r & \partial^2 f/\partial c^2 \end{vmatrix}.$$

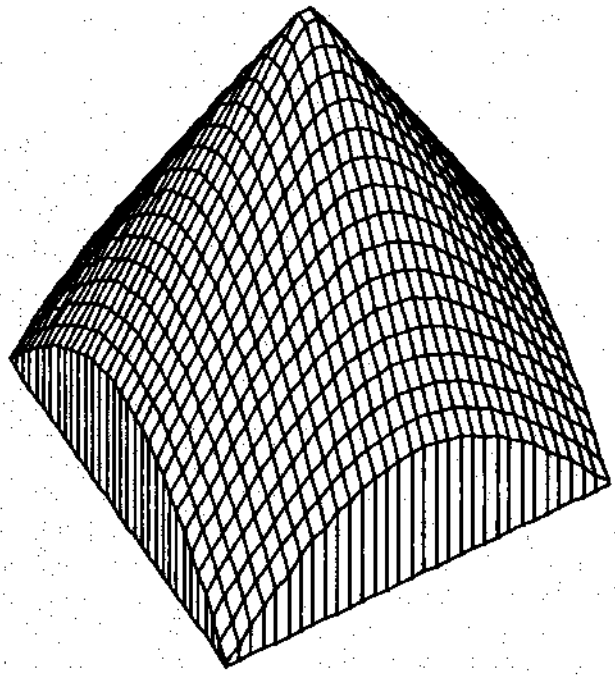
Hessian matrices are used extensively in nonlinear programming. Only three parameters are required to determine the Hessian matrix H, since the order of differentiation of the cross partials may be interchanged. That is,

$$\partial^2 f/\partial r\, \partial c = \partial^2 f/\partial c\, \partial r.$$

The eigenvalues of the Hessian are the values of the extrema of the second directional derivative, and their associated eigenvectors are the directions in which the second directional derivative is extremized. This can easily be seen by rewriting f''_β as the quadratic form

$$f''_\beta = (\sin \beta \cos \beta) * H * \begin{vmatrix} \sin \beta \\ \cos \beta \end{vmatrix}$$

Fig. 1. Right circular cone.



Thus

$$H\omega^{(1)} = \lambda_1\omega^{(1)}, \text{ and } H\omega^{(2)} = \lambda_2\omega^{(2)}.$$

Furthermore, the two directions represented by the eigenvectors are orthogonal to one another. Since H is a 2×2 symmetric matrix, calculation of the eigenvalues and eigenvectors can be done efficiently and accurately using the method of Rutishauser (1971). We may obtain the values of the first directional derivative by simply taking the dot product of the gradient with the appropriate eigenvector:

$$\nabla f \cdot \omega^{(1)}$$
$$\nabla f \cdot \omega^{(2)}.$$

There is a direct relationship between the eigenvalues λ_1 and λ_2 and curvature in the directions $\omega^{(1)}$ and $\omega^{(2)}$: When the first directional derivative $\nabla f \cdot \omega^{(1)} = 0$, then $\lambda_i/(1 + (\nabla f \cdot \nabla f))^{1/2}$ is the curvature in the direction $\omega^{(1)}$, $i = 1$ or 2.

Having the gradient magnitude and direction and the eigenvalues and eigenvectors of the Hessian, we can describe the topographic classification scheme.

2.2.1. *Peak*

A peak (knob) occurs where there is a local maxima in all directions. In other words, we are on a peak if, no matter what direction we look in, we see no point that is as high as the one we are on (Fig. 1). The curvature is downward in all directions. At a peak the gradient is zero, and the second directional derectional derivative is negative in all directions. To test whether the second directional derivative is negative in all directions, we just have to examine the value of the second directional derivative in the directions that make it smallest and largest. A point is therefore classified as a peak if it satisfies the following conditions:

$$\|\nabla f\| = 0, \lambda_1 < 0, \lambda_2 < 0.$$

2.2.2. *Pit*

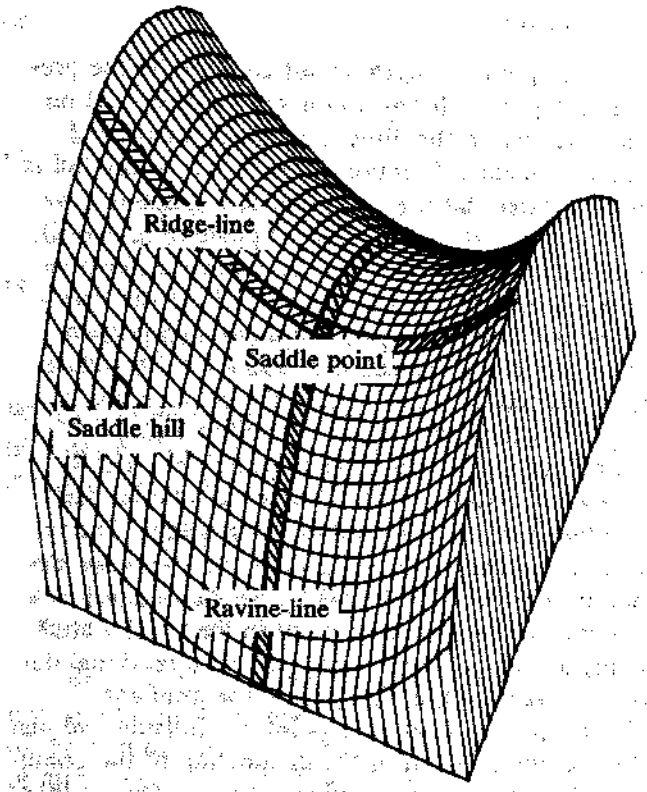
A pit (sink, bowl) is identical to a peak except that it is a local minima in all directions rather than a local maxima. At a pit the gradient is zero, and the second directional derivative is positive in all directions. A point is classified as a pit if it satisfies the following conditions:

$$\|\nabla f\| = 0, \lambda_1 > 0, \lambda_2 > 0.$$

2.2.3. *Ridge*

A ridge occurs on a ridge-line, a curve consisting of a series of ridge points. As we walk along the ridge-line, the points to the right and left of us are lower than the ones we are on. Furthermore, the ridge-line may be flat, slope upward, slope downward, curve upward, or curve downward. A ridge occurs where there is a local maximum in one direction, as illustrated in Fig. 2. Therefore, it must have negative second-directional derivative in the direction across the ridge and also a zero first-directional derivative in that same direction. The direction in which the local maximum occurs may correspond to either of the directions in which the curvature is "extremized," since the ridge itself may be curved. For nonflat ridges, this leads to the first two cases below for ridge characterization. If the ridge is flat,

Fig. 2. Saddle surface.



then the ridge-line is horizontal and the gradient is zero along it. This corresponds to the third case. The defining characteristic is that the second directional derivative in the direction of the ridge-line is zero, while the second directional derivative across the ridge-line is negative. A point is therefore classified as a ridge if it satisfies any one of the following three sets of conditions:

$$\|\nabla f\| \neq 0,\ \lambda_1 < 0,\ \nabla f \cdot \omega^{(1)} = 0$$

or

$$\|\nabla f\| \neq 0,\ \lambda_2 < 0,\ \nabla f \cdot \omega^{(2)} = 0$$

or

$$\|\nabla f\| = 0,\ \lambda_1 < 0,\ \lambda_2 = 0.$$

A geometric way of thinking about the definition for ridge is to realize that the condition $\nabla f \cdot \omega^{(1)} = 0$ means that the gradient direction (which is defined for nonzero gradients) is orthogonal to the direction $\omega^{(1)}$ of extremized curvature.

2.2.4. *Ravine*

A ravine (valley) is identical to a ridge except that it is a local minimum (rather than maximum) in one direction. As we walk along the ravine-line, the points to the right and left of us are higher than the one we are on (see Fig. 2). A point is classified as a ravine if it satisfies any one of the following three sets of conditions:

$$\|\nabla f\| \neq 0,\ \lambda_1 > 0,\ \nabla f \cdot \omega^{(1)} = 0$$

or

$$\|\nabla f\| \neq 0,\ \lambda_2 > 0,\ \nabla f \cdot \omega^{(2)} = 0$$

or

$$\|\nabla f\| = 0,\ \lambda_1 > 0,\ \lambda_2 = 0.$$

2.2.5. *Saddle*

A saddle occurs where there is a local maximum in one direction and a local minimum in a perpendicular direction, as illustrated in Fig. 2. A saddle must therefore have positive curvature in one direction and negative curvature in a perpendicular direction. At a saddle, the gradient magnitude must be zero and the extrema of the second directional derivative must have opposite signs. A point is classified as a saddle if it satisfies the following conditions:

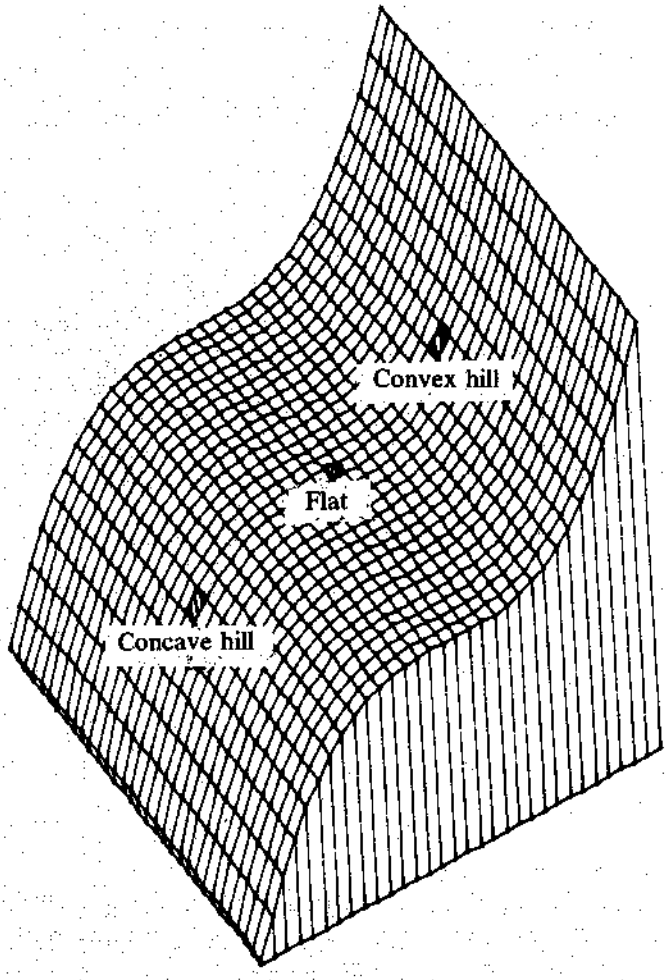
$$\|\nabla f\| = 0,\ \lambda_1 * \lambda_2 < 0.$$

2.2.6. *Flat*

A flat (plain) is a simple, horizontal surface, as illustrated in Fig. 3. It, therefore, must have zero gradient and no curvature. A point is classified as a flat if it satisfies the following conditions:

$$\|\nabla f\| = 0,\ \lambda_1 = 0,\ \lambda_2 = 0.$$

Fig. 3. Hillside.



Given that the above conditions are true, a flat may be further classified as a *foot* or *shoulder*. A foot occurs at that point where the flat just begins to turn up into a hill. At this point, the third directional derivative in the direction toward the hill will be nonzero, and the surface increases in this direction. The shoulder is an analogous case and occurs where the flat is ending and turning down into a hill. At this point, the maximum magnitude of the third directional derivative is nonzero, and the surface decreases in the direction toward the hill. If the third directional derivative is zero in all directions, then we are in a flat, not near a hill. Thus a flat may be further qualified as being a foot or shoulder, or not qualified at all.

2.2.7. *Hillside*

A hillside point is anything not covered by the previous categories. It has a nonzero gradient and no strict extrema in the directions of maximum and minimum second directional derivative. If the hill is simply a tilted flat (i.e., has constant gradient), we call it a *slope*. If its curvature is positive (upward), we call it a *convex hill*. If its curvature is negative (downward), we call it a *concave hill*. If the curvature is up in one direction and down in a perpendicular direction, we call it a *saddle hill*. A saddle hill is illustrated in Fig. 2, and the slope, convex hill, and concave hill are illustrated in Fig. 3.

A point on a hillside is an *inflection point* if it has a zero-crossing of the second directional derivative taken in the direction of the gradient. The inflection-point class is the same as the *step edge* defined by Haralick (1982), who classifies a pixel as a step edge if there is some point in the pixel's area having a zero-crossing of the second directional derivative taken in the direction of the gradient.

To determine whether a point is a hillside, we just take the complement of the disjunction of the conditions given for all the previous classes. Thus if there is no curvature, then the gradient must be nonzero. If there is curvature, then the point must not be a relative extremum. Therefore, a point is classified as a hillside if all three sets of the following conditions are true ('→' represents the operation of logical implication):

$$\lambda_1 = \lambda_2 = 0 \rightarrow \|\nabla f\| \neq 0,$$

and

$$\lambda_1 \neq 0 \rightarrow \nabla f \cdot \omega^{(1)} \neq 0,$$

and

$$\lambda_2 \neq 0 \rightarrow \nabla f \cdot \omega^{(2)} \neq 0.$$

Rewritten as a disjunction of clauses rather than a conjunction of clauses, a point is classified as a hillside if any one of the following four sets of conditions are true:

$$\nabla f \cdot \omega^{(1)} \neq 0,\ \nabla f \cdot \omega^{(2)} \neq 0$$

or

$$\nabla f \cdot \omega^{(1)} \neq 0,\ \lambda_2 = 0$$

or

$$\nabla f \cdot \omega^{(2)} \neq 0,\ \lambda_1 = 0$$

or

$$\|\nabla f\| \neq 0,\ \lambda_1 = 0,\ \lambda_2 = 0.$$

We can differentiate between different classes of hillsides by the values of the second directional derivative. The distinction can be made as follows:

Slope	if $\lambda_1 = \lambda_2 = 0$
Convex	if $\lambda_1 >= \lambda_2 >= 0,\ \lambda_1 \neq 0$
Concave	if $\lambda_1 <= \lambda_2 <= 0,\ \lambda_1 \neq 0$
Saddle hill	if $\lambda_1 * \lambda_2 < 0$

A slope, convex, concave, or saddle hill is classified as an inflection point if there is a zero-crossing of the second directional derivative in the direction of maximum first directional derivative (i.e., the gradient).

2.2.8. *Summary of the Topographic Categories*

A summary of the mathematical properties of our topographic structures on continuous surfaces can be found in Table 1. The table exhaustively defines the topographic classes by their gradient magnitude, second directional derivative extrema values, and the first directional derivatives taken in the directions which extremize second directional derivatives. Each entry in the table is either 0, +, −, or *. The 0 means not significantly different from zero; + means significantly different from zero on the positive side; − means significantly different from zero on the negative side, and * means it does not matter. The label "Cannot occur" means that it is impossible for the gradient to be nonzero and the first directional derivative to be zero in two orthogonal directions.

From the table, one can see that our classification scheme is complete. All possible combinations of first and second directional derivatives have a corresponding entry in the table. Each topographic category has a set of mathematical properties that uniquely determines it.

(Note: Special attention is required for the degenerate case $\lambda_1 = \lambda_2 \neq 0$, which implies that $\omega^{(1)}$ and $\omega^{(2)}$ can be *any* two orthogonal directions. In this case, there *always* exists an extreme direction ω which is orthogonal to ∇f, and thus the first directional derivative $\nabla f \cdot \omega$ is *always* zero in an extreme direction. To avoid spurious zero directional derivatives, we choose $\omega^{(1)}$ and $\omega^{(2)}$ such that $\nabla f \cdot \omega^{(1)} \neq 0$ and $\nabla f \cdot \omega^{(2)} \neq 0$, unless the gradient is zero.)

Table 1. Mathematical Properties of Topographic Structures

$\|\nabla f\|$	λ_1	λ_2	$\nabla f \cdot \omega^{(1)}$	$\nabla f \cdot \omega^{(2)}$	Label
0	−	−	0	0	Peak
0	−	0	0	0	Ridge
0	−	+	0	0	Saddle
0	0	0	0	0	Flat
0	+	−	0	0	Saddle
0	+	0	0	0	Ravine
0	+	+	0	0	Pit
+	−	−	−, +	−, +	Hillside
+	−	*	0	*	Ridge
+	*	−	*	0	Ridge
+	−	0	−, +	*	Hillside
+	−	+	−, +	−, +	Hillside
+	0	0	*	*	Hillside
+	+	−	−, +	−, +	Hillside
+	+	0	−, +	*	Hillside
+	+	*	0	*	Ravine
+	*	+	*	0	Ravine
+	+	+	−, +	−, +	Hillside
+	*	*	0	0	Cannot occur

2.3. The Invariance of the Topographic Categories

In this section, we show that the topographic labels (peak, pit, ridge, ravine, saddle, flat, and hillside), the gradient direction, and directions of second directional derivative extrema for peak, pit, ridge, ravine, and saddle are all invariant under monotoni-

cally increasing gray tone transformations. We take *monotonically increasing* to mean positive derivative everywhere.

Let the original underlying gray tone surface be $f(r, c)$. Let w be a monotonically increasing gray tone transformation, and let $g(r, c)$ denote the transformed image: $g(r, c) = w(f(r, c))$. It is directly derivable that

$$g'_\beta(r, c) = w'(f(r, c)) * f'_\beta(r, c),$$

from which we obtain that

$$g''_\beta(r, c) = w'(f(r, c)) * f''_\beta(r, c) + w''(f_\beta(r, c)) * f'_\beta(r, c)^2.$$

Let us fix a position (r, c). Since w is a monotonically increasing function, w' is positive. In particular, w' is not zero. Hence the direction β which maximizes g'_β also maximizes f'_β, thereby showing that the gradient directions are the same. The categories peak, pit, ridge, ravine, saddle, and flat all have in common the essential property that the first directional derivative is zero when taken in a direction that extremizes the second directional derivative. To see the invariance, let β be an extremizing direction of f''_β. Then for points (r, c) having a label (peak, pit, ridge, ravine, saddle, or flat), $f'_\beta(r, c) = 0$, and $\partial f''_\beta(r, c)/\partial\beta = 0$. Notice that

$$\frac{\partial g''_\beta}{\partial\beta} = w' * \frac{\partial}{\partial\beta} f''_\beta + 2 * w' * f'_\beta \frac{\partial f'_\beta}{\partial\beta} + (f'_\beta)^2 \frac{\partial w''}{\partial\beta}.$$

Hence for these points, $g'_\beta(r, c) = 0$, and

$$\partial g''_\beta(r, c)/\partial\beta = 0,$$

thereby showing that at these points the directions that extremize f''_β are precisely the directions that extremize g''_β, and that g''_β will always have the same sign as f''_β. A similar argument shows that if β extremizes g''_β and satisfies $g'_\beta = 0$, then β must also extremize f''_β and satisfy $f'_\beta = 0$. Therefore, any points in the original image with the labels peak, pit, ridge, saddle, or flat retain the same label in the transformed image and, conversely, any points in the transformed image will have the same label in the original image.

Any pixel with a label not in the set (peak, pit, ridge, ravine, saddle, and flat) must have a hillside label. Thus, a point labeled hillside must be transformed to a hillside-labeled point. However, the subcategories (inflection point, slope, convex hill, concave hill, and saddle hill) may change under the gray tone transformation.

2.4. Ridge and Ravine Continua

Although the definitions given for ridge and ravine are intuitively pleasing, they may lead to the unexpected consequence of having entire areas of a surface classified as all ridge or all ravine. To see how this can occur, observe that the eigenvalue $\lambda = \lambda(r, c)$ satisfies

$$\lambda(r, c) = \frac{1}{2}\left|\frac{\partial^2 f}{\partial r^2}(r, c) + \frac{\partial^2 f}{\partial c^2}(r, c)\right| \pm \left|\left|\frac{\partial^2 f}{\partial r\,\partial c}(r, c)\right|^2 + \left|\frac{1}{2}\left|\frac{\partial^2 f}{\partial r^2}(r, c) - \frac{\partial^2 f}{\partial c^2}(r, c)\right|\right|^2\right|^{1/2}.$$

For there to be a ridge or ravine at a point (r, c), the corresponding eigenvector $\omega(r, c)$ must be perpendicular to the gradient direction. Therefore, $\nabla f \cdot \omega = 0$. If this equation holds for a point (r, c) and not all points in a small neighborhood about (r, c), there is a ridge or ravine in the commonly understood sense. However, if this equation holds for all points in a neighborhood about (r, c), then we have a ridge or ravine continuum by the criteria of Sections 2.2.3 and 2.2.4.

Unfortunately, there are "nonpathologic" surfaces having ridge or ravine continuums. Simple, radially symmetric examples include the inverted right circular cone defined by

$$f(r, c) = (r^2 + c^2)^{1/2},$$

the hemisphere defined by

$$f(r, c) = (k^2 - r^2 - c^2)^{1/2},$$

or, in fact, any function of the form $h(r^2 + c^2)$. In

the case of the cone, the gradient is proportional to (r, c), and the unnormalized eigenvectors corresponding to eigenvalues

$$\lambda(r, c) = (r^2 + c^2)^{-1/2} \text{ and } 0$$

are $(-c, r)$ and (r, c) respectively. The eigenvector corresponding to the nonzero eigenvalue is orthogonal to the gradient direction. The entire surface of the inverted cone, except for the apex, is a ravine. Other, nonradially symmetric examples exist as well.

The identification of points that are really ridge or ravine continuums can be made as a postprocessing step. Points that are labeled as ridge or ravine and that have neighboring points in a direction orthogonal to the gradient that are also labeled ridge or ravine are ridge or ravine continuums. These continuums can be reclassified as hillsides.

3. The Topographic Classification Algorithm

The definitions of Section 2 cannot be used directly since there is a problem of where in a pixel's area to apply the classification. If the classification were only applied to the point at the center of each pixel, then a pixel having a peak near one of its corners, for example, would get classified as a concave hill rather than as a peak. The problem is that the topographic classification we are interested in must be a sampling of the actual topographic surface classes. Most likely, the interesting categories of peak, pit, ridge, ravine, and saddle will never occur precisely at a pixel's center, and if they do occur in a pixel's area, then the pixel must carry that label rather than the class label of the pixel's center point. Thus one problem we must solve is to determine the dominant label for a pixel given the topographic class label of every point in the pixel. The next problem we must solve is to determine, in effect, the set of all topographic classes occurring within a pixel's area without having to do the impossible brute-force computation.

For the purpose of solving these problems, we divide the set of topographic labels into two subsets: (1) those that indicate that a strict, local, one-dimensional extremum has occurred (peak, pit, ridge, ravine, and saddle) and (2) those that do not indicate that a strict, local, one-dimensional extremum has occurred (flat and hillside). By *one-dimensional*, we mean along a line (in a particular direction). A strict, local, one-dimensional extremum can be located by finding those points within a pixel's area where a zero-crossing of the first directional derivative occurs.

So that we do not search the pixel's entire area for the zero-crossing, we only search in the directions of extreme second directional derivative, $\omega^{(1)}$ and $\omega^{(2)}$. Since these directions are well aligned with curvature properties, the chance of overlooking an important topographic structure is minimized, and, more importantly, the computational cost is small.

When $\lambda_1 = \lambda_2 \neq 0$, the directions $\omega^{(1)}$ and $\omega^{(2)}$ are not uniquely defined. We handle this case by searching for a zero-crossing in the direction given by $\mathbf{H}^{-1} * \nabla f$. This is the *Newton direction*, and it points directly toward the extremum of a quadratic surface.

For inflection-point location (first derivative extremum), we search along the gradient direction for a zero-crossing of second directional derivative. For one-dimensional extrema, there are four cases to consider: (1) no zero-crossing, (2) one zero-crossing, (3) two zero-crossings, and (4) more than two zero-crossings. The next four sections discuss these cases.

3.1. Case One: No Zero-Crossing

If no zero-crossing is found along either of the two extreme directions within the pixel's area, then the pixel cannot be a local extremum and therefore must be assigned a label from the set (flat or hillside). If the gradient is zero, we have a flat. If it is nonzero, we have a hillside. If the pixel is a hillside, we classify it further into inflection point, slope, convex hill, concave hill, or saddle hill. If there is a zero-crossing of the second directional derivative in the direction of the gradient within the pixel's area, the pixel is classified as an inflection point. If no such zero-crossing occurs, the label assigned to the pixel is based on the gradient magnitude and Hessian eigenvalues calculated at the center of the pixel, local coordinates (0, 0), as in Table 2.

Table 2. Pixel Label Calculation for Case One: No Zero-Crossing

$\|\nabla f\|$	λ_1	λ_2	Label
0	0	0	Flat
+	−	−	Concave hill
+	−	0	Concave hill
+	−	+	Saddle hill
+	0	0	Slope
+	+	−	Saddle hill
+	+	0	Convex hill
+	+	+	Convex hill

3.2. Case Two: One Zero-Crossing

If a zero-crossing of the first directional derivative is found within the pixel's area, then the pixel is a strict, local, one-dimensional extremum and must be assigned a label from the set (peak, pit, ridge, ravine, or saddle). At the location of the zero-crossing, the Hessian and gradient are recomputed, and if the gradient magnitude at the zero-crossing is zero, Table 3 is used.

If the gradient magnitude is nonzero, then the choice is either ridge or ravine. If the second directional derivative in the direction of the zero-crossing is negative, we have a ridge. If it is positive, we have a ravine. If it is zero, we compare the function value at the center of the pixel, $f(0, 0)$, with the function value at the zero-crossing, $f(r, c)$. If $f(r, c)$ is greater than $f(0, 0)$, we call it a ridge, otherwise we call it a ravine.

3.3. Case Three: Two Zero-Crossings

If we have two zero-crossings of the first directional derivative, one in each direction of extreme curvature, then the Hessian and gradient must be recomputed at each zero-crossing. Using the procedure described in Section 3.2, we assign a label to each zero-crossing. We call these labels LABEL1 and LABEL2. The final classification given the pixel is based on these two labels and is given in Table 4.

If both labels are identical, the pixel is given that label. In the case of both labels being ridge, the pixel

Table 3. Pixel Label Calculation for Case Two: One Zero-Crossing

$\|\nabla f\|$	λ_1	λ_2	Label
0	−	−	Peak
0	−	0	Ridge
0	−	+	Saddle
0	+	−	Saddle
0	+	0	Ravine
0	+	+	Pit

Table 4. Final Pixel Classification, Case Three: Two Zero-Crossings

LABEL1	LABEL2	Resulting Label
Peak	Peak	Peak
Peak	Ridge	Peak
Pit	Pit	Pit
Pit	Ravine	Pit
Saddle	Saddle	Saddle
Ridge	Ridge	Ridge
Ridge	Ravine	Saddle
Ridge	Saddle	Saddle
Ravine	Ravine	Ravine
Ravine	Saddle	Saddle

may actually be a peak, but experiments have shown that this case is rare. An anlogous argument can be made for both labels being ravine. If one label is ridge and the other ravine, this indicates we are at or very close to a saddle point, and thus the pixel is classified as a saddle. If one label is peak and the other ridge, we choose the category giving us the "most information," which in this case is peak. The peak is a local maximum in all directions, while the ridge is a local maximum in only one direction. Thus, peak conveys more information about the image surface. An analogous argument can be made if the labels are pit and ravine. Similarly, a saddle gives us more information than a ridge or valley. Thus, a pixel is assigned saddle if its zero-crossings have been labeled ridge and saddle or ravine and saddle.

It is apparent from Table 4 that not all possible label combinations are accounted for. Some combi-

nations, such as peak and pit, are omitted because of the assumption that the underlying surface is smooth and sampled frequently enough that a peak and pit will not both occur within the same pixel's area. If such a case occurs, our convention is to choose arbitrarily one of LABEL1 or LABEL2 as the resulting label for the pixel.

3.4. Case Four: More than Two Zero-Crossings

If more than two zero-crossings occur within a pixel's area, then in at least one of the extrema directions there are two zero-crossings. If this happens, we choose the zero-crossing closest to the pixel's center and ignore the other. If we ignore the further zero-crossings, then this case is identical to case 3. This situation has yet to occur in our experiments.

4. Surface Estimation

In this section we discuss the estimation of the parameters required by the topographic classification scheme of Section 2 using the local cubic facet model (Haralick 1981). It is important to note that the classification scheme of Section 2 and the algorithm of Section 3 are independent of the method used to estimate the first- and second-order partials of the underlying digital image-intensity surface at each sampled point. Although we are currently using the cubic model and discuss it here, we expect that a spline-based estimation scheme or a discrete-cosines estimation scheme may, in fact, provide better estimates.

4.1. Local Cubic Facet Model

In order to estimate the required partial derivatives, we perform a least-squares fit with a two-dimensional surface, f, to a neighborhood of each pixel. It is required that the function f be continuous and have continuous first- and second-order partial derivatives with respect to r and c in a neighborhood around each pixel in the rc plane.

We choose f to be a cubic polynomial in r and c expressed as a combination of discrete orthogonal polynomials. The function f is the best discrete least-squares polynomial approximation to the image data in each pixel's neighborhood. More details can be found in Haralick's paper (1981), in which each coefficient of the cubic polynomial is evaluated as a linear combination of the pixels in the fitting neighborhood.

To express the procedure precisely and without reference to a particular set of polynomials tied to neighborhood size, we will canonically write the fitted bicubic surface for each fitting neighborhood as

$$\begin{aligned} f(r, c) = {} & k_1 + k_2 r + k_3 c \\ & + k_4 r^2 + k_5 rc + k_6 c^2 \\ & + k_7 r^3 + k_8 r^2 c + k_9 r c^2 + k_{10} c^3, \end{aligned}$$

where the center of the fitting neighborhood is taken as the origin. It quickly follows that the needed partials evaluated at local coordinates (r, c) are

$$\begin{aligned} \partial f/\partial r &= k_2 + 2k_4 r + k_5 c + 3k_7 r^2 + 2k_8 rc + k_9 c^2 \\ \partial f/\partial c &= k_3 + k_5 r + 2k_6 c + k_8 r^2 + 2k_9 rc + 3k_{10} c^2 \\ \partial^2 f/\partial r^2 &= 2k_4 + 6k_7 r + 2k_8 c \\ \partial^2 f/\partial c^2 &= 2k_6 + 2k_9 r + 6k_{10} c \\ \partial^2 f/\partial r\, \partial c &= k_5 + 2k_8 r + 2k_9 c. \end{aligned}$$

It is easy to see that if the above quantities are evaluated at the center of the pixel where local coordinates $(r, c) = (0, 0)$, only the constant terms will be of significance. If the partials need to be evaluated at an arbitrary point in a pixel's area, then a linear or quadratic polynomial value must be computed.

4.2. An Observation About Cubic Fits

A two-dimensional cubic polynomial includes an arbitrary quadratic polynomial, and thus features like pit, peak, and saddle can be replicated exactly. For other surface features like ridges or ravines, cubics are either exact or fairly decent approximations. It is

Fig. 4. Cubic fit of step edge that causes ravine and ridge to occur.

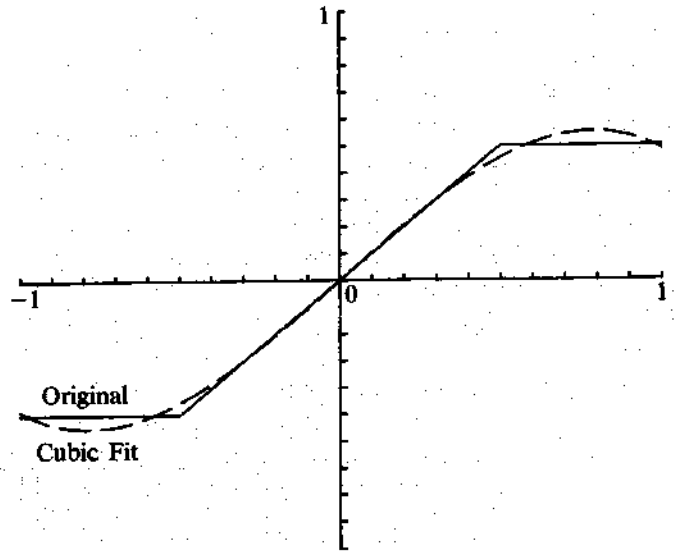
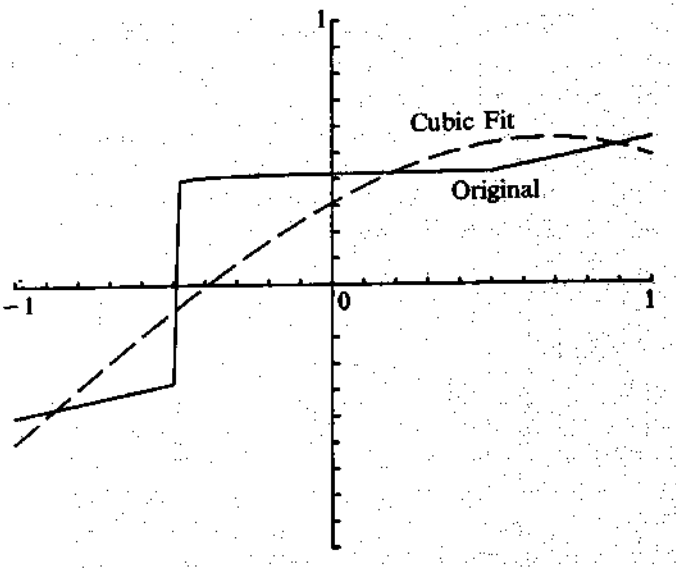


Fig. 5. Cubic fit (dashed line) of step edge (solid line) with inflection point outside window.



frequently possible to classify surface characteristics correctly even though the surface is not like any polynomial. For example, the center of the step edge shown in Fig. 4 can be accurately predicted by the inflection point of the cubic fit, which is quite good. However, there are smooth surfaces that (over the window being used) simply do not look like cubic polynomials, and feature classifications based on the best least-squares cubic polynomial approximation will be incorrect. For example, in Fig. 4, the cubic fit shows a prominent ravine at the foot of the slope, and the foot pixel would be (incorrectly) labeled a ravine pixel. Figure 5 shows a steplike edge whose cubic fit has an inflection point outside the entire window!

5. Summary of the Topographic Classification Scheme

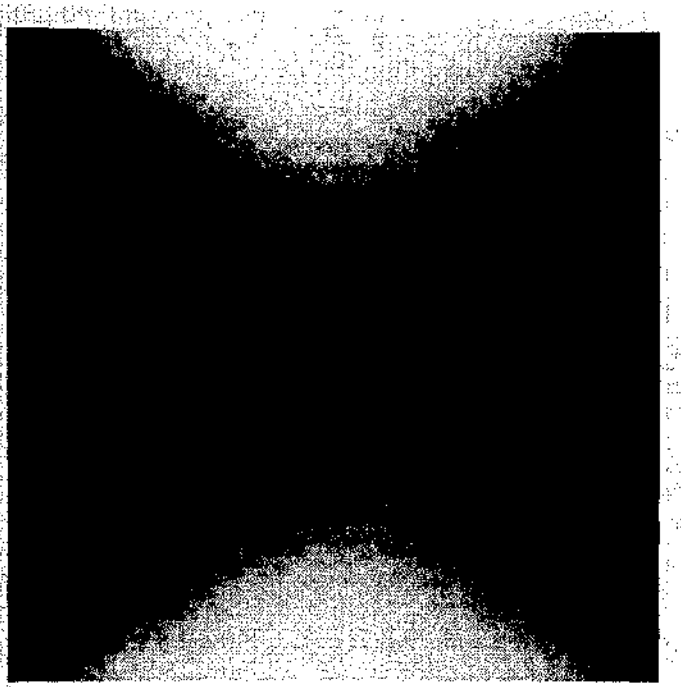
The scheme is a parallel process for topographic classification of every pixel, which can be done in one pass through the image. At each pixel of the image, the following four steps need to be performed.

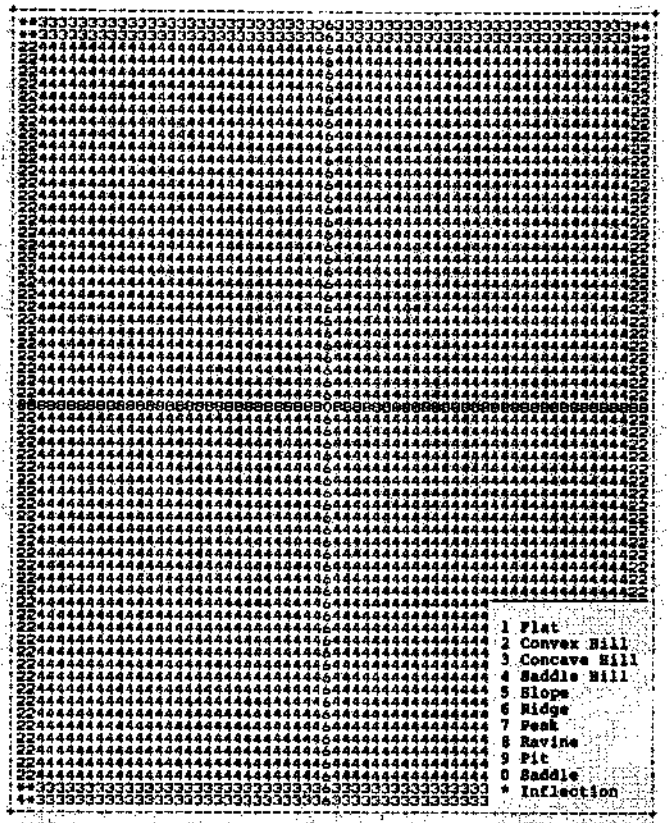
1. Calculate the fitting coefficients, k_1 through k_{10}, of a two-dimensional cubic polynomial in an n-by-n neighborhood around the pixel. These coefficients are easily computed by convolving the appropriate masks over the image.
2. Use the coefficients calculated in step 1 to find the gradient, gradient magnitude, and the eigenvalues and eigenvectors of the Hessian at the center of the pixel's neighborhood, (0, 0).
3. Search in the direction of the eigenvectors calculated in step 2 for a zero-crossing of the first directional derivative within the pixel's area. (If the eigenvalues of the Hessian are equal and nonzero, then search in the Newton direction.)
4. Recompute the gradient, gradient magnitude, and values of second directional derivative extrema at each zero-crossing. Then apply the labeling scheme as described in Sections 3.1–3.4.

6. Examples

In this section, we show the results of the topographic primal sketch on several test images, three of which are simply described mathematical surfaces

Fig. 6. A. Saddle surface. B. Topographic labeling of saddle surface.





and two of which are real images. We will also examine how the size of the window affects the results of the classifier.

6.1. Saddle Surface

A perfect saddle surface of size 64 × 64 having no noise can be generated by the equation

$$f(r, c) = r^2 - c^2.$$

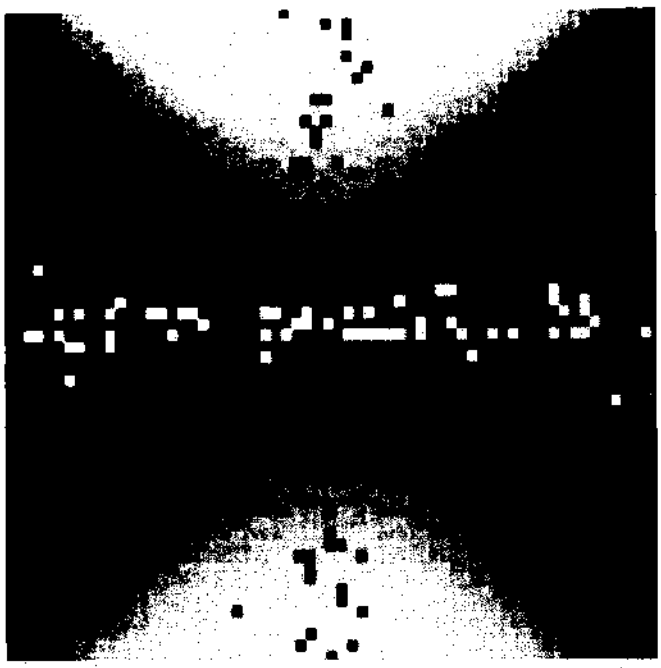
Taking the origin at image coordinates (32, 33), the surface plot is as illustrated in Fig. 2, and the gray-level image of the saddle surface is as illustrated in Fig. 6A. The results of the classifier are shown in Fig. 6B. Each number in the figure represents the label assigned the pixel by the classifier. As expected, a ridge-line one pixel in width was found running north-south, and, orthogonal to the ridge-line, a ravine-line one pixel in width was found. The center pixel of the surface was correctly classified as a saddle point. All other pixels on the surface were correctly classified as saddle hillsides.

Next, we add Gaussian noise to the saddle surface. The noise has a mean of 0.0 and standard deviation of 4.0. The results of the classifier using different sized windows (5 × 5, 7 × 7, 9 × 9, 11 × 11) on the noisy surface are shown in Fig. 7. As the window size increases, the results of the classifier improve dramatically. The classification resulting from the 11 × 11 window is almost identical to the classification done on the original, perfectly smooth surface. This would seem to suggest using as large a window size as possible, but in the next example we will show that this is not always a good idea.

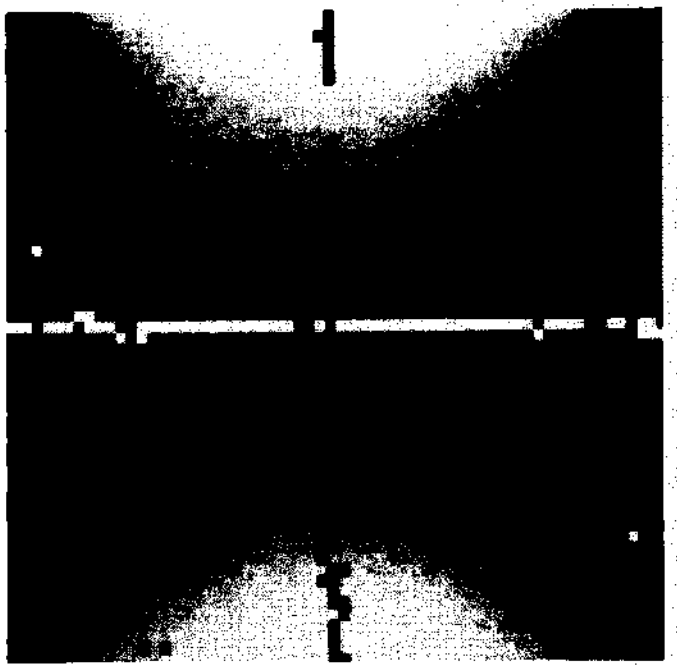
6.2. Ridges and Valleys

A series of ridges and valleys can be generated across the column direction by the following equation:

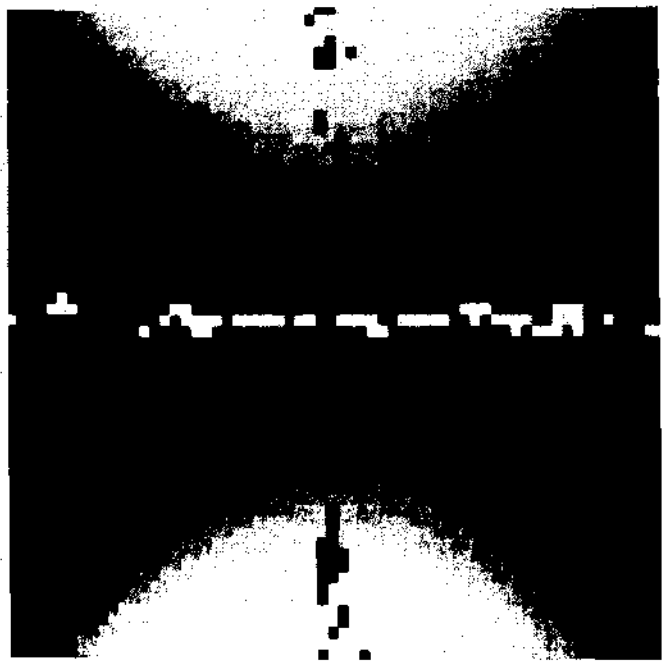
Fig. 7. Neighborhood topographic labeling of noisy saddle surface showing ridge (black) and ravine (white). A. 5 × 5 window. B. 7 × 7 window. C. 9 × 9 window. D. 11 × 11 window.



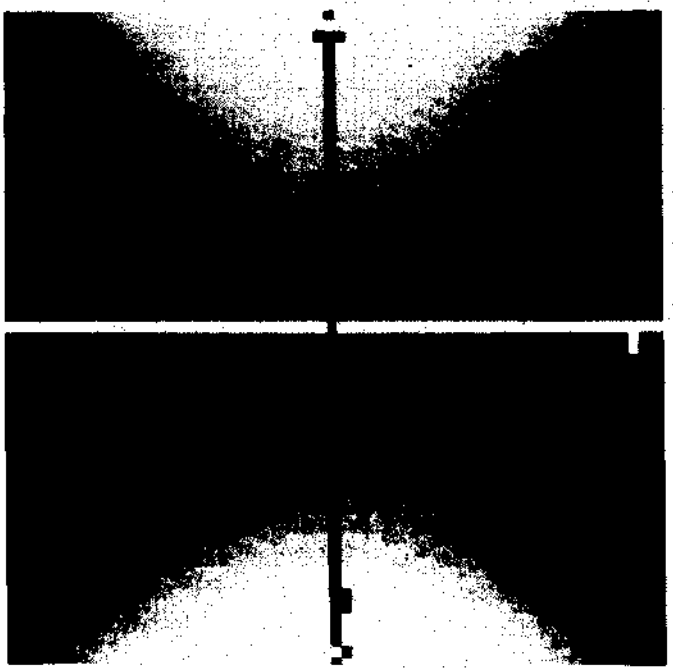
7a



7c

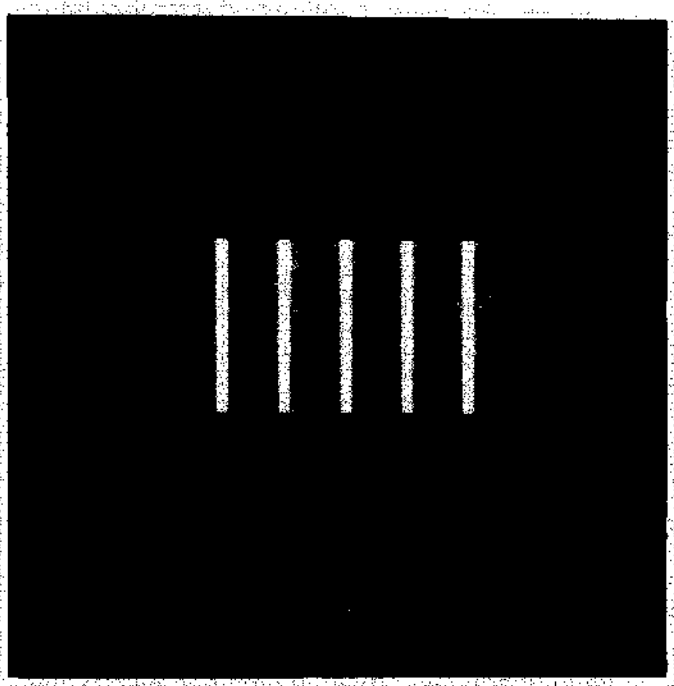


7b

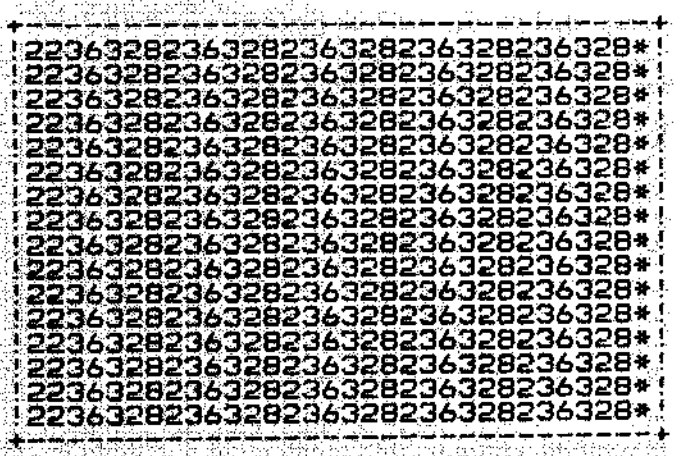


7d

Fig. 8. Image and topographic labeling of sine waves. A. Image. B. Topographic labeling with 5 × 5 window. C. Labeling with 9 × 9 window. D. Labeling with 13 × 13 window.



8a

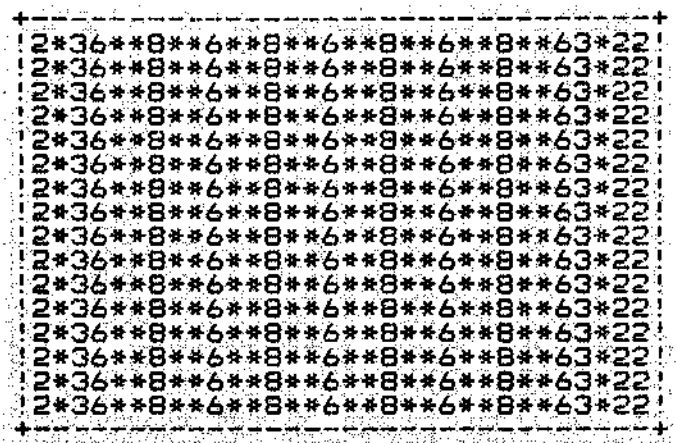


8b

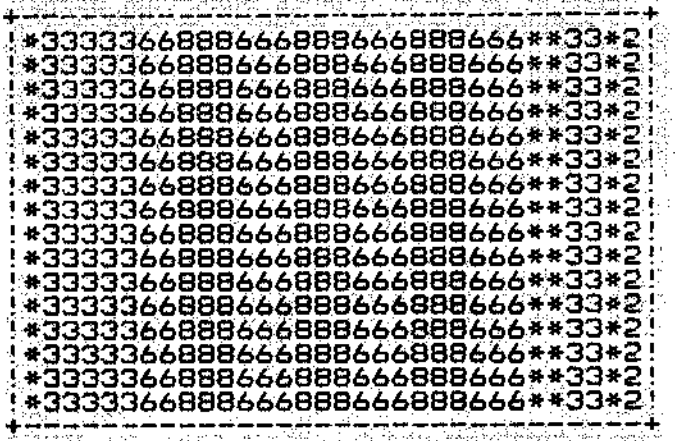
$$f(r, c) = \begin{cases} 1 \text{ if col} = 1, 7 \\ 2 \text{ if col} = 2, 6 \\ 3 \text{ if col} = 3, 5 \\ 4 \text{ if col} = 4 \end{cases}$$

where col = mod(c − 1, 7) + 1.

It is easy to see from the above equation that every row will be the same with a ridge occurring every sixth pixel beginning in column 4, and a ravine occurring every sixth pixel beginning in column 7. The gray level plot of the 16 × 32 image is shown in Fig. 8A. As expected, the ridges and ravines are correctly identified. The results of the classifier using window sizes 5 × 5, 9 × 9, and 13 × 13 are found in Figs. 8B, C, and D respectively. As the window size increases, the results of the classifier become less accurate. This result is exactly the opposite of what happened on the saddle surface. The reason is that this surface is much more "busy" than the saddle surface. The larger window size on this particular surface results in too many complexities for the cubic fit to handle.



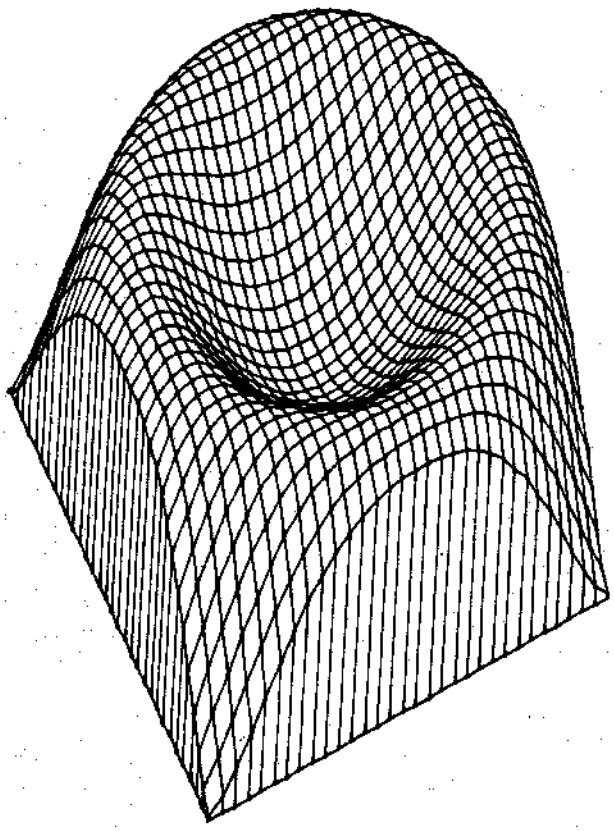
8c

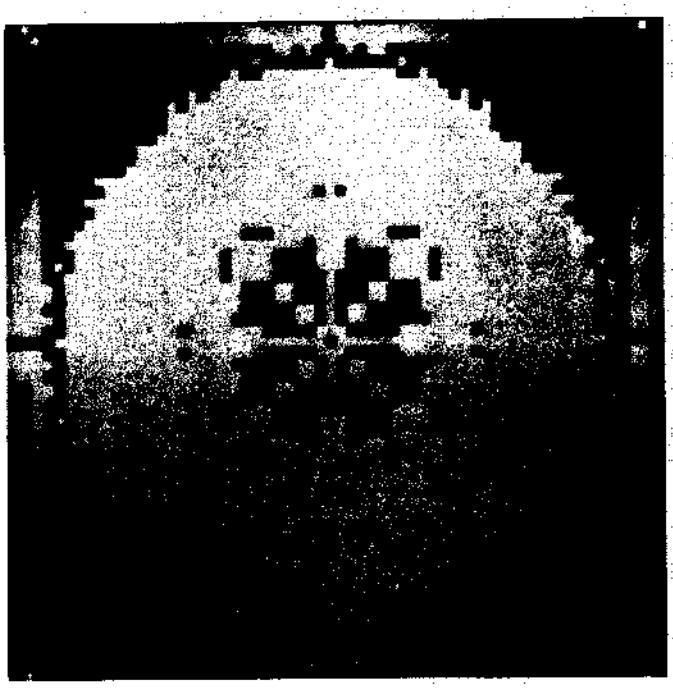


8d

The conclusion is that the window size used should be a function of the noise and the complexity of the image surface. One should use as big a window size as possible without allowing the complexity of the

Fig. 9. Surface of revolution. A. Surface plot. B. Ravine ridge (black) and ravine (white) labeling of the surface.





surface to degrade the cubic fit to any significant degree.

6.3. Surface of Revolution

A surface of revolution of size 64 × 64 with no noise can be generated by the equation

$$f(r, c) = k * \sin(0.5 * (r^2 + c^2)),$$

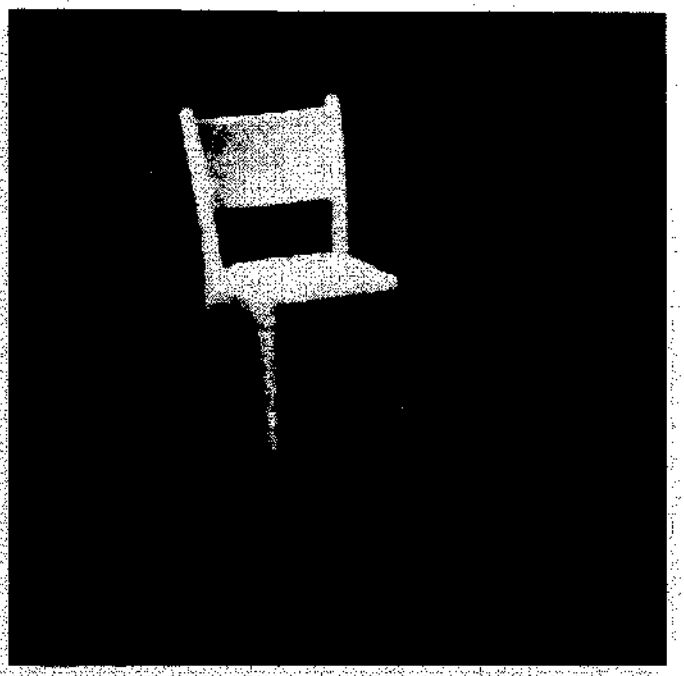
with origin at image coordinates (32, 32). The surface plot is illustrated in Fig. 9A. The topographic labeling on this surface shows some surprising results. A continuum of ridges and a continuum of ravines are found on the surface (see Fig. 9B). The reasons for these ridge and ravine points were discussed in Section 2.4. Also, the local cubic fits are very poor on this surface of revolution. This leads to some unexpected results, such as the peaks found on the rim of the surface where the ridges and ravines come together. The pixels labeled saddle on the image occur at locations where both a ridge and ravine were detected within the same pixel.

Notice that the labelings produced are not perfectly symmetric, as one would expect on a radially symmetric surface. The reason for this is that the cubic surface estimation is done with rectangular windows, which produces different cubic approximations at the same radial distance from the axis of revolution and hence radially unsymmetric labeling. Symmetric labeling would be produced by using a circular window, but choosing a particular window shape requires a priori knowledge of the nature of the image surface.

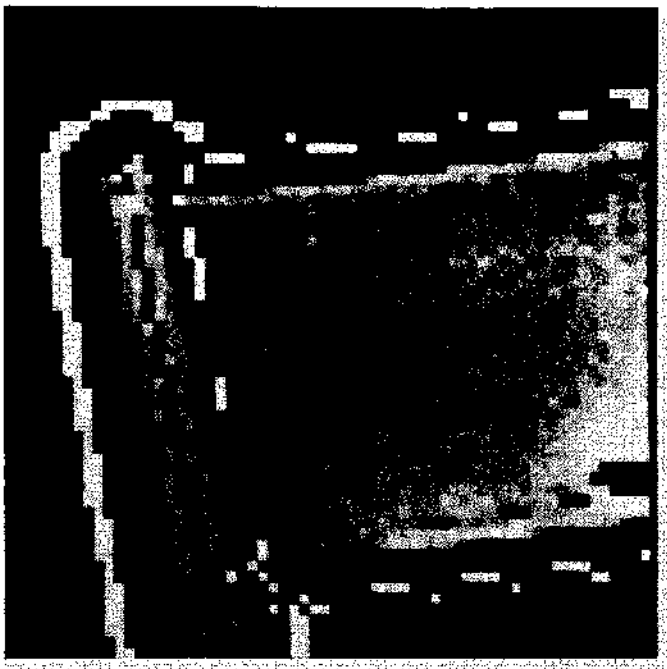
6.4. Real Image

In this section, we show the results of the classifier on two real images. The results on the top left corner of a chair image are illustrated in Fig. 10B. The results on the upper middle section of the bin of machine parts are illustrated in Fig. 11B. The vari-

Fig. 10. Results of the classifier on a real image. A. Chair. B. Upper left corner of chair. C. Ridges (black) and ravines (white). D. Hillside (white).

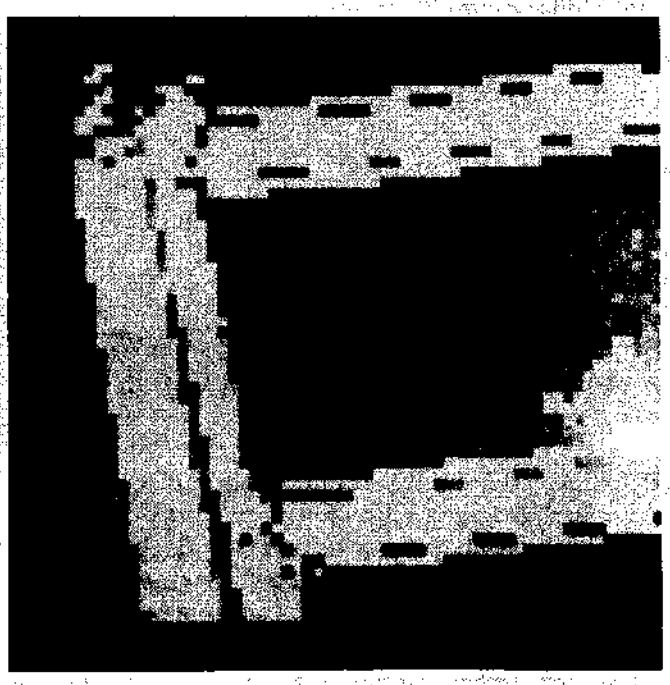


10a



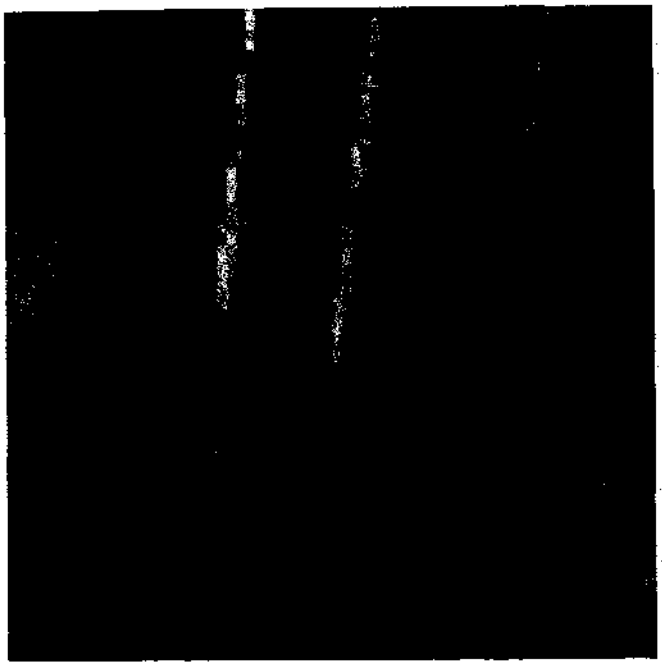
10c

10b

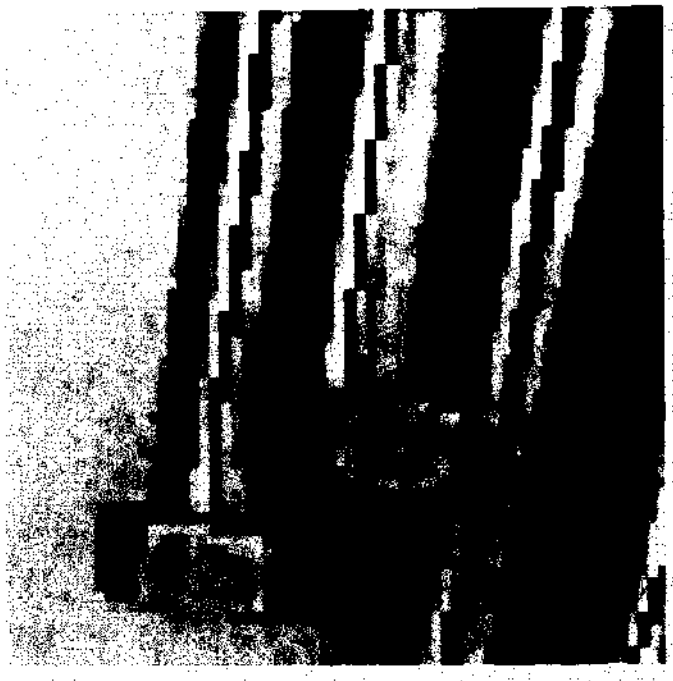


10d

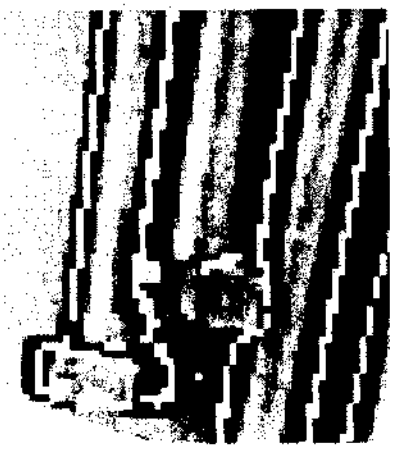
Fig. 11. A. Screw. B. Ridges (black). C. Ravines (white). D. Convex hillside (white). E. Concave hillside (black).



11a



11b

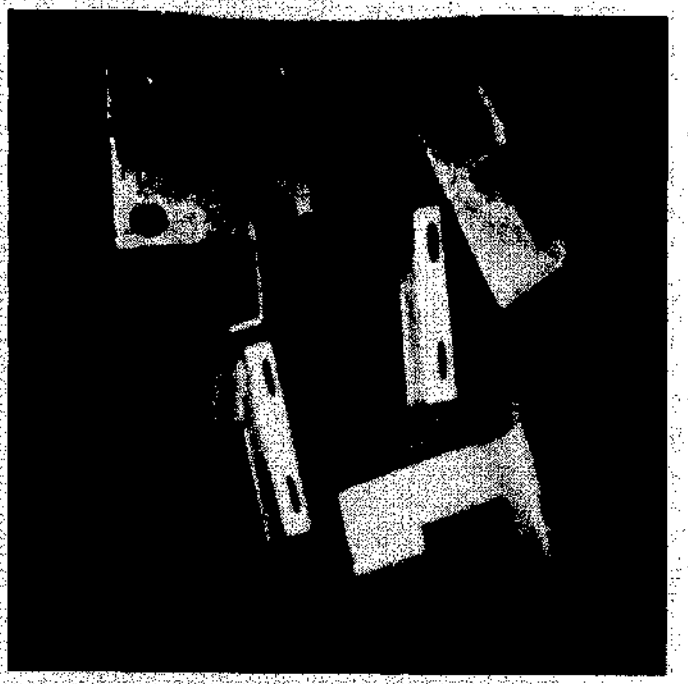


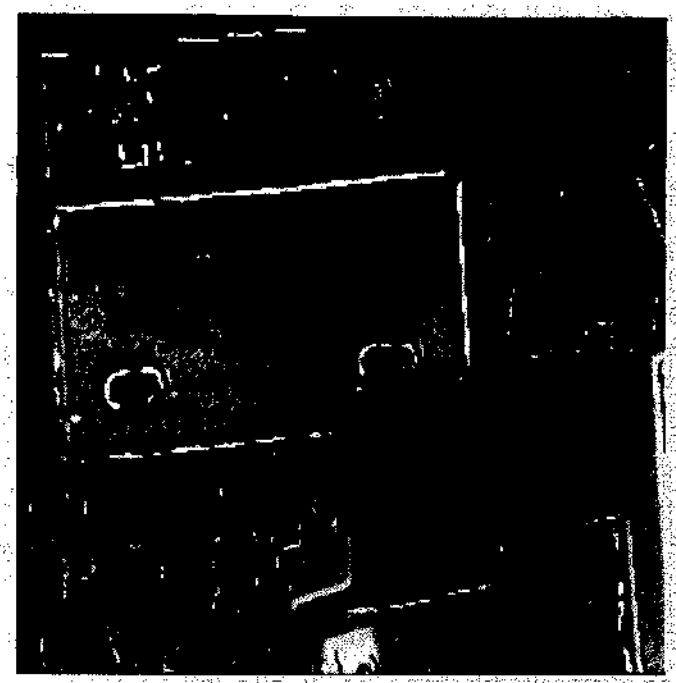
11c

11d

11e

Fig. 12. A. Machine parts. B. Upper left corner showing subimage ridges (black) and ravines (white). C. Center showing subimage ridges (black) and ravines (white).





ous nonflat labels in the backgrounds of the images are caused by very slight dips and rises in the cubic surface fit. These may be cleaned up by requiring the eigenvalues to be above a certain threshold to be considered nonzero (see Fig. 10C). Figure 12 shows the labeling on an image of manufacturing parts. Notice how the highlighting can occur depending on the positioning of the parts. The ridge labels are quite useful for determining where the highlighting occurs.

7. Conclusions

In this paper, we have given a precise mathematical description of the various topographic structures that occur in a digital image and have called the classified image the topographic primal sketch. Our set of topographic categories is invariant under gray tone, monotonically increasing transformations and consists of peak, pit, ridge, ravine, saddle, flat, and hillside, with hillside being broken down further into the subcategories inflection point, slope, convex hill,

concave hill, and saddle hill. The hillside subcategories are not invariant under the monotonic transformations.

The topographic label assigned a pixel is based on the pixel's first- and second-order directional derivatives. We use a two-dimensional cubic polynomial fit based on the local facet model to estimate the directional derivatives of the underlying gray tone intensity surface. The calculation of the extrema of the second directional derivative can be done efficiently and stably by forming the Hessian matrix and calculating its eigenvalues and their associated eigenvectors. Strict, local, one-dimensional extrema (such as pit, peak, ridge, ravine, and saddle) are found by searching for a zero-crossing of the first directional derivative in the directions of extreme second directional derivative (the eigenvectors of the Hessian). We have also identified another direction of interest, the Newton direction, which points toward the extremum of a quadratic surface. The classification scheme was found to give satisfactory results on a number of test images.

7.1. Directions for Further Research

Further research on the topographic primal sketch needs to be done to (1) develop better basis functions, (2) make use of fitting error, (3) find a solution for the ridge (ravine) continuum problem, and (4) develop techniques for grouping of the topographic structures. Basis functions worth considering include trigonometric polynomials, polynomials of higher order, and piecewise polynomials of lower order than cubic. The basis functions problem is to find a set of basis functions and an associated inner product for least-squares approximation that can correctly replicate all common image surface features and be simultaneously computationally efficient and numerically stable.

Fitting error needs to be used in deciding into which class a pixel falls. Noise causes the fitting error to increase, and increased fitting error increases the uncertainty of the labeling. Also, global knowledge of how the topographic structures fit together could be used to correct the misclassification error caused by noise. The way the neighborhood size affects the surface fitting error and the classification scheme needs to be investigated in detail.

The ridge (ravine) continuum problem needs to be solved. It may be that there is no way to distinguish between a true ridge and a ridge continuum using only the values of partial derivatives at a point. The solution may require complete use of the partial derivatives in a local area about the pixel.

Most important for the use of the primal sketch in a general robotics computer vision system is the development of techniques for grouping and assembling topographically labeled pixels to form the primitive structures involved in higher-level matching and correspondence processes. How well can stereo correspondence or frame-to-frame time-varying image correspondence tasks be accomplished using the primitive structures in the topographic primal sketch? How effectively can the topographic sketch be used in undoing the confounding effects of shading and shadowing? How well will the primitive structures in the topographic sketch perform in the two-dimensional to three-dimensional object-matching process?

References

Ehrich, R. W., and Foith, J. P. 1978. Topology and semantics of intensity arrays. *Computer vision systems*, New York: Academic, pp. 111–128.

Grender, G. C. 1976. TOPO III: A Fortran program for terrain analysis. *Comput. Geosci.* 2:195–209.

Haralick, R. M. 1980. Edge and region analysis for digital image data. *Comput. Graphics Image Processing* 12(1): 60–73.

Haralick, R. M. 1981. The digital edge. *Proc. 1981 Conf. Pattern Recognition Image Processing*. New York: IEEE Computer Society, pp. 285–294.

Haralick, R. M. 1982. Zero-crossing of second directional derivative edge operator. *SPIE Proc. Robot Vision*. Bellingham, Wa.: SPIE.

Hsu, S., Mundy, J. L., and Beaudet, P. R. 1978. Web representation of image data. *4th Int. Joint Conf. Pattern Recognition*. New York: IEEE Computer Society, pp. 675–680.

Johnston, E. G., and Rosenfeld, A. 1975. Digital detection of pits, peaks, ridges, and ravines. *IEEE Trans. Syst. Man Cybern.*, July, pp. 472–480.

Laffey, T. J., Haralick, R. M., and Watson, L. T. 1982. Topographic classification of digital image intensity surfaces. *Proc. IEEE Workshop Comput. Vision: Theory Contr.* New York: IEEE Computer Society, pp. 171–177.

Lee, H. C., and Fu, K. S. 1981. The GLGS image representation and its application to preliminary segmentation and pre-attentive visual search. *Proc. 1981 Conf. Pattern Recognition Image Processing*. New York: IEEE Computer Society, pp. 256–261.

Marr, D. 1976. Early processing of visual information. *Philosophical Trans. Royal Soc. London* B 275:483–524.

Marr, D. 1980. Visual information processing: The structure and creation of visual representations. *Philosophical Trans. Royal Soc. London* B 290:199–218.

Paton, K. 1975. Picture description using Legendre polynomials. *Comput. Graphics Image Processing* 4(1): 40–54.

Peuker, T. K., and Johnston, E. G. 1972 (Nov.). Detection of surface-specific points by local parallel processing of discrete terrain elevation data. Tech. Rept. 206. College Park Md.: University of Maryland Computer Science Center.

Peuker, T. K., and Douglas, D. H. 1975. Detection of surface-specific points by local parallel processing of discrete terrain elevation data. *Comput. Graphics Image Processing* 4(4):375–387.

Rutishauser, H. 1971. Jacobi method for real symmetric matrix. *Handbook for automatic computation, volume II, linear algebra, ed.* J. H. Wilkinson and C. Reinsch. New York: Springer-Verlag.

Strang, G. 1980. *Linear algebra and its applications,* 2nd ed. New York: Academic, pp. 243–249.

Toriwaki, J., and Fukumura, T. 1978. Extraction of structural information from grey pictures. *Comput. Graphics Image Processing* 7(1):30–51.

Reprinted from *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume PAMI-9, Number 4, July 1987, pages 532-550.

Image Analysis Using Mathematical Morphology

ROBERT M. HARALICK, FELLOW, IEEE, STANLEY R. STERNBERG, AND XINHUA ZHUANG

***Abstract*—For the purposes of object or defect identification required in industrial vision applications, the operations of mathematical morphology are more useful than the convolution operations employed in signal processing because the morphological operators relate directly to shape. The tutorial provided in this paper reviews both binary morphology and gray scale morphology, covering the operations of dilation, erosion, opening, and closing and their relations. Examples are given for each morphological concept and explanations are given for many of their interrelationships.**

***Index Terms*—Closing, dilation, erosion, filtering, image analysis, morphology, opening, shape analysis.**

I. Introduction

MATHEMATICAL morphology provides an approach to the processing of digital images which is based on shape. Appropriately used, mathematical morphological operations tend to simplify image data preserving their essential shape characteristics and eliminating irrelevancies. As the identification of objects, object features, and assembly defects correlate directly with shape, it becomes apparent that the natural processing approach to deal with the machine vision recognition process and the visually guided robot problem is mathematical morphology.

Morphologic operations are among the first kinds of image operators used. Kirsch, Cahn, Ray, and Urban [13] discussed some binary 3×3 morphologic operators. Other early papers include Unger [37] and Moore [21].

Machines which perform morphologic operations are not new. They are the essence of what cellular logic machines such as the Golay logic processor [8], Diff3 [9], PICAP [15], the Leitz Texture Analysis System TAS [14], the CLIP processor arrays [3], and the Delft Image Processor DIP [6] all do. A number of companies now manufacture industrial vision machines which incorporate video rate morphological operations. These companies include Machine Vision International, Maitre, Synthetic Vision Systems, Vicom, Applied Intelligence Systems, Inc., and Leitz.

The 1985 IEEE Computer Society Workshop on Computer Architecture For Pattern Analysis and Image Database Management had an entire session devoted to computer architecture specialized to perform morphological operations. Papers included those by McCubbrey and Lougheed [19], Wilson [39], Kimmel, Jaffe, Manderville, and Lavin [12], Leonard [16], Pratt [27], and Haralick [11]. Gerritsen and Verbeek [7] show how convolution followed by a table look up operation can accomplish binary morphologic operations.

But although the techniques are being used in the industrial world, the basis and theory of mathematical morphology tend to be (with the exception of the highly mathematical books by Matheron [18] and Serra [31]) not covered in the textbooks or journals which discuss image processing or computer vision. It is the intent of this tutorial to help fill this void.

The paper is divided into three parts. Section II discusses the basic operations of dilation and erosion in an N-dimensional Euclidean space. Section III discusses the derived operations of opening and closing. Section IV gives the corresponding definition for the dilation and erosion operations for gray tone images and shows how with these definitions all the properties of dilation and erosion, opening, and closing previously derived and explained in Sections II and III hold.

II. Dilation and Erosion

The language of mathematical morphology is that of set theory. Sets in mathematical morphology represent the shapes which are manifested on binary or gray tone images. The set of all the black pixels in a black and white image, (a binary image) consitutes a complete description of the binary image. Sets in Euclidean 2-space denote foreground regions in binary images. Sets in Euclidean 3-space may denote time varying binary imagery or static gray scale imagery as well as binary solids. Sets in higher dimensional spaces may incorporate additional image information, like color, or multiple perspective imagery. Mathematical morphological transformations apply to sets of any dimensions, those like Euclidean N-space, or those like its discrete or digitized equivalent, the set of N-tuples of integers, Z^N. For the sake of simplicity we will refer to either of these sets as E^N.

Those points in a set being morphologically transformed are considered as the selected set of points and those in the complement set are considered as not selected. Hence, morphology from this point of view is binary morphology. We begin our discussion with the binary morphological operations of dilation and erosion.

A. Dilation

Dilation is the morphological transformation which combines two sets using vector addition of set elements.

Manuscript received January 24, 1986; revised May 28, 1986. Recommended for acceptance by S. W. Zucker.

R. M. Haralick is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195.

S. R. Sternberg is with Machine Vision International, Ann Arbor, MI 48104.

X. Zhuang is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195, on leave from the Zhejiang Institute of Computing, Zhejiang, China.

IEEE Log Number 8715075.

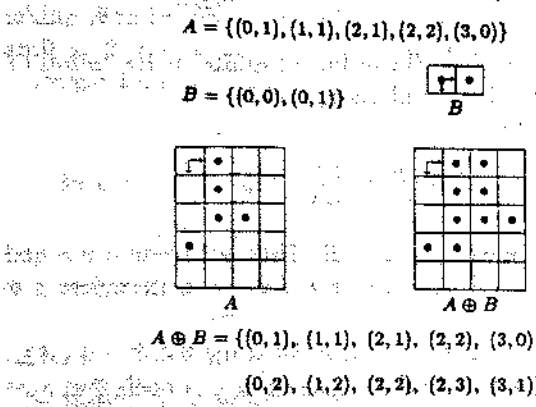
0-8186-9102-6/91 $1.00 © 1987 IEEE

If A and B are sets in N-space (E^N) with elements a and b, respectively, $a = (a_1, \cdots, a_N)$ and $b = (b_1, \cdots, b_N)$ being N-tuples of element coordinates, then the dilation of A by B is the set of all possible vector sums of pairs of elements, one coming from A and one coming from B.

Definition 1: Let A and B be subsets of E^N. The dilation of A by B is denoted by $A \oplus B$ and is defined by

$$A \oplus B = \{c \in E^N \mid c = a + b \text{ for some } a \in A \text{ and } b \in B\}.$$

Example: This illustrates an instance of the dilation operation. The coordinate system we use for all the examples in the next few sections is (row, column).



Dilation as a set theoretic operation was proposed by H. Minkowski [20] to characterize integral measures of certain open (sparse) sets. Dilation as an image processing operation was employed by several early investigators in image processing as smoothing operations [13], [37], [21], [8], [28], [29]. Dilation as an image operator for shape extraction and estimation of image parameters was explored by Matheron [18] and Serra [30]. All of these early applications dealt with binary images only.

Matheron uses the term "dilatation" for dilation and both Matheron and Serra define dilation slightly differently. In essense, they define the dilation of A by B as the set $\{c \in E^N \mid c = a - b \text{ for some } a \in A \text{ and } b \in B\}$.

In morphological dilation, the roles of the sets A and B are symmetric, that is, the dilation operation is commutative because addition is commutative.

Proposition 2:

$$A \oplus B = B \oplus A.$$

Proof:

$$\begin{aligned} A \oplus B &= \{c \mid c = a + b \text{ for some } a \in A,\ b \in B\} \\ &= \{c \mid c = b + a \text{ for some } a \in A,\ b \in B\} \\ &= B \oplus A. \end{aligned}$$

In practice, A and B are handled quite differently. The first operand A is considered as the image undergoing analysis, while the second operand B is referred to as the structuring element, to be thought of as constituting a single shape parameter of the dilation transformation. In the remainder of the paper, we will refer to A as the image and B as the structuring element.

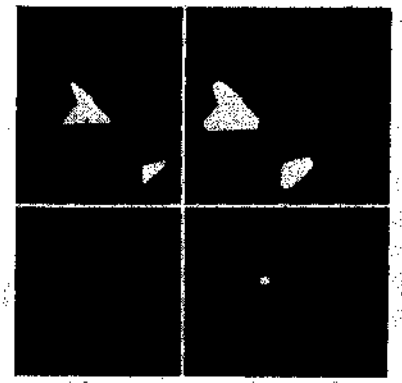


Fig. 1. The upper left shows the input image consisting of two objects. The lower right shows the octagonal structuring element. The upper right shows the input image dilated by the octagonal structuring element.

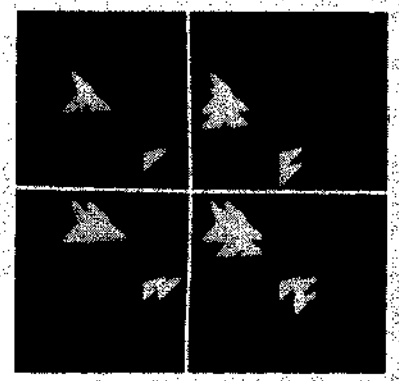


Fig. 2. The upper left shows the input image consisting of two objects. The upper right shows the input image dilated by the structuring element $\{(0, 0), (14, 0)\}$. The lower left shows the input image dilated by the structuring element $\{(0, 0), (0, 14)\}$. The lower right shows the input image dilated by the structuring element $\{(0, 0), (14, 0), (0, 14)\}$.

Dilation by disk structuring elements correspond to isotropic swelling or expansion algorithms common to binary image processing. Dilation by small squares (3×3) is a neighborhood operation easily implemented by adjacency connected array architectures (grids) and is the one many image processing people know by the name "fill," "expand," or "grow." Some example dilation transformations are illustrated in Figs. 1 and 2.

Neighborhood connected image processors such as CLIP [4], Cytocomputer [32], [34], and MPP [1], [26] can implement some dilations (not all) by structuring elements larger than the neighborhood size by iteratively dilating with a sequence of neighborhood structuring elements. In particular, if image A is to be dilated by structuring element D which itself can be expressed as the dilation of B by C, then $A \oplus D$ can be computed as

$$A \oplus D = A \oplus (B \oplus C) = (A \oplus B) \oplus C$$

since addition is associative.

The form $(A \oplus B) \oplus C$ represents a considerable savings in number of operations to be performed when A is the image and $B \oplus C$ is the structuring element. The savings come about because a brute force dilation by $B \oplus C$ might take as many as N^2 operations while first dilating

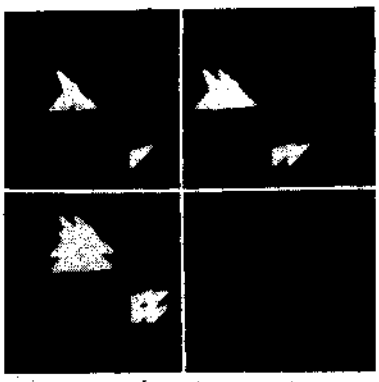


Fig. 3. The upper left shows the input image consisting of two objects. The upper right shows the input image dilated by the structuring element $\{(0, 0)\ (0, 14)\}$. The lower left shows the input image dilated by the structuring element $\{(0, 0), (14, 0), (0, 14)\ (14, 14)\}$, which is shown in the lower right. Notice that the dilated image of the lower left can be obtained by dilating the image shown in the upper right by the structuring element $\{(0, 0), (14, 0)\}$. This is a consequence of the chain rule for dilations and because $\{(0, 0), (14, 0)\} \oplus \{(0, 0), (0, 14)\} = \{(0, 0), (0, 14), (14, 0), (14, 14)\}$.

A by B and then dilating the result by C could take as few as $2N$ operations, where N is the number of elements in B and in C. This computational complexity advantage is not as strong for machines which can implement dilations only as neighborhood operations.

Proposition 3:

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C.$$

Proof: $x \in A \oplus (B \oplus C)$ if and only if there exists $a \in A$, $b \in B$, and $c \in C$ such that $x = a + (b + c)$. $x \in (A \oplus B) \oplus C$ if and only if there exists $a \in A$, $b \in B$, and $c \in C$ such that $x = (a + b) + c$. But $a + (b + c) = (a + b) + c$ since addition is associative. Therefore, $A \oplus (B \oplus C) = (A \oplus B) \oplus C$.

Proposition 3 is commonly referred to as the "chain rule" for dilations. An example of performing a dilation transformation as a chain of dilations is shown in Fig. 3. Notice that this dilation transformation which can be done as a chain of dilations is not able to be done as a chain of neighborhood operations.

Since dilation is commutative, the order of application of the constituent dilations is immaterial.

Dilating an image as an iterative sequence of neighborhood operations is not necessarily the most efficient or universal approach to implementing the dilation transformation. For example, not all structuring elements can be decomposed into iterative neighborhood dilations. An example of a dilation transformation which cannot be implemented as an iterative sequence of neighborhood operations is the dilation by any of the structuring elements $\{(0, 0), (0, 14)\}$, $\{(0, 0), (14, 0)\}$ or $\{(0, 0), (0, 14), (14, 0)\}$ which are shown in Fig. 2.

Also, the implementation may not be particularly efficient in terms of processing time or computer hardware requirements. An alternative involves considering dilations in terms of image translations. So first we need the definition for translation.

Definition 4: Let A be a subset of E^N and $x \in E^N$. The translation of A by x is denoted by $(A)_x$ and is defined by

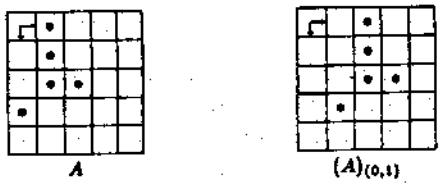
$$(A)_x = \{c \in E^N \mid c = a + x \text{ for some } a \in A\}.$$

Example: This illustrates an instance of translation.

$$A = \{(0, 1), (1, 1), (2, 1), (2, 2), (3, 0)\}$$

$$x = (0, 1)$$

$$(A)_x = \{(0, 2), (1, 2), (2, 2), (2, 3), (3, 1)\}.$$



The dilation of A by B can be computed as the union of translations of A by the elements of B.

Proposition 5:

$$A \oplus B = \bigcup_{b \in B} (A)_b.$$

Proof: Suppose $x \in A \oplus B$. Then for some $a \in A$ and $b \in B$, $x = a + b$. Hence, $x \in (A)_b$ and therefore $x \in \bigcup_{b \in B} (A)_b$.

Suppose $x \in \bigcup_{b \in B} (A)_b$. Then for some $b \in B$, $x \in (A)_b$. But $x \in (A)_b$ implies there exists an $a \in A$ such that $x = a + b$. Now by definition of dilation, $a \in A$, $b \in B$, and $x = a + b$ imply $x \in A \oplus B$.

Historically, the dilation transformation was defined by Minkowski in this manner, hence the name Minkowski addition is applied to Proposition 5 in the literature (for example, see [10]). Unfortunately, Minkowksi failed to define the dual of his set addition operation, and Minkowski subtraction expressed as the intersection of translations of A by the elements of B was not formally proposed until done so by Hadwiger.

Proposition 5 emphasizes the role of image shifting to implement dilation. In pipeline digital image processors employing raster scanning, image shifting is accomplished by delay elements in the transmission path. But delay elements can only cause an image shift in a direction opposite to the row scanning direction of the raster conversion. Thus it is important to know that dilating a shifted image, which arises from previous pipeline delays, shifts the dilated result by an equivalent amount. This fact permits pipeline processors to successively operate morphologically on shifted images and to undo the total resulting shift by performing an opposite shift by the scrolling operation in the output image buffer. We call this property the translation invariance of dilation.

Translation Invariance of Dilation Proposition 6:

$$(A)_x \oplus B = (A \oplus B)_x.$$

Proof: $y \in (A)_x \oplus B$ if and only if for some $z \in (A)_x$ and $b \in B$, $y = z + b$. But $z \in (A)_x$ if and only if $z = a + x$ for some $a \in A$. Hence, $y = (a + x) + b = (a +$

$b) + x$. Now by definition of dilation and translation $y \in (A \oplus B)_x$.

A corollary to Proposition 6 applies to dilations implemented through the chain rule (Proposition 3). The corollary states that shifting any one of the structuring elements in a dilation decomposition shifts the dilated image by an equivalent amount.

Corollary 7:

$$A \oplus B_1 \oplus \cdots \oplus (B_n)_x \oplus \cdots \oplus B_N = (A \oplus B_1 \oplus \cdots \oplus B_n \oplus \cdots \oplus B_N)_x.$$

Image shift can be compensated for in the definition of the structuring element. In particular, let the structuring element B be compensating for a shift in the image A by taking B to be shifted in the opposite direction. Then the shift in B compensates for the shift in A.

Proposition 8:

$$(A)_x \oplus (B)_{-x} = A \oplus B.$$

Proof:

$$(A)_x \oplus (B)_{-x} = (A \oplus (B)_{-x})_x = (A \oplus B)_{x-x} = A \oplus B.$$

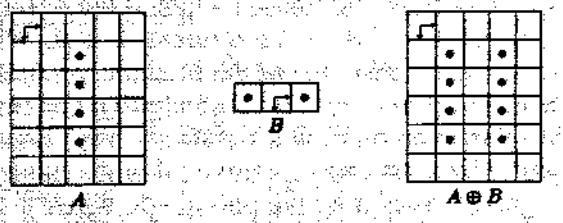
Similarly, compensating shifts within the sequence of decomposed structuring element dilations can balance image shifts and cause an unshifted result.

Corollary 9:

$$(A)_x \oplus B_1 \oplus \cdots \oplus (B_n)_{-x} \oplus \cdots \oplus B_N = A \oplus B_1 \oplus \cdots \oplus B_n \oplus \cdots \oplus B_N$$

In addition to being commutative, the dilation transformation is necessarily extensive when the origin belongs to the structuring element, extensivity meaning that the dilated result contains the original.

Example: This example shows that when the origin is not in the structuring element B, it may happen that the dilation of A by B has nothing in common with A.



A corollary to Proposition 10 states that if the origin belongs to each of the structuring elements, in a dilation composition, each structuring element in the decomposition is necessarily contained in the original composed structuring element.

Corollary 11: If $0 \in B_1, \cdots, B_N$ then $B_m \in B_1 \oplus \cdots \oplus B_N$, $m = 1, \cdots, N$.

The dilation transformation is increasing, that is, containment relationships are maintained through dilation.

Dilation Is Increasing Proposition 12: $A \subseteq B$ implies $A \oplus D \subseteq B \oplus D$.

Proof: Suppose $A \subseteq B$. Let $x \in A \oplus D$. Then for some $a \in A$ and $d \in D$, $x = a + d$. Since $a \in A$ and $A \subseteq B$, $a \in B$. But $a \in B$ and $d \in D$ implies $x \in B \oplus D$.

Corollary 13: $A \subseteq B$ implies $D \oplus A \subseteq D \oplus B$.

The order of an image intersection operation and a dilation operation cannot be interchanged. Rather, the result of intersecting two images followed by a dilation of the intersection result is contained in the intersection of the dilation of the two images.

Proposition 14:

$$(A \cap B) \oplus C \subseteq (A \oplus C) \cap (B \oplus C)$$

$$(A \oplus (B \cap C) \subseteq (A \oplus B) \cap (A \oplus C)$$

Proof: Suppose $x \in (A \cap B) \oplus C$. Then for some $y \in A \cap B$ and $c \in C$, $x = y + c$. Now $y \in A \cap B$ implies $y \in A$ and $y \in B$. But $y \in A$, $c \in C$, and $x = y + c$ implies $x \in A \oplus C$; $y \in B$, $c \in C$, and $x = y + c$ implies $x \in B \oplus C$. Hence $x \in (A \oplus C) \cap (B \oplus C)$.

$(A \oplus (B \cap C) \subseteq (A \oplus B) \cap (A \oplus C)$ comes about immediately from the previous result since dilation is commutative.

On the other hand, the order of image union and dilation can be interchanged. The dilation of the union of two images is equal to the union of the dilations of these images.

Proposition 15:

$$(A \cup B) \oplus C = (A \oplus C) \cup (B \oplus C)$$

Proof:

$$(A \cup B) \oplus C = \bigcup_{x \in A \cup B} (C)_x = \left[\bigcup_{x \in A} (C)_x\right] \cup \left[\bigcup_{x \in B} (C)_x\right] = (A \oplus C) \cup (B \oplus C).$$

By the commutativity of dilation, we immediately have the following.

Corollary 16:

$$A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C).$$

This equality is significant. It permits for a further decomposition of a structuring element into a union of structuring elements. Previously we saw that the decomposition of a structuring element into the dilation of elemental structuring elements led to a chain rule for dilation. Here we see that decomposing a structuring element into the union of elemental structuring elements leads to another method of evaluating the dilation.

The distinction between structuring element decomposition by dilation and by union deserves further mention. The issue bears upon the efficiency of computing the dilations. Consider the structuring element of Fig. 4.

Structuring element B of Fig. 4 top consists of 16 points, hence it can be decomposed into the union of 16 structuring elements, each structuring element consisting

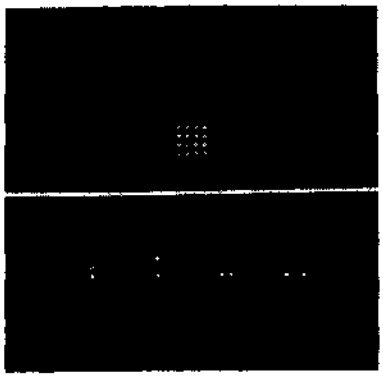


Fig. 4. This figure shows how the chain rule dilation decomposition can save operations. To dilate an image by the structuring element shown in the top half requires 15 operations. To dilate using the chain decomposition shown in the bottom half requires only 4 operations.

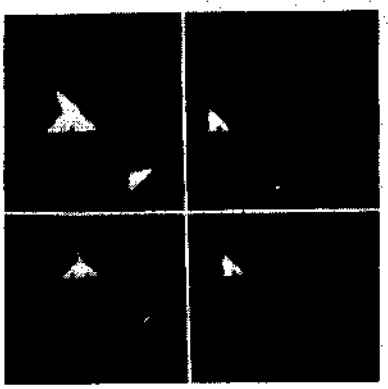


Fig. 5. The upper left shows the input image consisting of two blobs. The upper right shows the input image eroded by the structuring element $\{(0, 0), (-14, 0)\}$. The lower left shows the input image eroded by the structuring element $\{(0, 0)\,(0, -14)\}$. The lower right shows the input image eroded by the structuring element $\{(0, 0), (0, -14), (-14, 0)\}$.

of a single point which is suitably displaced from the origin. Dilation by a structuring element consisting of a single point is simply a shift of the original image, hence Proposition 15 becomes equivalent to the expression of Proposition 5 for the dilation, involving 15 shifts and 15 unions. By contrast, the decomposition of structuring element B into the four elemental structuring elements of Fig. 4 bottom permits dilation by B through the chain rule of Proposition 3. Here we see that only four shifts and four unions are required. Computationally, the difference involves a shift and union of the previously computed result in the case of Proposition 3's chain rule, while decomposition by union as in Proposition 15 independently accumulates the individual shifts of the original image.

B. Erosion

Erosion is the morphological dual to dilation. It is the morphological transformation which combines two sets using the vector subtraction of set elements. If A and B are sets in Euclidean N-space, then the erosion of A by B is the set of all elements x for which $x + b \in A$ for every $b \in B$. Some image processing people use the name shrink or reduce for erosion.

Definition 17: The erosion of A by B is denoted by $A \ominus B$ and is defined by

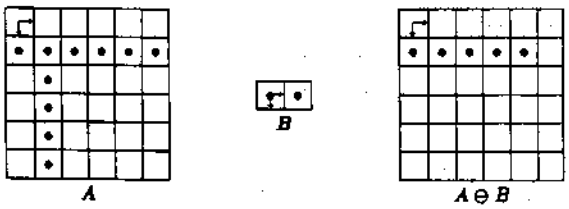
$$A \ominus B = \{x \in E^N \,|\, x + b \in A \text{ for every } b \in B\}$$

Example: This illustrates an instance of erosion.

$$A = \{(1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (3, 1), (4, 1), (5, 1),\}$$

$$B = \{(0, 0), (0, 1)\}$$

$$A \ominus B = \{(1, 0), (1, 1), (1, 2), (1, 3), (1, 4)\}$$



Expressed as a difference of elements a and b, Definition 17 becomes

$$A \ominus B = \{x \in E^N \,|\, \text{for every } b \in B, \text{ there exists an } a \in A \text{ such that } x = a - b\}$$

This is tne definition used for erosion by [10].

The utility of the erosion transformation is better appreciated when the erosion is expressed in a different form. The erosion of an image A by a structuring element B is the set of all elements x of E^N for which B translated to x is contained in A. In fact, this was the definition used for erosion by [18]. The proof is immediate from the definition of erosion and the definition of translation.

Proposition 18:

$$A \ominus B = \{x \in E^N \,|\, (B)_x \subseteq A\}.$$

Thus the structuring element B may be visualized as a probe which slides across the image A, testing the spatial nature of A at every point. Where B translated to x can be contained in A (by placing the origin of B at x), then x belongs to the erosion $A \ominus B$. The erosion transformation is illustrated in Fig. 5.

The careful reader should beware that the symbol $\ominus$ used by [31] does not designate erosion. Rather it designates the Minkowski subtraction which is the intersection of all translations of A by the elements $b \in B$. Whereas the dilation transformation and the Minkowski addition of sets are identical, the erosion transformation and the Minkowski subtraction differ in a significant way. Erosion of an image A by a structuring element B is the intersection of all translations of A by the points $-b$, where $b \in B$.

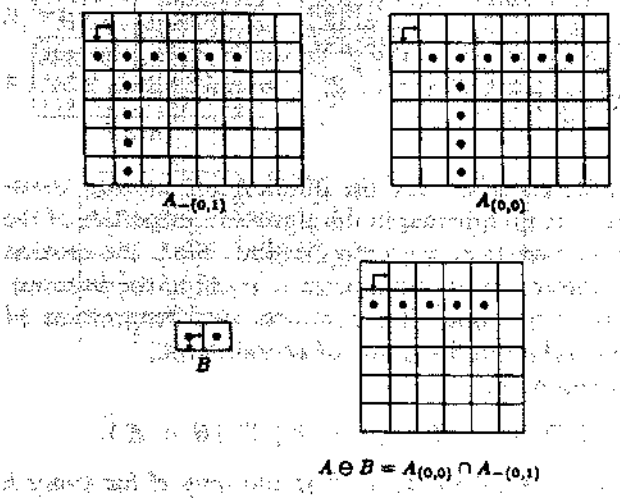
Proposition 19:

$$A \ominus B = \bigcap_{b \in B} (A)_{-b}.$$

Proof: Let $x \in A \ominus B$. Then for every $b \in B$, $x + b \in A$. But $x + b \in A$ implies $x \in (A)_{-b}$. Hence for every $b \in B$, $x \in (A)_{-b}$. This implies $x \in \bigcap_{b \in B} (A)_{-b}$.

Let $x \in \bigcap_{b \in B} (A)_{-b}$. Then for every $b \in B$, $x \in (A)_{-b}$. Hence, for every $b \in B$, $x + b \in A$. Now by definition of erosion $x \in A \ominus B$.

Example: This illustrates how erosion can by computed as an intersection of translates of A.

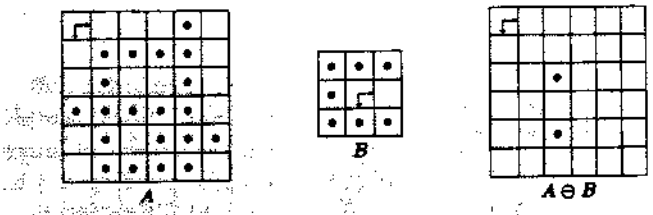


The erosion transformation is popularly conceived of as a shrinking of the original image. In set terms, the eroded set is often thought of as being contained in the original set. A transformation having this property is called anti-extensive. However, the erosion transformation is necessarily anti-extensive only if the origin belongs to the structuring element.

Proposition 20: If $0 \in B$, then $A \ominus B \subseteq A$.

Proof: Let $x \in A \ominus B$. Then $x + b \in A$ for every $b \in B$. Since $0 \in B$, $x + 0 \in A$. Hence $x \in A$.

Example: This illustrates how eroding with a structuring element which does not contain the origin can lead to a result which has nothing in common with the set being eroded.



Like dilation, erosion is a translation invariant and increasing transformation.

Translation Invariance of Erosion Proposition 21:

$$A_x \ominus B = (A \ominus B)_x$$

$$A \ominus B_x = (A \ominus B)_{-x}.$$

Proof: $y \in A_x \ominus B$ if and only if for every $b \in B$, $y + b \in A_x$. But $y + b \in A_x$ if and only if $y + b - x \in A$. Now, $y + b - x = (y - x) + b$. Hence for every $b \in B$, $(y - x) + b \in A$. By definition of erosion, $y - x \in A \ominus B$ and, therefore, $y \in (A \ominus B)_x$.

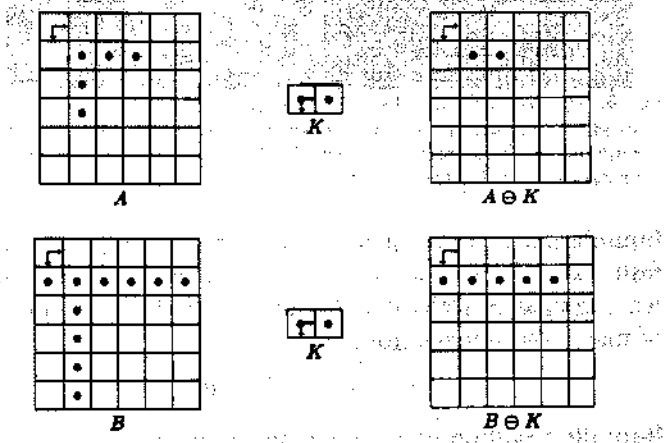
$y \in A \ominus B_x$ if and only if $y + b \in A$ for every $b \in B_x$. But $y + b \in A$ for every $b \in B_x$ if and only if $y - x \in A \ominus B$. Finally $y - x \in A \ominus B$ if and only if $y \in (A \ominus B)_{-x}$.

If image A is contained in image B, then the erosion of A is contained in the erosion of B.

Erosion Is Increasing Proposition 22: $A \subseteq B$ implies $A \ominus K \subseteq B \ominus K$.

Proof: Let $x \in A \ominus K$. Then $x + k \in A$ for every $k \in K$. But $A \subseteq B$. Hence, $x + k \in B$ for every $k \in K$. By definition of erosion, $x \in B \ominus K$.

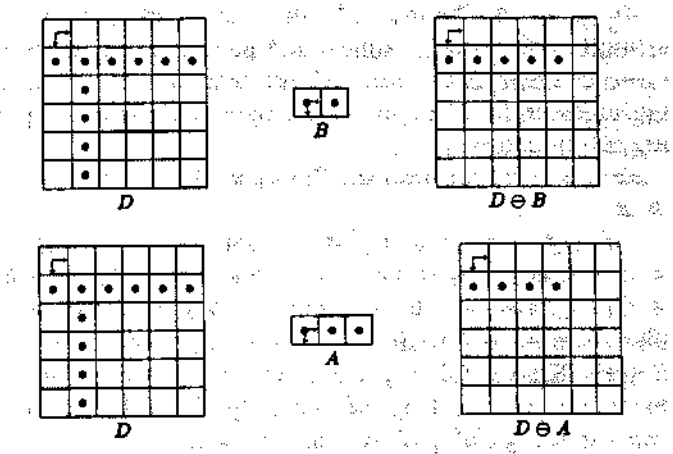
Example: This illustrates an instance showing the increasing property of erosion. On the other hand, if A and B are structuring elements and B is contained in A, then the erosion of an image D by A is necessarily more severe than erosion by B, that is, D eroded by A will necessarily be contained in D eroded by B.



Proposition 23: $A \supseteq B$ implies $D \ominus A \subseteq D \ominus B$.

Proof: Let $x \in D \ominus A$. Then $x + a \in D$ for every $a \in A$. But $B \subseteq A$. Hence, $x + a \in D$ for every $a \in B$. Now by definition of erosion, $x \in D \ominus B$.

Example: This illustrates an instance showing that larger structuring elements have a more severe effect than smaller ones on the erosion process.



This proposition leads to a natural ordering of the erosions by structuring elements having the same shape but different sizes. It is the basis of the morphological distance transformations. Fig. 6 illustrates these distance relationships.

The dilation and erosion transformations bear a marked similarity, in that what one does to the image foreground the other does to the image background. Indeed, their similarity can be formalized as a duality relationship. Recall that two operators are dual when the negation of a