



Prof. Dr.-Ing. Bärbel Mertsching
Universität Paderborn
Fakultät für Elektrotechnik, Informatik und Mathematik - GET Lab

Lecture Notes

Digital Image Processing

Prof. Dr.-Ing. Bärbel Mertsching

Note:

Current information on the course “Digital Image Processing I” is located at:

<http://getwww.uni-paderborn.de/teaching/dip-I>

Please submit ideas, corrections and comments on the material to the following eMail address:

dip-I-ws19_20-owner@get.upb.de

These lecture notes are protected by copyright. The constituted rights, specifically relating to the translation, reproduction, excerpt of figures, photomechanical or other type of reproduction and data storage in computers, also for using excerpts, are reserved.

Introduction

These lecture notes were produced with the L^AT_EX typesetting system. The typesetting of these lecture notes is based on portable document styles with imaPDF, a collection of packages and settings making the creation of pdf-files using pdfLatex, which was developed by Dipl.-Inform. Alan Wendland and the author, easier.

Developing teaching material requires a lot of commitment and the help of many individuals: The author gives her thanks to Ms. Karin Heyen, Ms. Astrid Haar and Ms. Tanja Steins for typesetting these lecture notes. Special thanks go to Dr.-Ing. Marcus Hund for his comprehensive assistance in maintaining and expanding imaPDF. Dipl.-Ing. Frank Schmidtmeier, Dipl.-Inf. Holger Täubig and Dipl.-Inf. Oliver Kutter also supported this project. My thanks also go to them.

Bärbel Mertsching

Contents

1	Introduction	1-I
1.1	Motivation	1-1
1.2	Digital Images	1-14
1.2.1	Basics of Computer Graphics	1-14
1.2.2	Raster versus Vector Graphics	1-18
1.2.3	Graphics File Formats	1-20
1.3	DIP: Application Examples	1-29
1.3.1	Gamma-Ray Imaging	1-32
1.3.2	X-ray Imaging	1-34
1.3.3	Imaging in the Ultraviolet Band	1-36
1.3.4	Imaging in the Visible and Infrared Band	1-38
1.3.5	Imaging in the Terahertz Band	1-47
1.3.6	Imaging in the Microwave Band	1-50
1.3.7	Imaging in the Radio Band	1-52
1.3.8	Other Imaging Modalities	1-54

1.4	Human Vision	1-61
1.4.1	Deception of the Human Eye?	1-63
1.4.2	Gestalt Principles of Visual Organization	1-68
1.4.3	Human Object Perception	1-78
1.4.4	Computer vision vs. biological vision	1-90
1.5	History of Computer Vision	1-91
1.6	Fundamental Steps in DIP	1-92

2 Image Formation and Image Models **2-I**

2.1	Electromagnetic Spectrum	2-1
2.2	Camera Models	2-5
2.2.1	Perspective Projection	2-6
2.2.2	Thin Lens Model	2-8
2.3	Image Formation	2-12
2.3.1	Formation of Natural Images	2-14
2.3.2	Intensity of sensor signals	2-15
2.3.3	Simple image formation model	2-18

2.4	Image Sampling and Quantization	2-21
2.4.1	Basic concepts in Sampling and quantization	2-21
2.4.2	Representing digital images	2-24
2.4.3	Aliasing and Moiré Patterns	2-32
3	Image Enhancement in the Spatial Domain	3-I
3.1	Basics	3-2
3.2	Gray-Level Transformation Functions	3-3
3.2.1	Linear Transformations	3-8
3.2.2	Logarithmic Transformations	3-9
3.2.3	Power-Law Transformations	3-11
3.2.4	Piecewise-Linear Functions	3-15
3.3	Histogram Processing	3-20
3.3.1	Histogram Equalization	3-23
3.3.2	Local Enhancement	3-43
3.3.3	Using Histogram Statistics for Image Enhancement	3-44
3.4	Spatial Filtering	3-52

3.5	Smoothing Spatial Filters	3-60
3.5.1	Linear Smoothing Filters	3-61
3.5.2	Ranking Filters	3-75
3.6	Sharpening spatial filters	3-79
3.6.1	Differentiation of Images	3-80
3.6.2	Second-Order Derivatives - The Laplacian	3-85
3.6.3	First-Order Derivatives - The Gradient	3-100
4	Image Enhancement in the Frequency Domain	4-I
4.1	Introduction to the Fourier Transform and the Frequency Domain	4-2
4.1.1	1D Fourier Transform and its Inverse	4-3
4.1.2	2D Fourier Transform and Its Inverse	4-18
4.1.3	Some Useful Theorems of the (2D) Fourier Transform	4-23
4.1.4	Correspondence between Filtering in the Spatial and Frequency Domains	4-31
4.1.5	Filtering in the Frequency Domain	4-40
4.2	Smoothing Frequency Domain Filter	4-50
4.2.1	Ideal Lowpass Filter	4-51

4.2.2	Butterworth Lowpass Filter	4-59
4.2.3	Gaussian Lowpass Filter	4-64
4.2.4	Additional examples of Lowpass Filtering	4-68
4.3	Sharpening Frequency-Domain Filters	4-70
4.3.1	Ideal Highpass Filters	4-74
4.3.2	Butterworth Highpass Filters	4-76
4.3.3	Gaussian Highpass Filters	4-78
4.4	Implementation of the Fourier Transform	4-80
4.4.1	Separability	4-80
4.4.2	Implementation of the inverse Fourier transform	4-82
4.4.3	Function Padding	4-84
4.4.4	Convolution and Correlation Theorems	4-96
4.4.5	Fast Fourier Transform	4-101
5	Color Image Processing	5-I
5.1	Color Fundamentals	5-2
5.2	Color Spaces	5-23

5.2.1	RGB color space	5-24
5.2.2	YUV Color Space	5-36
5.2.3	Digital Television and Photography	5-42
5.2.4	CMY and CMYK Color Models	5-45
5.2.5	HSI Color Model	5-51
5.3	Pseudo Color Image Processing	5-67
5.3.1	Intensity Slicing	5-67
5.3.2	Gray Level to Color Transformation	5-74
5.4	Full-Color Image Processing	5-83
5.4.1	Color Transformations	5-86
5.4.2	Color Complements	5-94
5.4.3	Color Slicing	5-97
5.4.4	Tone and Color Corrections	5-102
5.4.5	Histogram Processing	5-111
5.5	Smoothing and Sharpening	5-113
5.5.1	Color Image Smoothing	5-113
5.5.2	Color Image Sharpening	5-119

5.6	Color Segmentation	5-122
5.6.1	Segmentation in HSI Color Space	5-122
5.6.2	Segmentation in RGB Vector Space	5-126
5.6.3	Color Edge Detection	5-132
5.7	Color Image Compression	5-141

Appendix	A-1
-----------------	------------

A.1	Noise	A-1
A.2	Theorems of 2D Fourier Transform	A-9
A.3	Excursion: Basics of Complex Numbers	A-14

Bibliography	B-1
---------------------	------------

Index	C-1
--------------	------------

1 Introduction

Contents

1.1 Motivation	1-1
1.2 Digital Images	1-14
1.2.1 Basics of Computer Graphics	1-14
1.2.2 Raster versus Vector Graphics	1-18
1.2.3 Graphics File Formats	1-20
GIF	1-21
JPG	1-22
PNG	1-23
PSD	1-24
RAW	1-25
TIFF	1-26
SVG	1-28
1.3 DIP: Application Examples	1-29
1.3.1 Gamma-Ray Imaging	1-32
1.3.2 X-ray Imaging	1-34

1.3.3	Imaging in the Ultraviolet Band	1-36
1.3.4	Imaging in the Visible and Infrared Band	1-38
1.3.5	Imaging in the Terahertz Band	1-47
1.3.6	Imaging in the Microwave Band	1-50
1.3.7	Imaging in the Radio Band	1-52
1.3.8	Other Imaging Modalities	1-54
1.4	Human Vision	1-61
1.4.1	Deception of the Human Eye?	1-63
1.4.2	Gestalt Principles of Visual Organization	1-68
1.4.3	Human Object Perception	1-78
1.4.4	Computer vision vs. biological vision	1-90
1.5	History of Computer Vision	1-91
1.6	Fundamental Steps in DIP	1-92

This course deals with a subset of cognitive systems. It is a comprehensive introduction to digital image processing (DIP). Other aspects of cognitive systems, as e. g. psychophysics, neurosciences, robotics, artificial intelligence, are covered by other courses.

1.1 Motivation

Why study **image processing**, **image analysis** and **machine vision**?

“One picture is worth more than ten thousand words.”

(Anonymous)

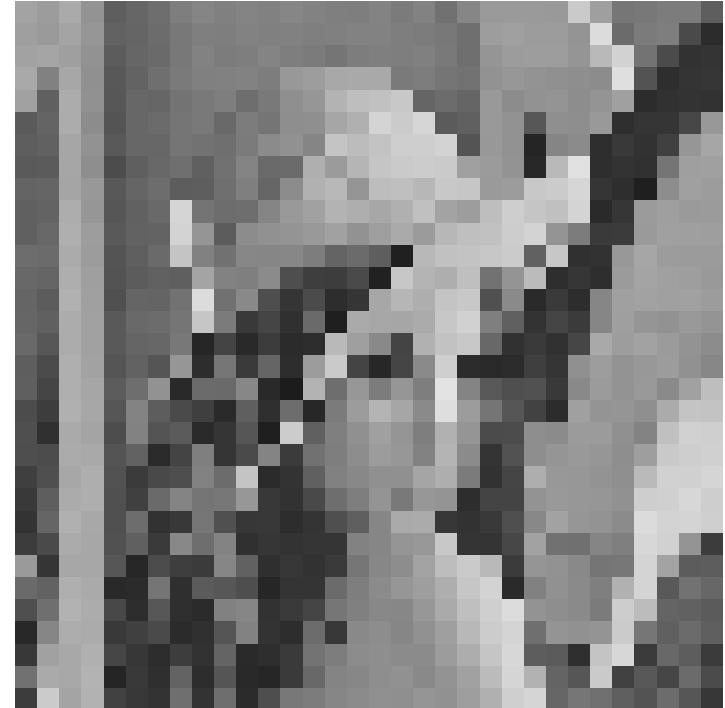
Aspects:

- Subfield of Computer Science and Electrical Engineering
- History of more than 40 years
- Rich methodology
- Interesting interdisciplinary ties
- Exciting insights into human vision
- Important applications
- Important information modality in the information age

What is **image processing**?

Aspects:

- Transforming images as a whole
- “Bildverarbeitung” in a narrow sense
- E. g. change of resolution, high pass filtering, noise removal

 \Rightarrow 

512 x 512 pixel

32 x 32 pixel

Fig. 1.1: Change of resolution.

What is **image analysis**?

Aspects:

- Computing image components and their properties
- “Bildanalyse”
- E. g. edge detection, object localization, motion tracking

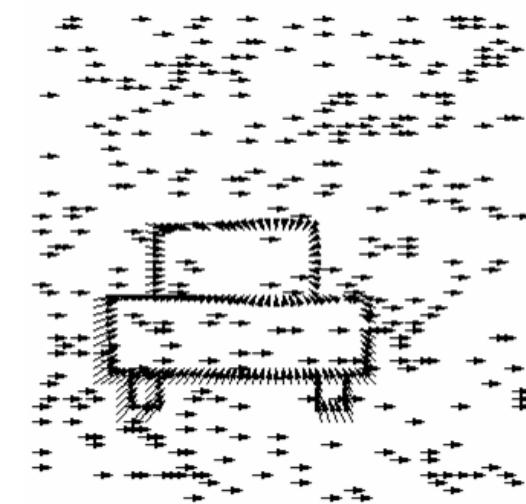
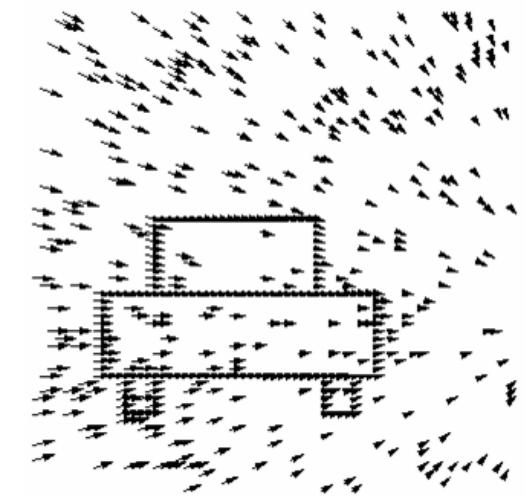
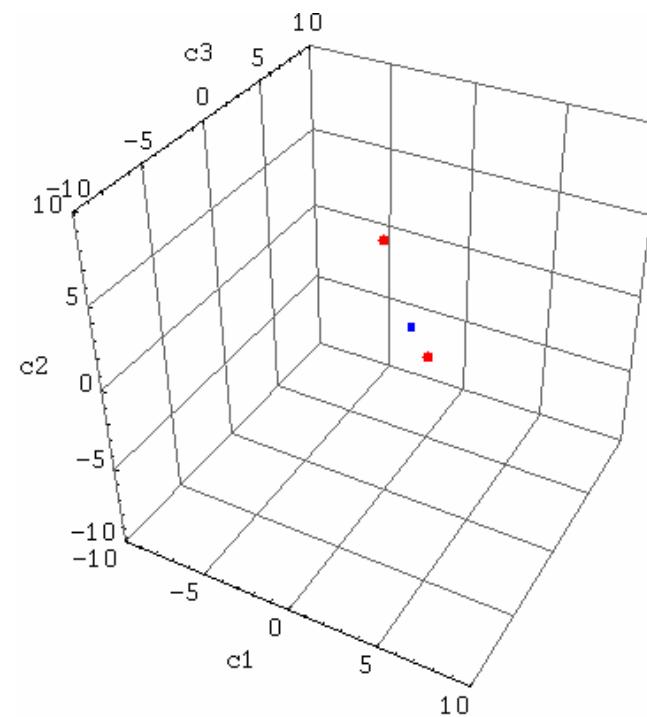
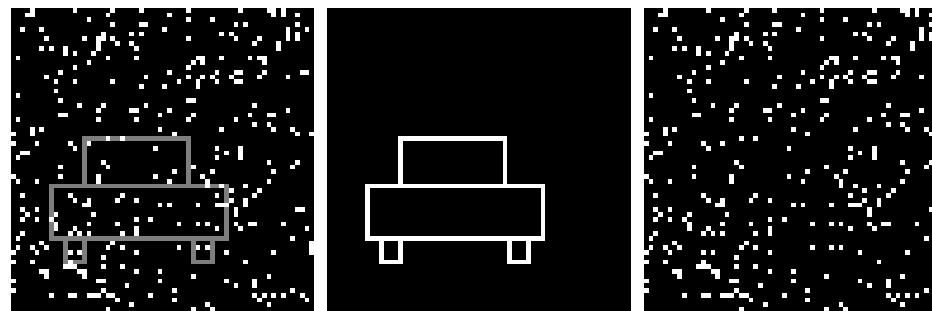


Fig. 1.2: Motion segmentation.

What is **image understanding**?

Aspects:

- Computing the semantics of images
- “Bildverstehen”
- E. g. object recognition, scene interpretation, vision and acting



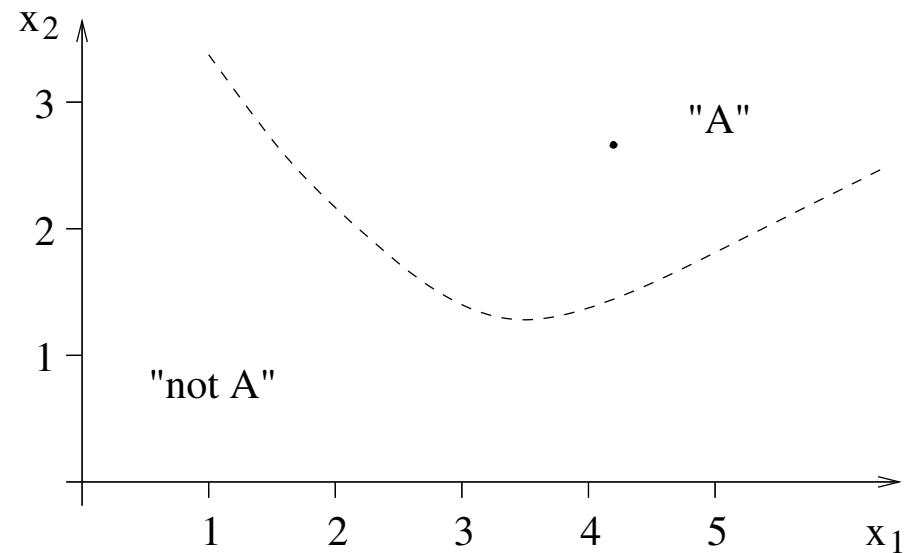
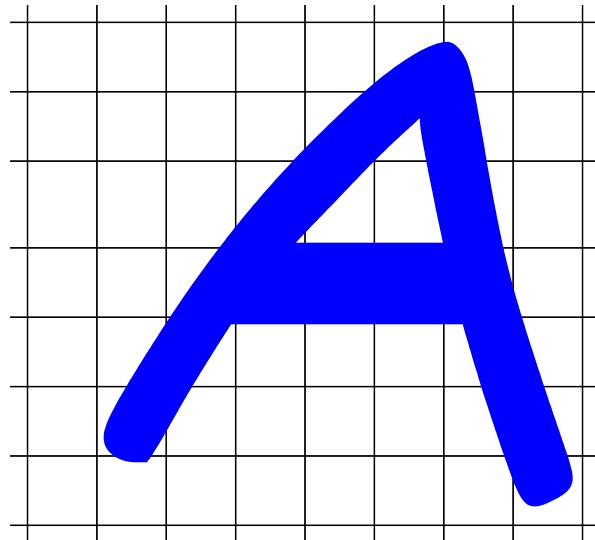
“A white taxi turns right into a narrow road.”

Fig. 1.3: Interpretation of an image sequence. (From [Nag]).

What is **pattern recognition**?

Aspects:

- In the narrow sense: object classification based on feature vectors
- In the wide sense: similar to image analysis
- “Mustererkennung”
- E. g. character recognition, crop classification, quality control



$$x_1 = 4, 2$$

$$x_2 = 2, 7$$

$$\vec{\mathbf{x}} = \begin{bmatrix} 4, 2 \\ 2, 7 \end{bmatrix}$$

“The unknown object is an A.”

Fig. 1.4: Example “character recognition”.

What is **computer vision**?

Aspects:

- Used as synonym for “machine vision” (“Maschinensehen”) or “high-level vision”
- Here: General term for the whole field, including image processing, image analysis, and image understanding
- Digital image processing in the wide sense

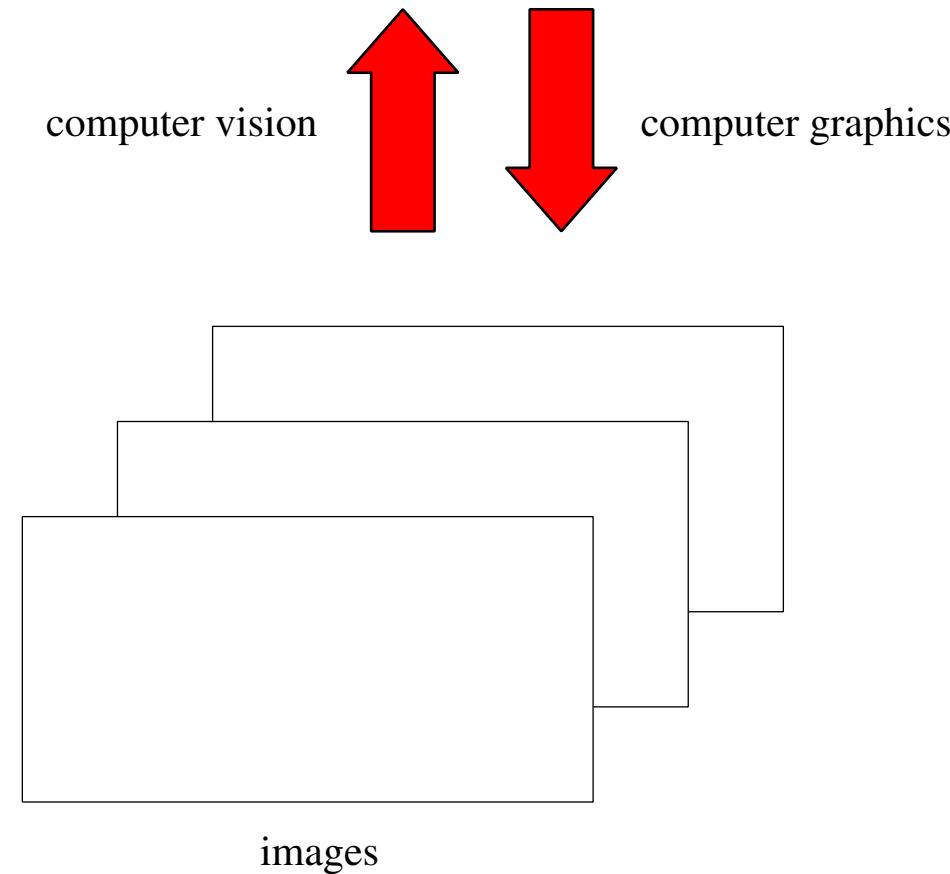


Fig. 1.5: Computer graphics vs. computer vision.

1.2 Digital Images

1.2.1 Basics of Computer Graphics

Two different formats for digital images:

- **Raster graphics image, or bitmap :**

Dot matrix data structure representing a grid of pixels (see fig. 1.6).

- **Vector graphics image:**

Image consists of geometrical primitives such as points, lines, curves, shapes, polygon(s) or general curves (**splines**) basing on mathematical expressions (see example 1.1).

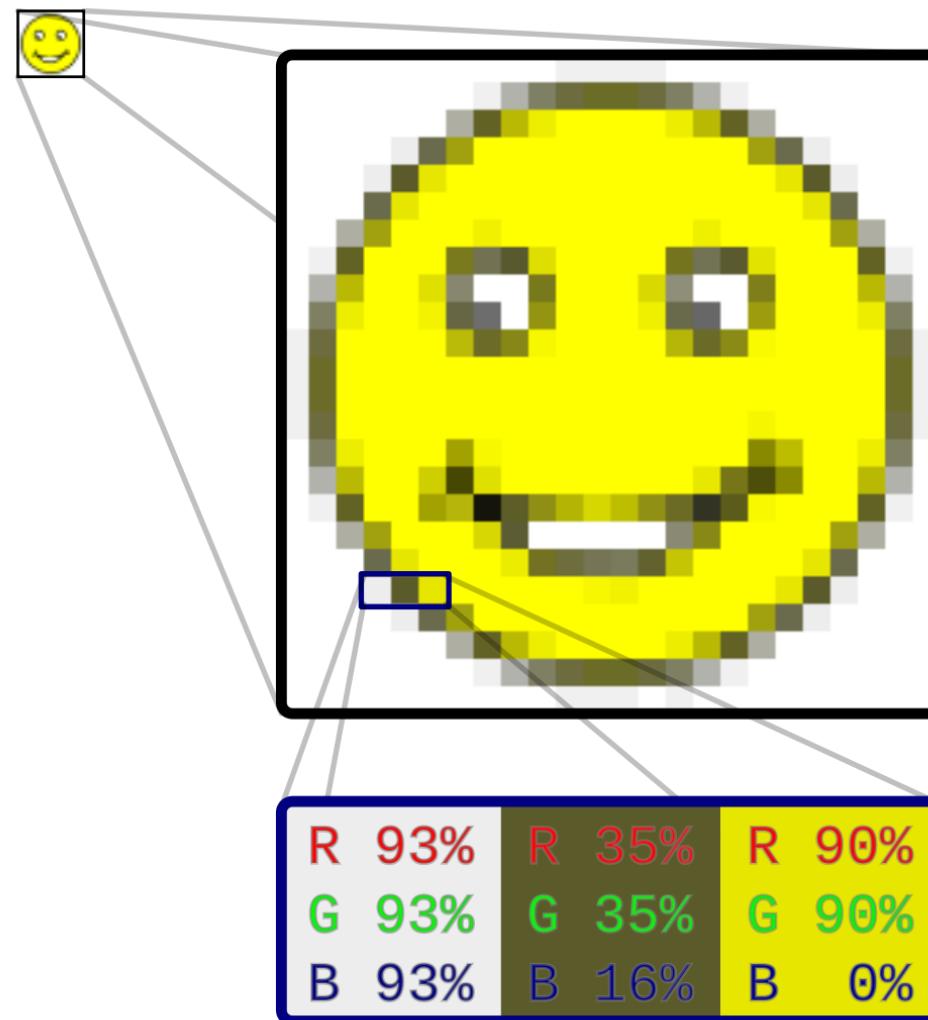


Fig. 1.6: Raster graphics image. (From [Gri]).

Example 1.1 Vector graphics: Circle

Information needed:

- indication that what is to be drawn is a circle,
- radius,
- location of the center point of the circle,
- stroke line style and color (possibly transparent) and
- fill style and color (possibly transparent).

Corresponding SVG code fragment¹:

```
<html>
<body>

<h1>Example: SVG – Circle Description</h1>

<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <circle cx="100" cy="50" r="40" stroke="black"
    stroke-width="2" fill="red" />
</svg>

</body>
```



¹ SVG: Scalable Vector Graphics.

1.2.2 Raster versus Vector Graphics

Feature	Raster Graphics	Vector Graphics
Details	Complex, photographic images	Vector graphic with small file size lacks detail compared with real world photo
Zoom	Proportional zoom	Zoom: infinitely possible; contours remain smooth; polygons remain polygons
File size	Big file sizes: Doubling of image size results in quadrupling of data volume	Small file sizes due to minimal amount of information (see fig. 1.7)
Editing	Individual pixels	Parameters of objects are stored and can be later modified
Shadows	Realistic shades	Shaded areas have to be constructed manually.
Application	Universal	Program dependent

Table 1.1: Comparison of formats for images.



Resolution:	Vector Size:	Bitmap Size:
250 x 78	36KB	57KB
800 x 248	36KB	581KB
1500 x 466	36KB	2000KB (2 MB)

Fig. 1.7: Comparison of file sizes. (From [Agr])

1.2.3 Graphics File Formats

Some criteria for the evaluation of graphics file formats:

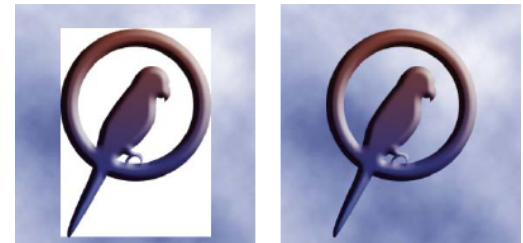
Criterion	Description	Visualization
Compression	Lossy / lossless	
Meta data	Camera settings and scene information recorded by camera into image file (e. g. shutter speed, date and time when a photo was taken, focal length, exposure compensation, etc.)	Example: EXIF (Exchangeable Image File) data.
Transparency	Transparency allows the use of non-rectangular images	 see [Ado17]

Table 1.2: Properties of file formats.

Survey on the mostly used graphics file formats in alphabetical order.

GIF

Description	Pros	Cons	Applications
Abbreviation for <i>Graphics Interchange Format</i>	Transparency	No stepwise transparency	Logos
Introduced by CompuServe in 1987 (see [w3.87])		Proprietary format (Only open since 2006)	Lettering
Compressed bitmap file	Good lossless compression		Simple graphics
256 colors available		Restricted color set	Web
Filename extension: .gif	Animations		Animations

Table 1.3: GIF format.

JPG

Description	Pros	Cons	Applications
Abbreviation for <i>Joint Photographic Experts Group</i>	16,7 million colors	Compression is lossy	Photos
Introduced by JPG committee in 1992	Excellent ratio of file size to image quality	Storing results in slight quality losses	Colorful graphics
Most used format for web applications	Widely used	Quality might not be sufficient for high-end printing	Ideal for mailing and web
Well established for digital cameras	For web applications	Artefacts after high compression	
Compressed bitmap file	Stores EXIF data		
Filename extension: .jpg	CMYK		

Table 1.4: JPG format.

PNG

Description	Pros	Cons	Applications
Abbreviation for <i>Portable Network Graphics</i> ²	Color depth up to 16 bit per channel	Limited use for web-pages	Photos
Introduced in 1995 after restrictions of GIF use due to patent	Transparency information in alpha channel and each color channel	No animations	Graphics with (semi-) transparent regions
Compressed bitmap file	Excellent lossless compression		Gradients from color to transparency
Filename extension: .png	Storing of meta data possible		

Table 1.5: PNG format.

² Inofficially: PNG's Not GIF.

PSD

Description	Pros	Cons	Applications
Developed in 1988 Photoshop 1.0 was released in 1990 for Macintosh exclusively	Lossless compression File includes layers for most imaging options available in Photoshop (e. g. layers with masks, color spaces, ICC profiles, CMYK mode, transparency, text, alpha channels, etc.)	File size limitation ³ Can't be used for web applications	For use in Photoshop Industry standard for professionals
Filename extension: <i>.psd</i>			

Table 1.6: Native Adobe Photoshop format.

³ PSD file: maximum height and width of 30,000 pixels, and a length limit of 2 Gigabytes.

RAW

Description	Pros	Cons	Applications
Unprocessed and un-compressed data from a camera (digital negative)	High-end image processing	Enormous file sizes	For professional enhancement of images
Filename extension depends on camera type ⁴ (e. g. <i>.raf</i> ⁵ or <i>.crw</i> ⁶)	Complete “freedom” for white balance, contrast, etc.	Delays when storing files	
	Different color depths (up to 14 bit)	Can’t be used for web applications	
	EXIF data		

Table 1.7: RAW format.

⁴ Standard raw image format (ISO 12234-2, TIFF/EP) is not widely accepted. DNG, Adobe’s candidate for a new standard format, has not been adopted by many major camera companies.

⁵ Fujifilm

⁶ Canon

TIFF

Description	Pros	Cons	Applications
Abbreviation for <i>Tagged Image File Format</i>	Lossless and lossy compression	Huge file sizes	Data exchange
Introduced in 1986	16 bit alpha channel	Can't be used for web applications	Scanning
For high resolution images	high resolution		Pre-press
Compressed bitmap file	ICPT text information ⁷		Archives
Filename extension: .tif	Layers		Aerial images, maps (GeoTIFF ⁸)
	CMYK		

Table 1.8: TIFF format.

⁷ International Press Telecommunications Council developed the Information Interchange Model (IIM) file structure and set of metadata attributes that can be applied to text, images and other media types to expedite the international exchange of news among newspapers and news agencies.

⁸ Public domain metadata standard which allows georeferencing information to be embedded within a TIFF file.

SVG

Description	Pros	Cons	Applications
Abbreviation for <i>Scalable Vector Graphics</i>	Files can be searched, indexed, scripted, and, compressed	Fragmented community	Prints with high quality at any resolution
Under development by the World Wide Web Consortium (W3C) since 1999	Support by web browsers	No support of z-indices	Integration with other W3C standards such as the DOM and XSL
XML-based file format for two-dimensional vector graphics	Allows three types of graphic objects: vector graphics, raster graphics, and text		Web
Filename extension: .svg	Animations		
	Meta data		

Table 1.9: SVG format.

1.3 DIP: Application Examples

The areas of application of DIP are very manyfold.

Here: Approach to capture the breadth of the field by categorizing images according to their source (e. g., visual, X-ray, and so on).

Principal energy source of images: **Electro magnetic energy spectrum** (see fig. 1.8)

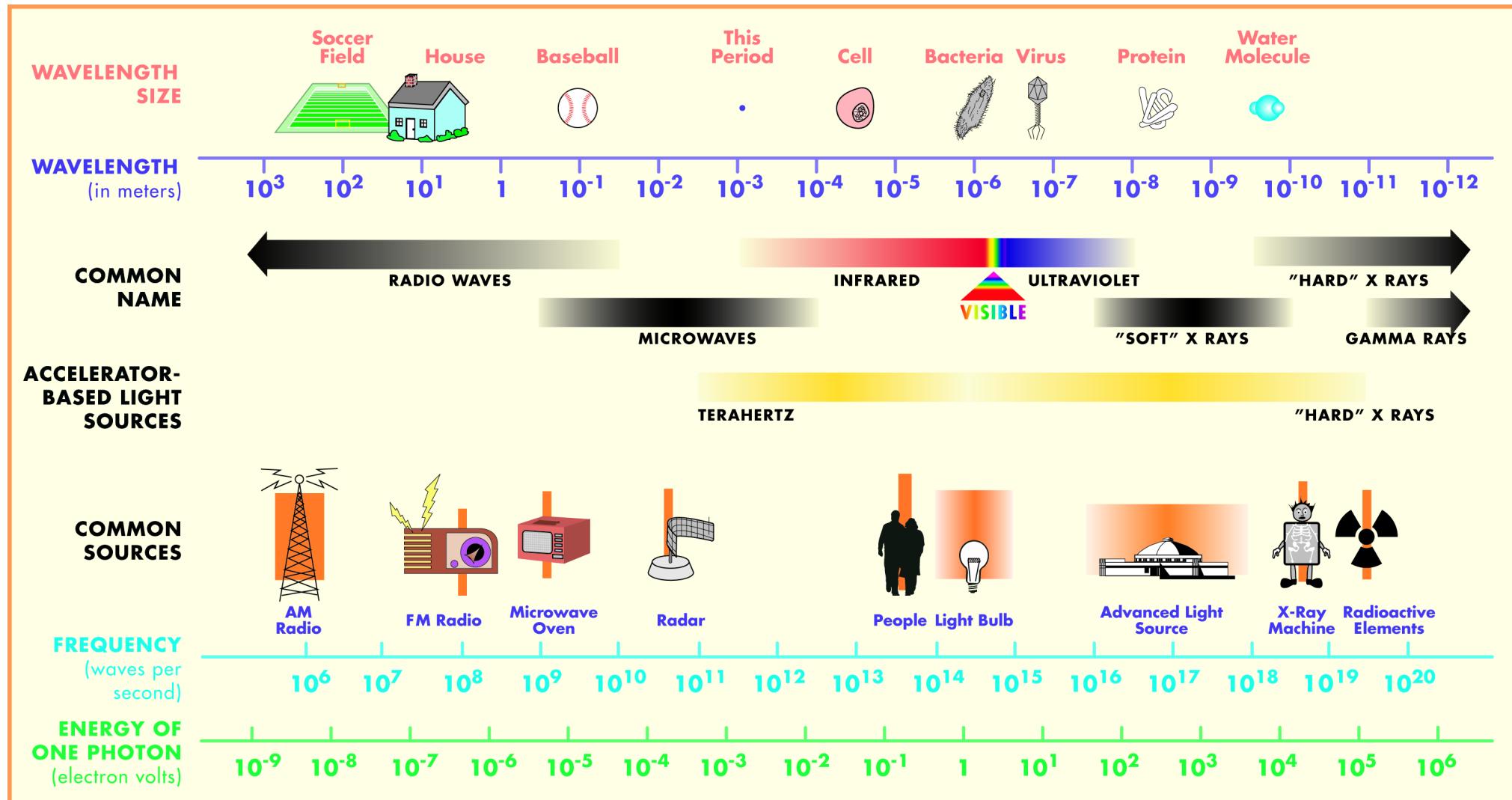


Fig. 1.8: Electromagnetic spectrum. [Courtesy of the Advanced Light Source, Lawrence Berkeley National Laboratory, 2019]

Conceptualization of electromagnetic waves:

- Sinusoidal waves of varying wavelengths
- Stream of massless particles, travelling in a wavelike pattern and moving at the speed of light
- Each particle contains a certain amount (bundle) of energy.
- Each bundle of energy is called a **photon**.

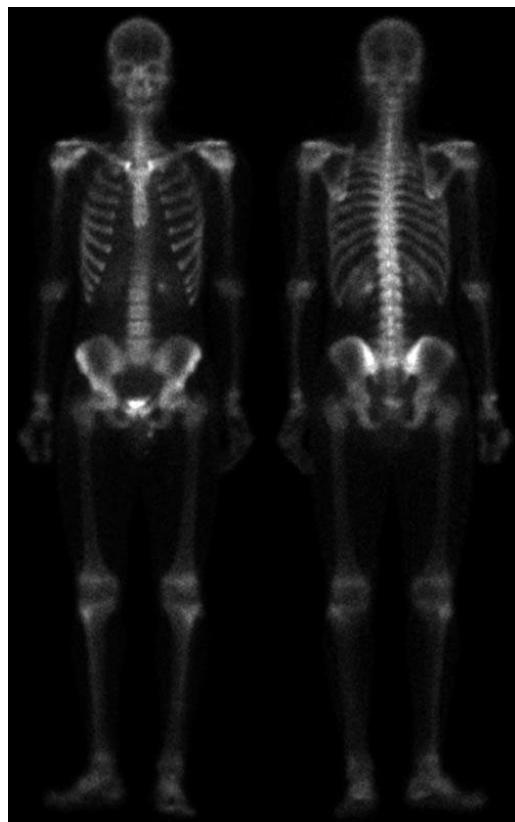
1.3.1 Gamma-Ray Imaging

Gamma rays are mostly used in

- nuclear medicine and
- astronomy

Special modality: **positron emission tomography (PET)**

- similar to X-ray tomography
- a patient gets a radioactive isotope that emits positrons as it decays (see fig. 1.9b)



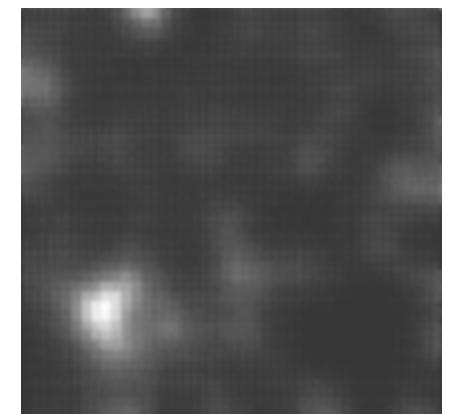
(a)



(b)



(c)



(d)

Fig. 1.9: Examples of gamma-ray imaging. (a) Bone scan. (b) PET image. (c) Cygnus Loop. (d) Gamma radiation from a reactor valve. (From [GW18])

1.3.2 X-ray Imaging

X-rays are widely used for imaging in a huge variety of fields, e. g.

- medical diagnostics
- industrial imaging
- astronomy

Special modality: **computerized axial tomography (CAT or computerized tomography (CT))**



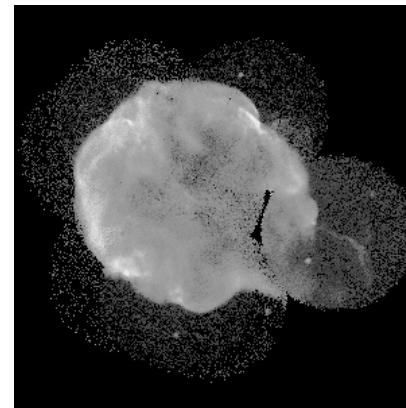
(a)



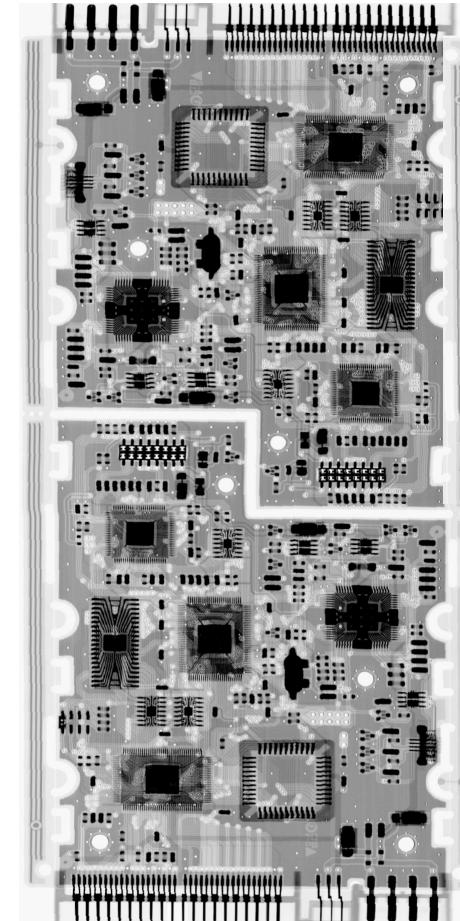
(c)



(b)



(d)



(e)

Fig. 1.10: Examples of X-ray imaging. (a) Chest X-ray. (b) Aortic angiogram. (c) Head CT. (d) Cygnus Loop (same constellation as in Fig. 1.9c, but this time imaged in the X-ray band). (e) Circuit boards. (From [GW18])

1.3.3 Imaging in the Ultraviolet Band

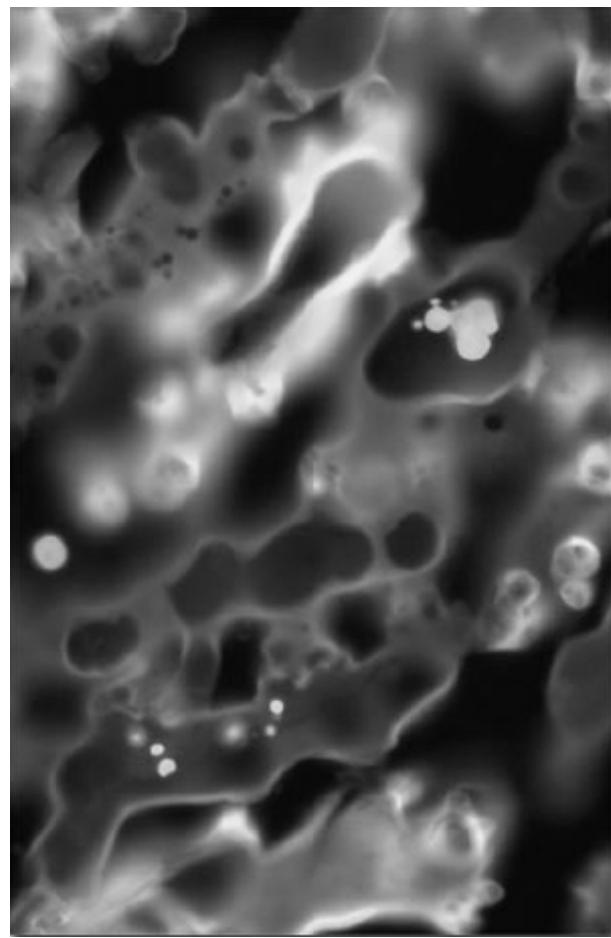
Wide range of applications, e. g.

- lithography,
- industrial inspection,
- microscopy,
- lasers,
- biological imaging, and
- astronomical observations

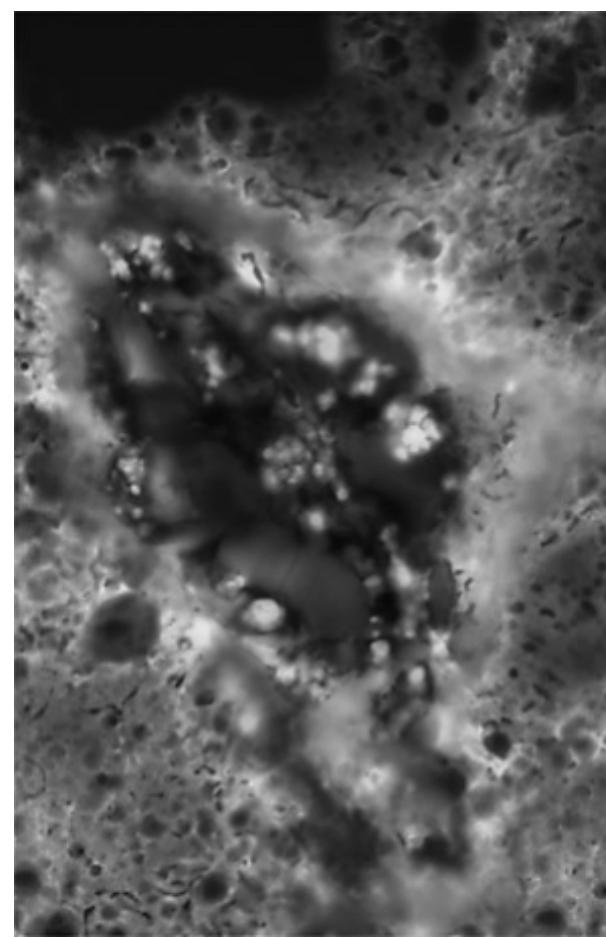
Special modality: **fluorescence⁹ microscopy**

- Mineral fluorspar fluoresces when ultraviolet light is directed upon it.
- Ultraviolet light itself is no visible, but emitted light is (red).
- Method for studying materials in their natural form (primary fluorescence) or when treated with chemicals capable of fluorescing (secondary fluorescence).

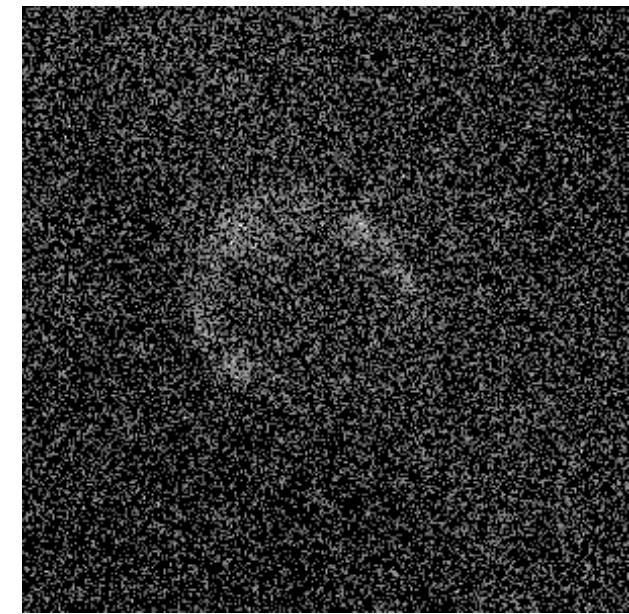
⁹ Optical phenomenon in cold bodies → Molecular absorption of a photon triggers the emission of another photon with a longer wavelength.



(a)



(b)



(c)

Fig. 1.11: Examples of ultraviolet imaging. (a) Normal corn. (b) Smut corn. (c) Cygnus Loop again. (From [GW18])

1.3.4 Imaging in the Visible and Infrared Band

Most familiar band in the electromagnetic spectrum.

Numerous applications.

Here: Consideration of applications in

- microscopy,
- astronomy,
- remote sensing,
- industry, and
- law enforcement

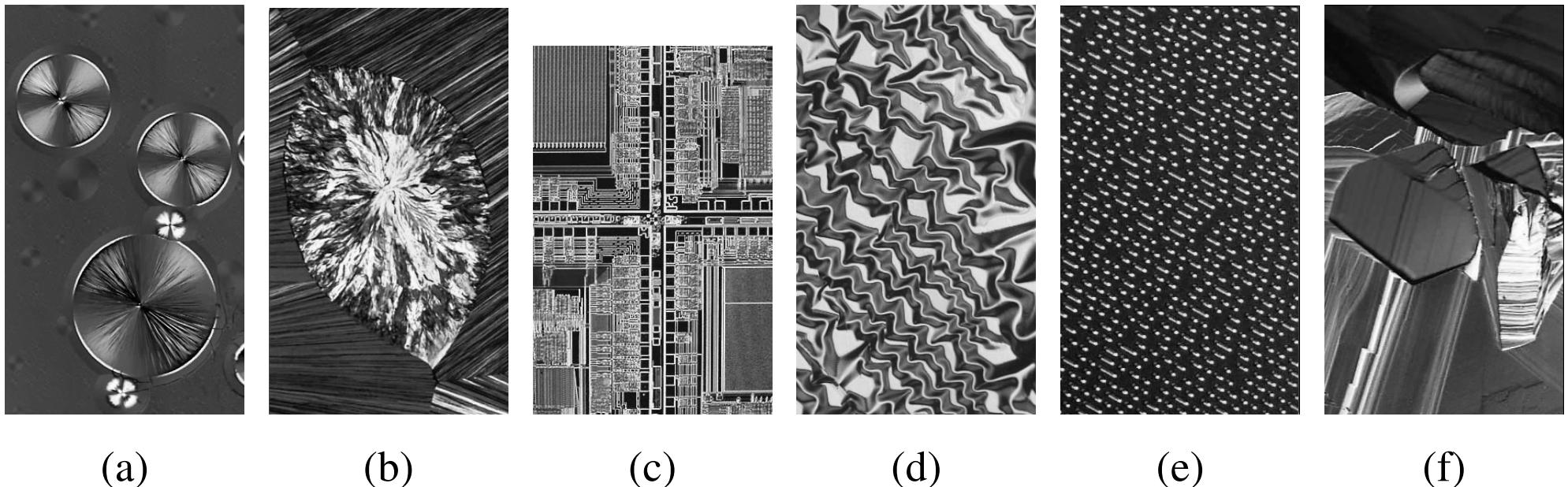


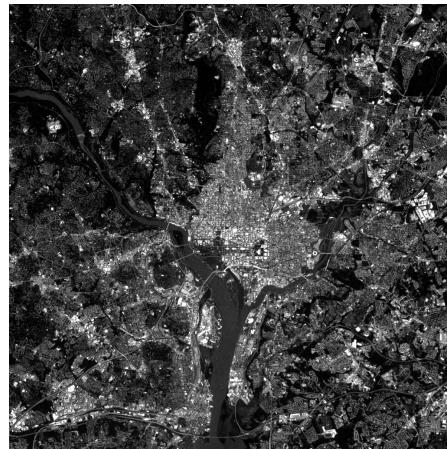
Fig. 1.12: Examples of light microscopy images. (a) Taxol (anti cancer agent), magnified 250x. (b) Cholesterol, 40x. (c) Microprocessor, 60x. (d) Nickel oxide thin film, 600x. (e) Surface of audio CD, 1750x. (f) Organic superconductor, 450x. (From [GW18])

Hint: The infrared band often is used in conjunction with visual imaging.

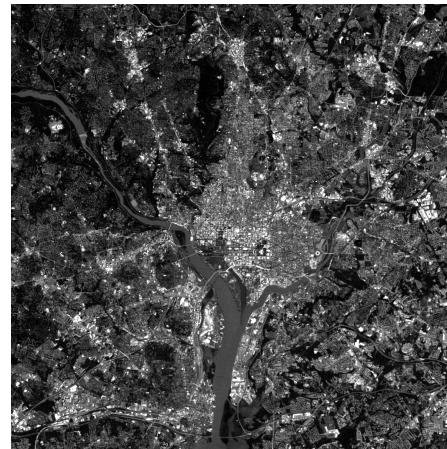
For remote sensing purposes typically several bands are used (compare table 1.10)

Band No.	Name	Wavelength (μ)	Characteristics and Uses
1	Visible blue	0.45-0.52	Maximum water penetration
2	Visible green	0.52-0.60	Good for measuring plant vigor
3	Visible red	0.63-0.69	Vegetation discrimination
4	Near infrared	0.76-0.90	Biomass and shoreline mapping
5	Middle infrared	1.55-1.75	Moisture content of soil and vegetation
6	Thermal infrared	10.4-12.5	Soil moisture: thermal mapping
7	Middle infrared	2.08-2.35	Mineral mapping

Table 1.10: Thematic bands in NASA's LANDSAT satellite. (From [GW18])



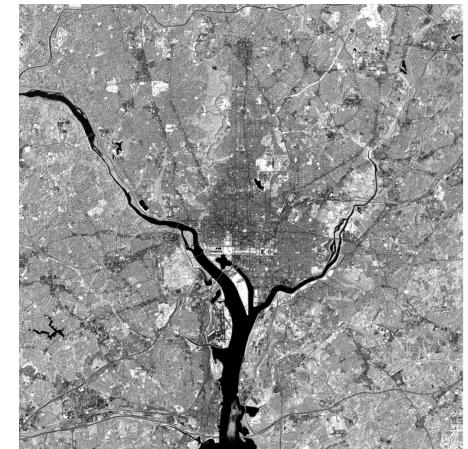
(1)



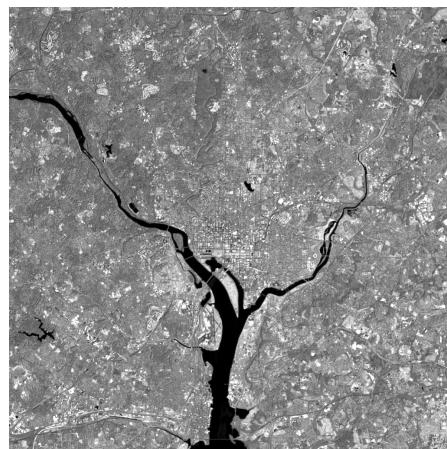
(2)



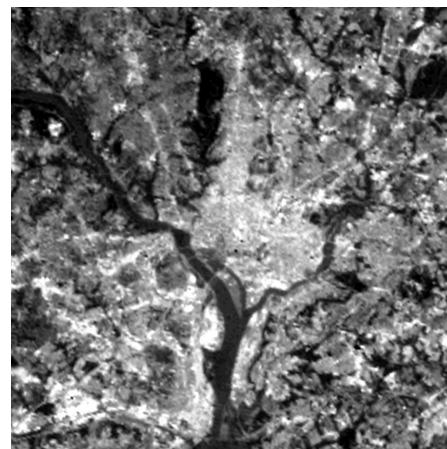
(3)



(4)



(5)



(6)



(7)

Fig. 1.13: LANDSAT satellite images of the Washington, D. C. area. The numbers refer to the thematic bands in Tab. 1.10. (From [GW18])

Weather observation and prediction are also major applications of multispectral images from satellites.

Fig 1.14 shows the infrared and visual image of Europe taken by METEOSAT for Monday 25 April, 2005 of 11:00 resp. 11:30 UTC.



Fig. 1.14: Multispectral images taken by METEOSAT (geostationary meteorological satellite). (From [DWD05]).

Fig. 1.15 shows an application of infrared imaging:
“Nighttime Lights of the World” data set, which provides a global inventory of human settlements.

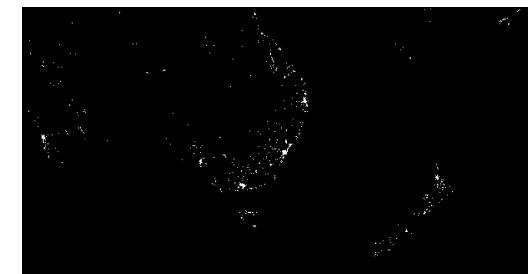
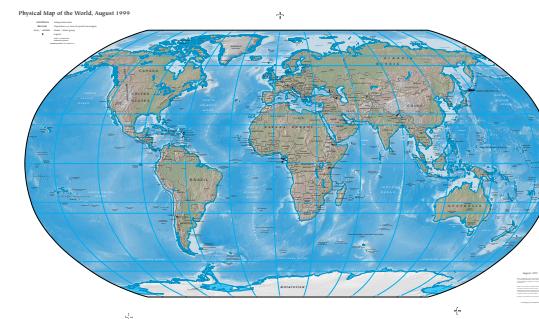
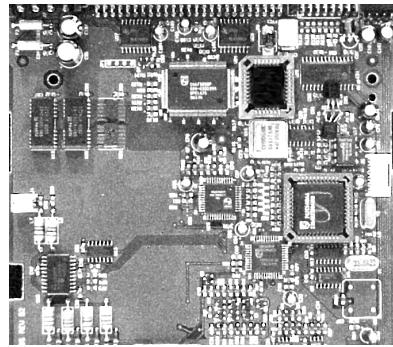
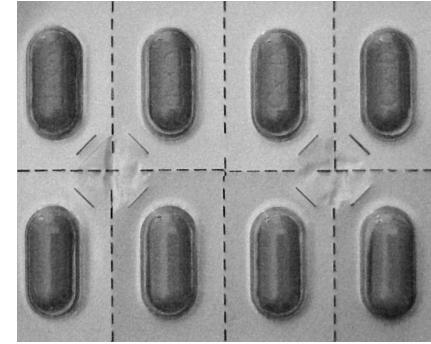


Fig. 1.15: Infrared satellite images of Europe, Africa, Asia, and Australia. (From [GW18])

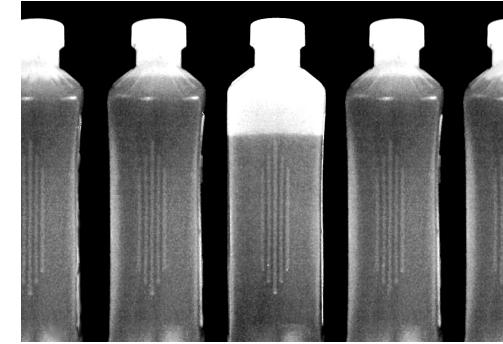
Another major area of imaging in the visual spectrum is an automated visual inspection of manufactured goods. Some examples can be found in fig. 1.16.



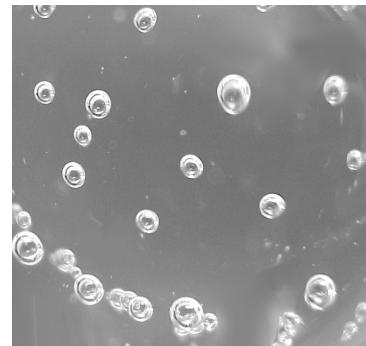
(a)



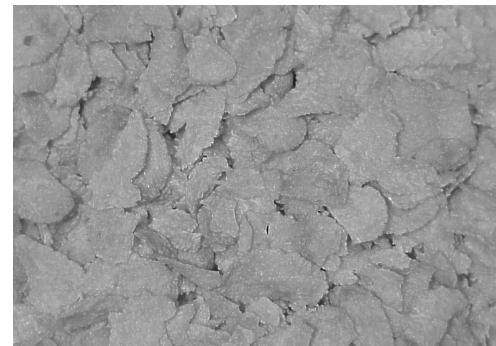
(b)



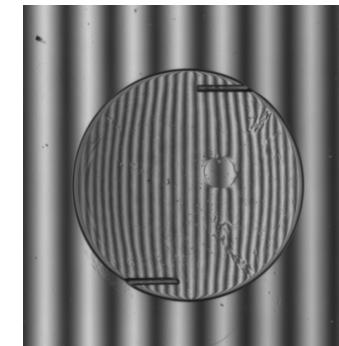
(c)



(d)



(e)



(f)

Fig. 1.16: Some examples of manufactured goods often checked using digital image processing. (a) A circuit board controller. (b) Packaged pills. (c) Bottles. (d) Bubbles in a clear-plastic product. (e) Cereal. (f) Image of intraocular implant. (From [GW18])

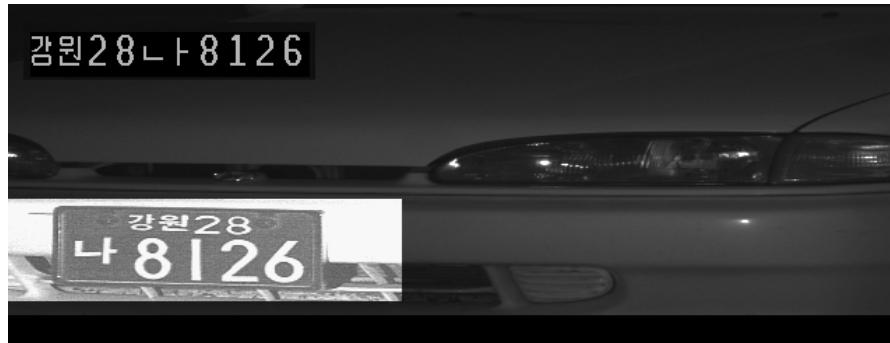
Fig. 1.17 shows image examples for inspection and monitoring purposes.



(a)



(b)



(c)



(d)

Fig. 1.17: Some additional examples of imaging in the visual spectrum. (a) Thumb print. (b) Paper currency. (c)+(d) Automated license plate reading. (From [GW18])

1.3.5 Imaging in the Terahertz Band

Terahertz band: Between IR and microwave band.

Backscatter X-ray:

- Millimetre Wave Scanner.
- X-ray imaging that used scattered x-ray or photons (see fig.1.18).
- Extremely high frequency radio waves are reflected off the body.
- Application: Detection of weapons (see fig. 1.19).

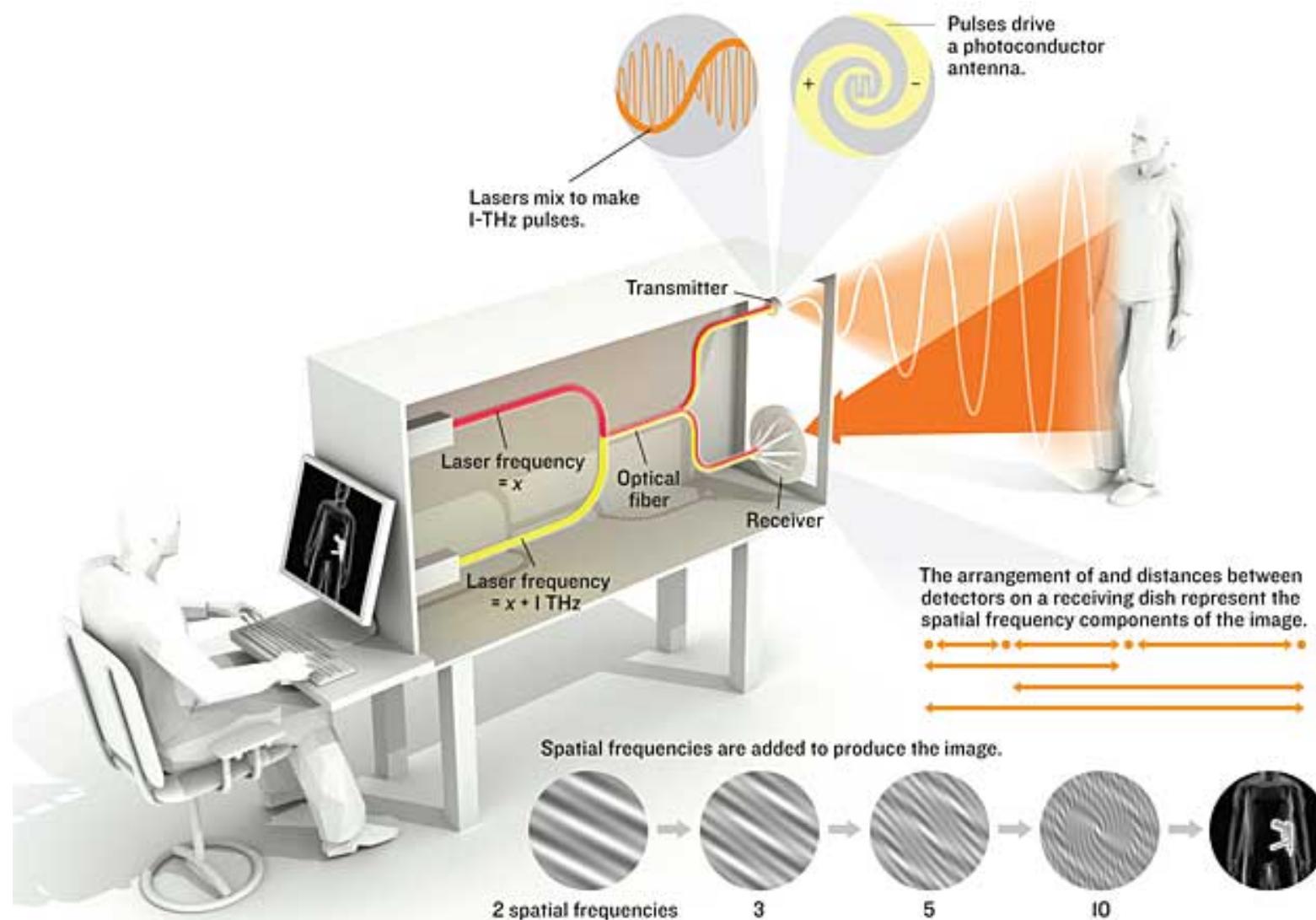


Fig. 1.18: Terahertz scanner. (From [Cen])

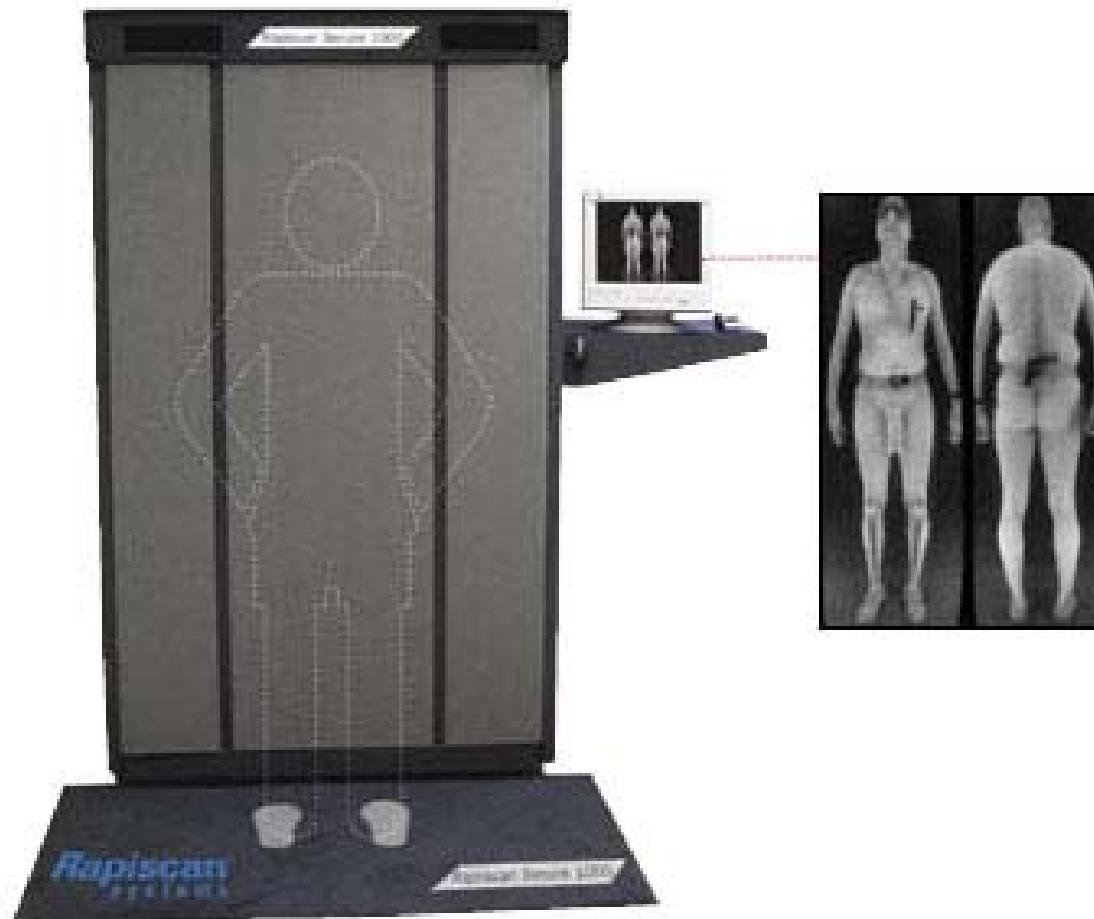


Fig. 1.19: Terahertz imaging in the security domain: Body scanner (From [Cen])

1.3.6 Imaging in the Microwave Band

Radar is the most dominant application of microwave imaging.

Advantages of this band are:

- Data can be collected regardless of weather and ambient lightning conditions.
- Some radars can penetrate vegetation, ice, and extremely dry sand.

An imaging radar works like a flash camera in that it provides its own illumination (microwave pulses) to illuminate an area on the ground and take a snapshot image.

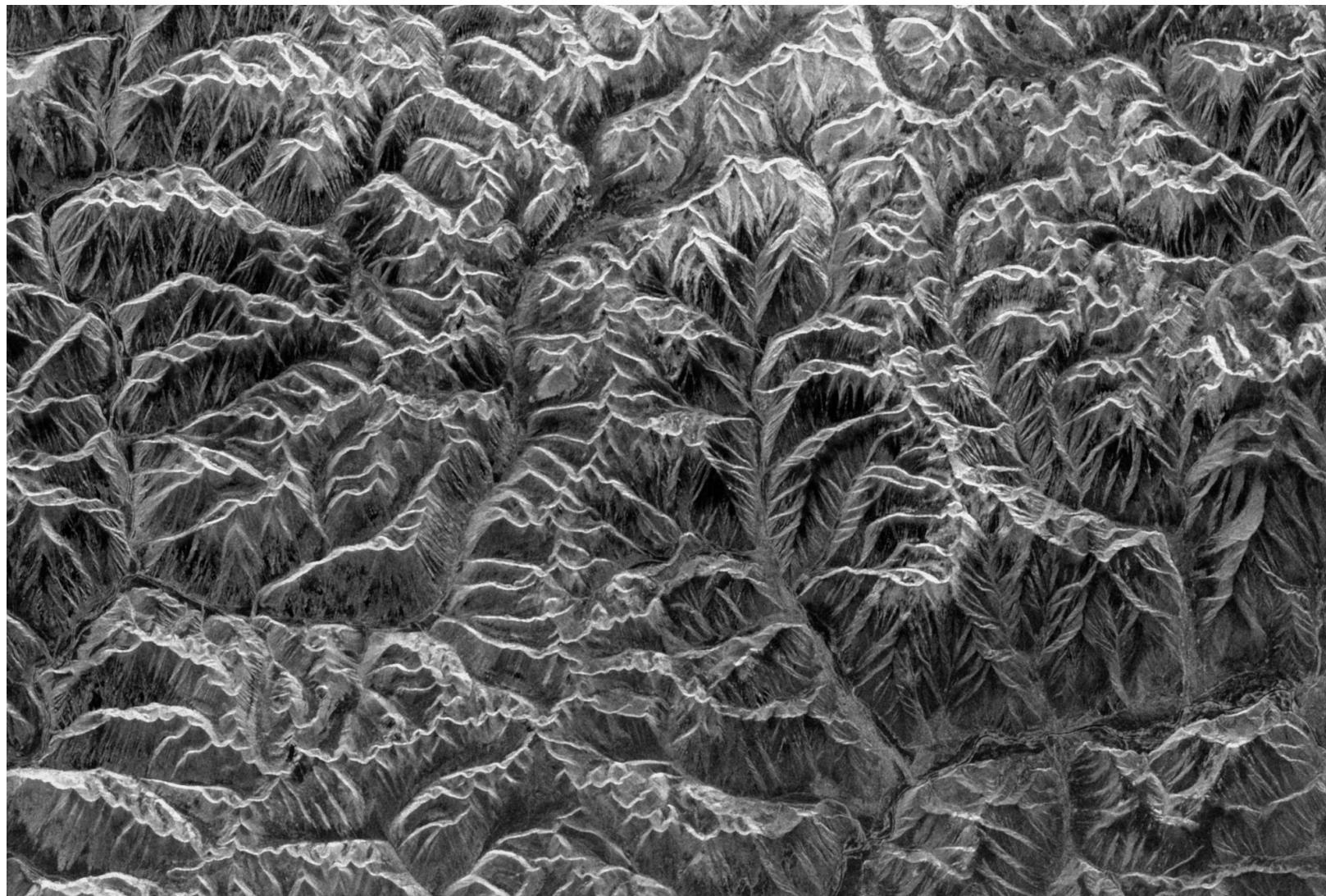


Fig. 1.20: Spaceborne radar image of mountains in southeast Tibet. (From [GW18])

1.3.7 Imaging in the Radio Band

Radio waves are mainly used in medicine and astronomy.

Special modality: **magnetic resonance imaging (MRT)**

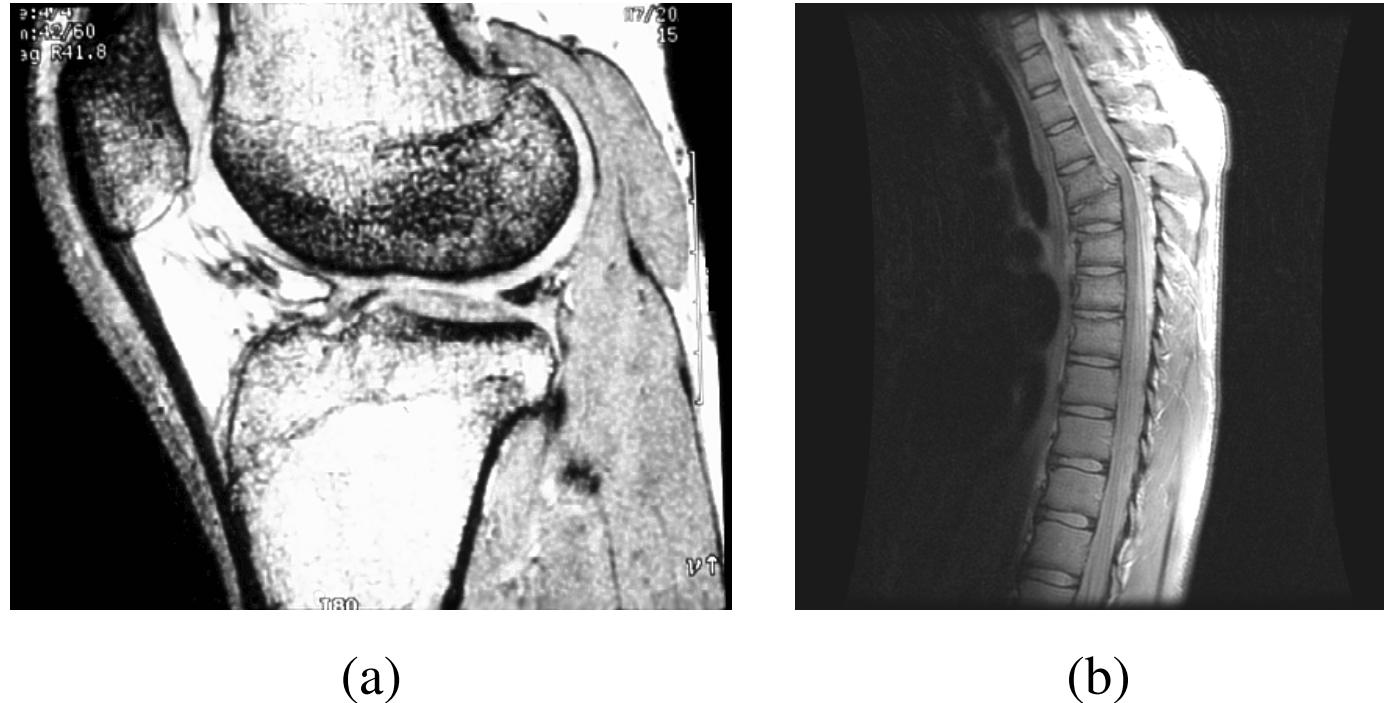


Fig. 1.21: MRI image of a human (a) knee and (b) spine. (From [GW18])

Fig. 1.22 shows an example from astronomy:
the image of the Crab Pulsar in the radio band and in the most bands discussed earlier.

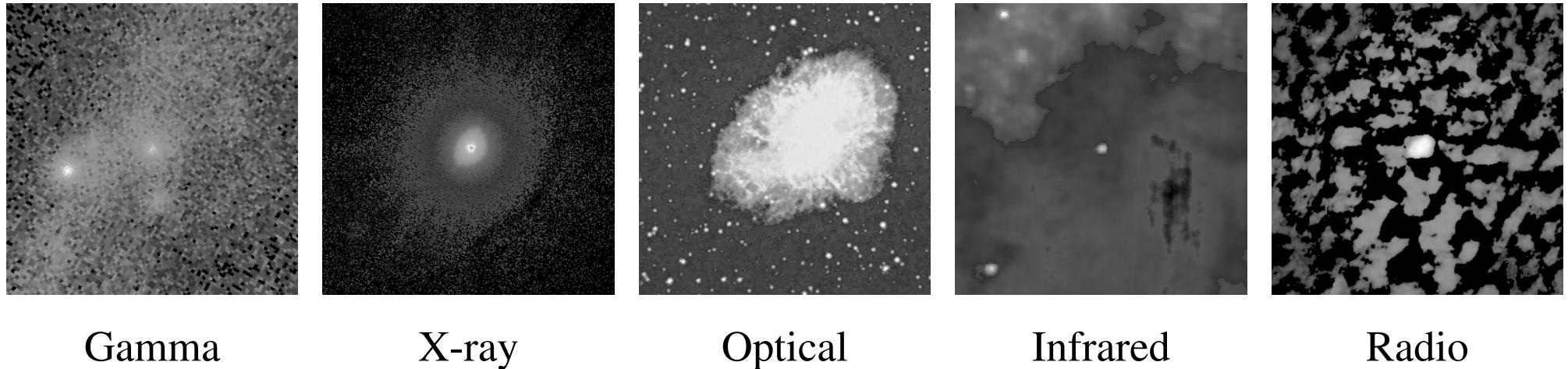


Fig. 1.22: Image of the Crab Pulsar (in the center of the images) covering the electromagnetic spectrum. (From [GW18])

1.3.8 Other Imaging Modalities

Besides imaging in the electromagnetic spectrum, there are various other imaging modalities, e.g.

- acoustic imaging,
- electron microscopy, and
- synthetic imaging

Acoustic imaging is used in

- geological exploration (hundreds of Hertz),
- industry, and
- medicine (ultrasound, millions of Hertz).

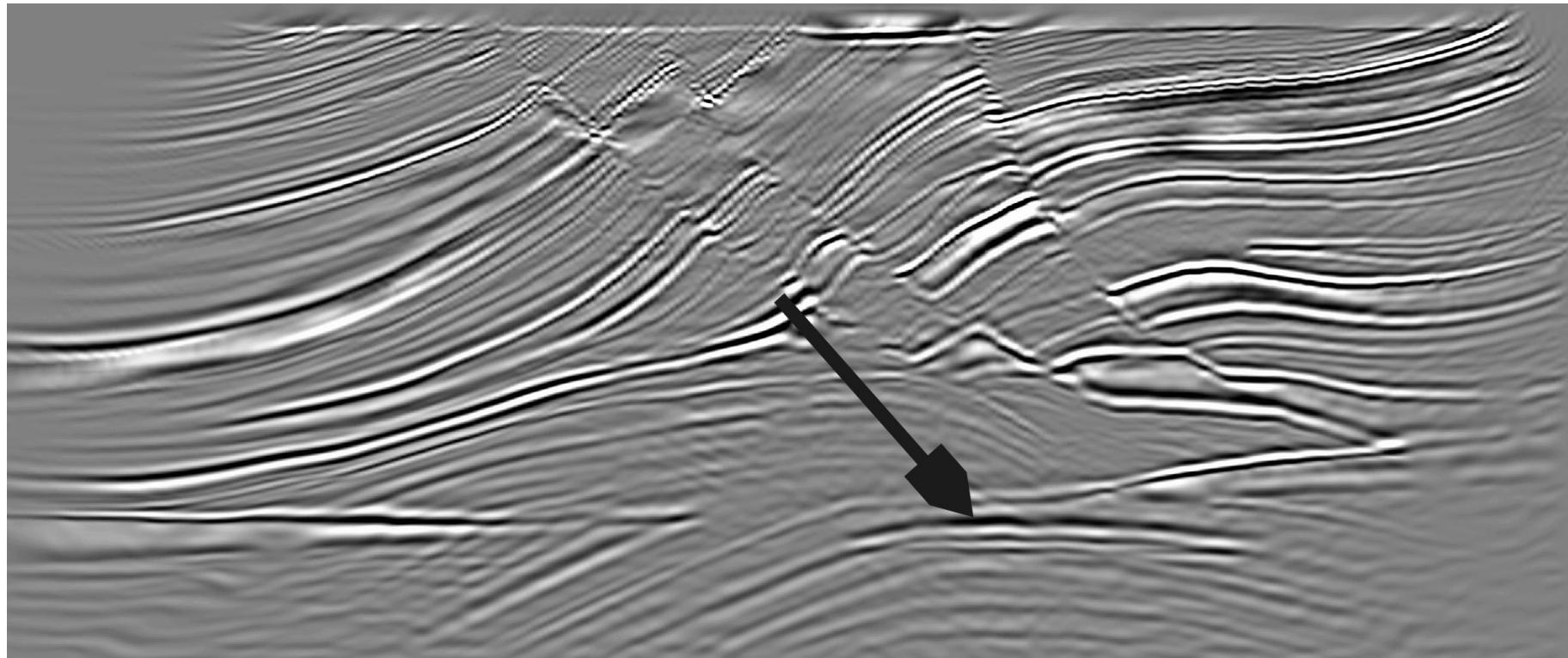


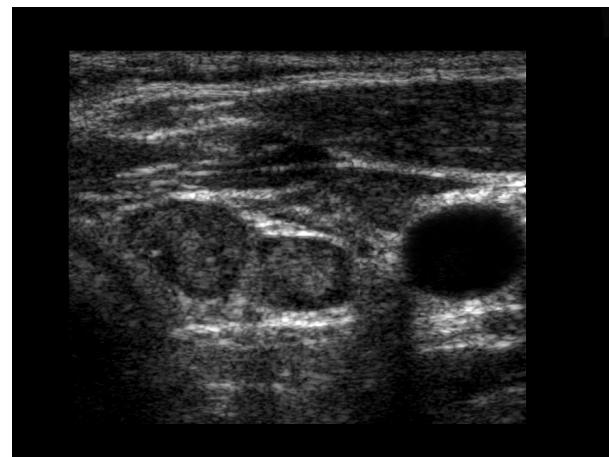
Fig. 1.23: Cross-sectional image of a seismic model. The arrow points to a hydrocarbon (oil and/or gas) trap.
(From [GW18])



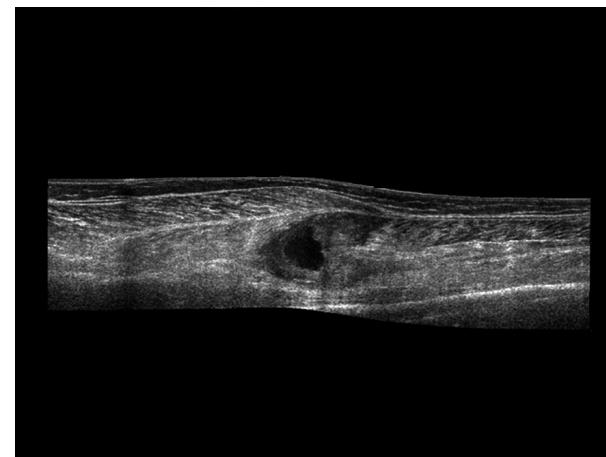
(a)



(b)



(c)



(d)

Fig. 1.24: Examples of ultrasound imaging. (a) Baby. (b) Another view of baby. (c) Thyroids. (d) Muscle layers showing lesion. (From [GW18]).

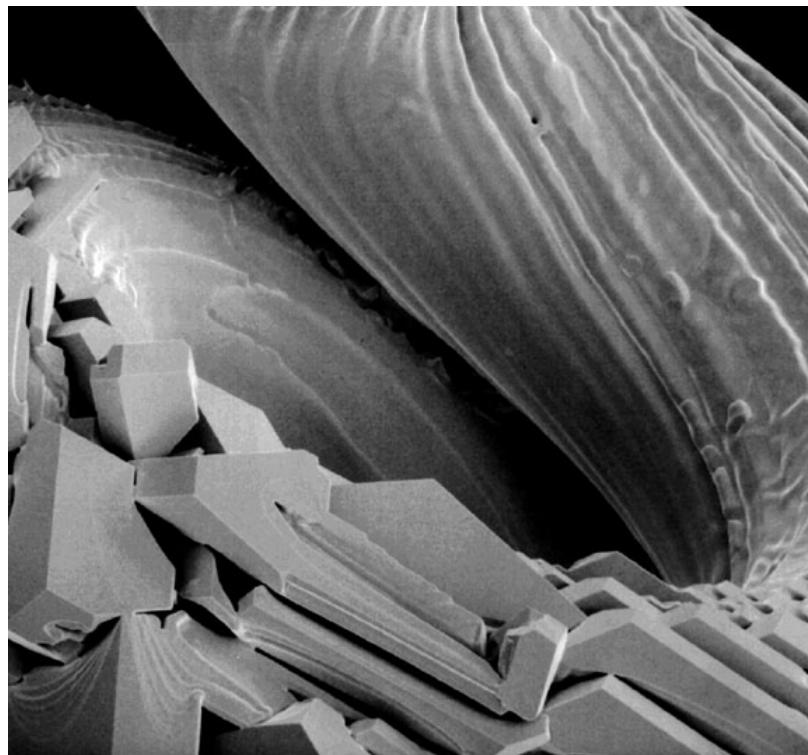
Electron microscopes function as their optical counterparts, except they use a focused beam of electrons instead of light to image a specimen.

Special modality:

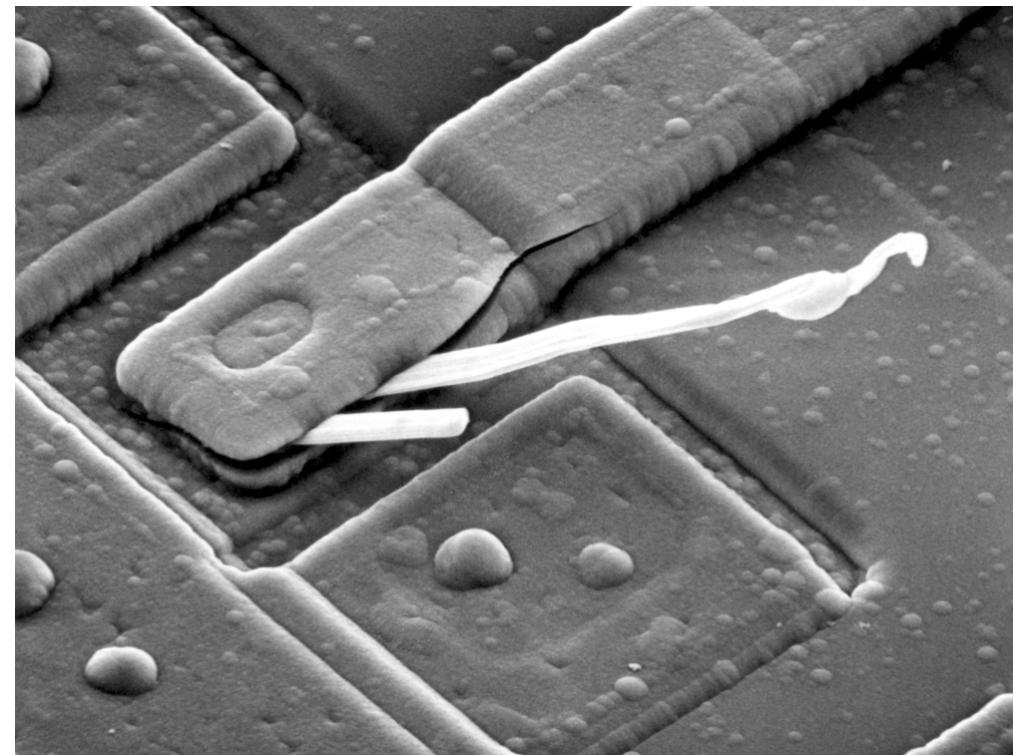
transmission electron microscope (TEM) or scanning electron microscope (SEM)

Magnification of microscopes:

- light microscopes (max. 1.000x),
- electron microscopes (10.000x and more).



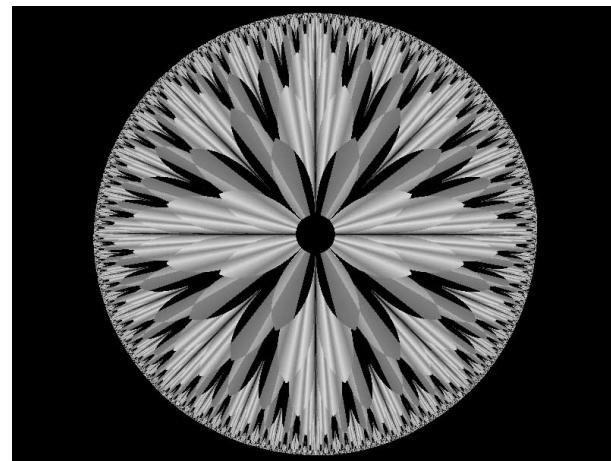
(a)



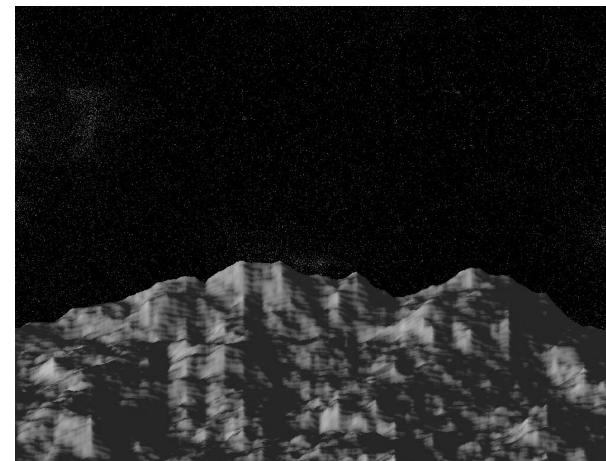
(b)

Fig. 1.25: (a) $250\times$ SEM image of a tungsten filament following thermal failure. (b) $2500\times$ SEM image of damaged integrated circuit. The white fibers are oxides resulting from thermal destruction. (From [GW18])

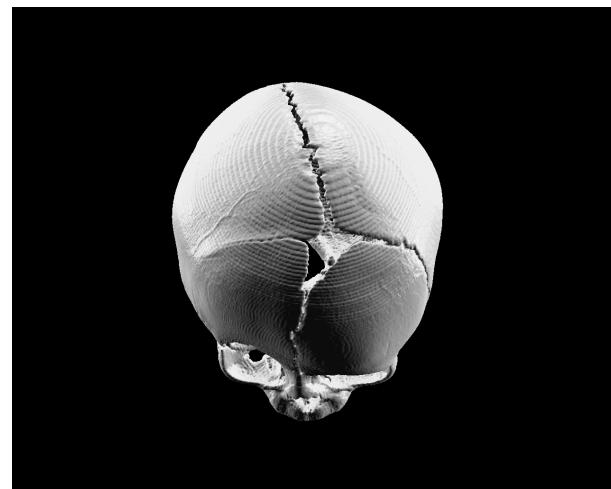
Besides imaging modalities obtained from physical objects computer generated images provide an interesting field.



(a)



(b)



(c)



(d)

Fig. 1.26: (a)+(b) Fractal images. (c)+(d) Images generated from 3-D computer models of the objects shown. (From [GW18]).

1.4 Human Vision

This section sheds some light on how humans perceive their environment.

“No matter how hard we look, we see very little of what we look at.”

(James Elkins)

1.4.1 Deception of the Human Eye?

Fig. 1.27 shows that perceived brightness is not a simple function of intensity.

The seemingly scalloped bands are called **Mach bands** after Ernst Mach, who first described the phenomenon in 1865.

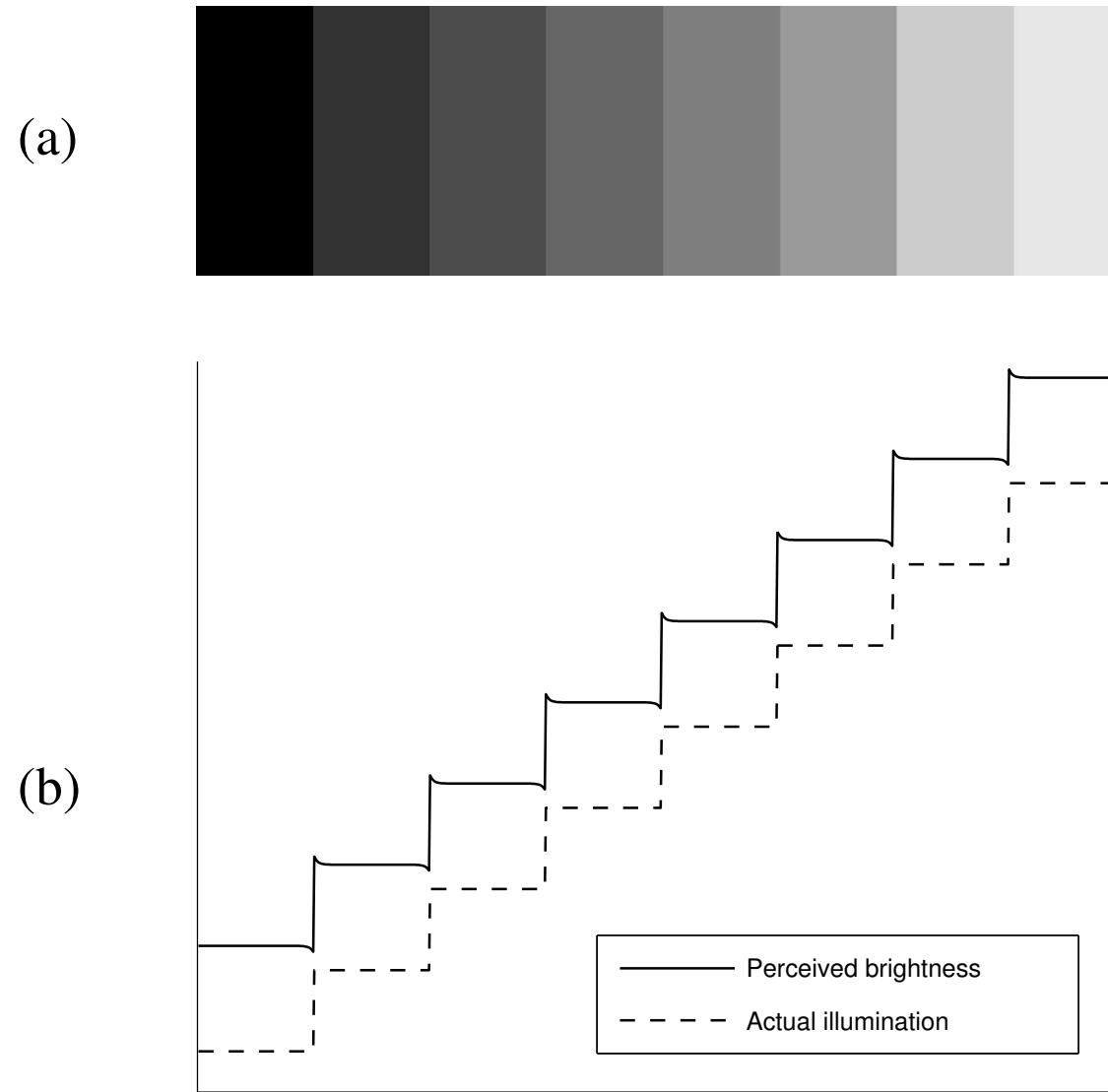


Fig. 1.27: Brightness perception. Hint: The relative vertical positions between the two profiles in (b) have no special significance; they were chosen for clarity.

Another related phenomenon is called **simultaneous contrast** (see fig. 1.28).

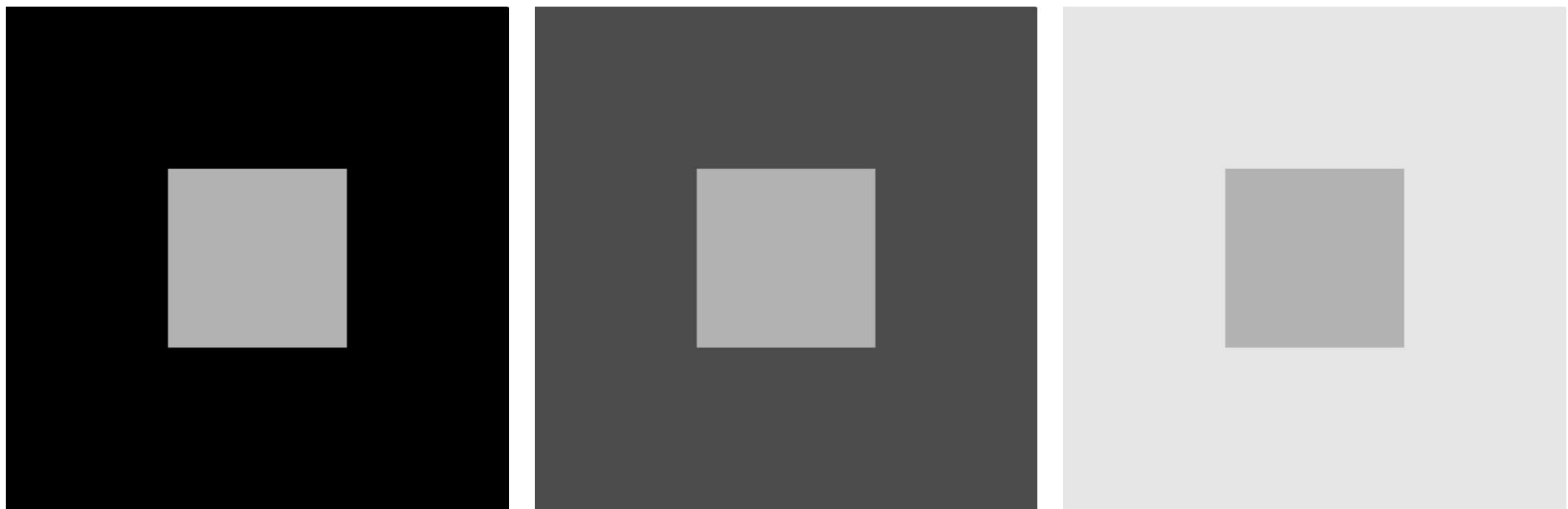


Fig. 1.28: Examples of simultaneous contrast. (From [GW18])

Optical illusions are other examples of human perception phenomena: apparently the eyes fill in nonexisting information or wrongly perceive geometrical properties of objects (see fig. 1.31).

At the following web pages some interesting visual phenomena are demonstrated dynamically:

Müller-Lyer Illusion:

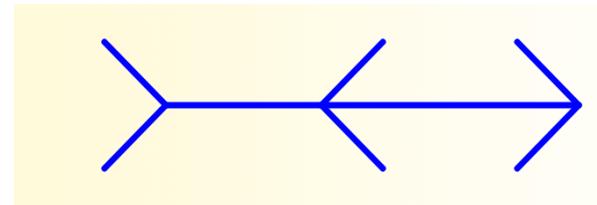


Fig. 1.29: From [Bac19b]

Poggendorff Illusion:

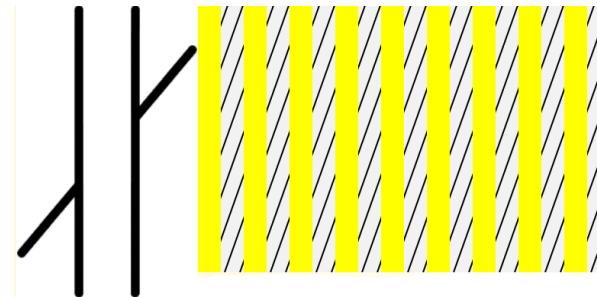
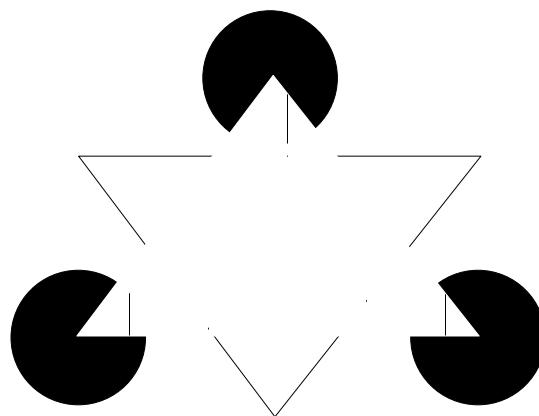
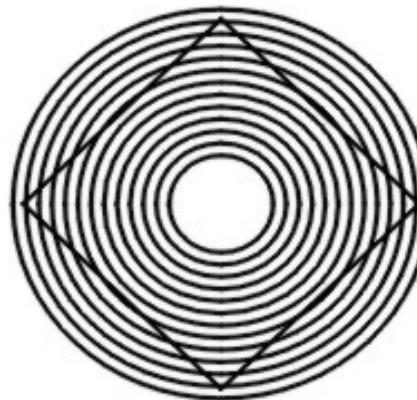


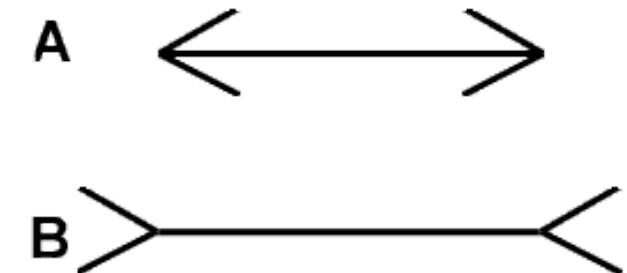
Fig. 1.30: From [Bac19c]



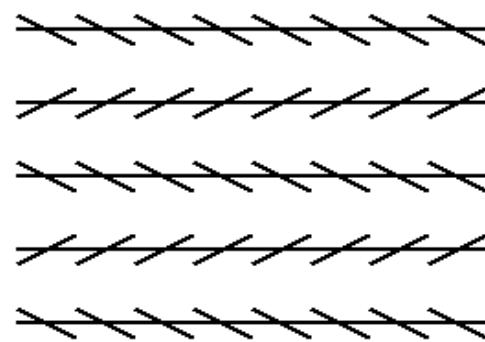
(a)



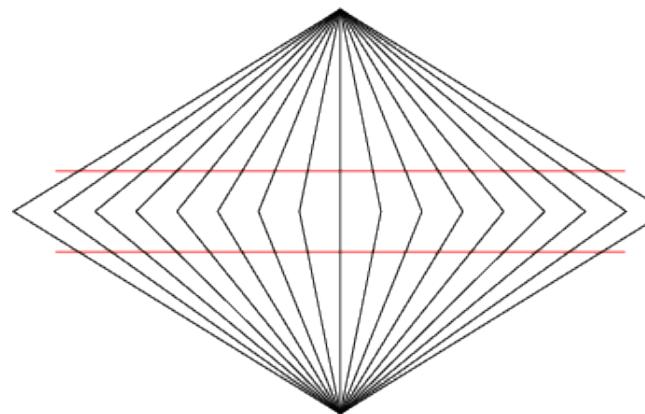
(b)



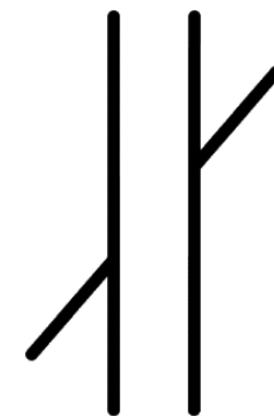
(c)



(d)



(e)



(f)

Fig. 1.31: Some well-known optical illusions. (a) Kanizsa triangle. (b) Ehrenstein illusion. (c) Müller-Lyer illusion. (d) Zöllner's deception. (e) Wundt illusion. (f) Poggendorff illusion.

1.4.2 Gestalt Principles of Visual Organization

Human distinguish between foregrounding and backgrounding.

The terms **figure** and **ground** were introduced by the Gestalt psychologists (**Gestalt Theorie**), notably Max Wertheimer (1880-1943), Wolfgang Köller (1887-1967), and Kurt Koffka (1886-1941).

They outlined what seemed to be several fundamental and universal principles (“laws”) of perceptual organization.

(a) Proximity

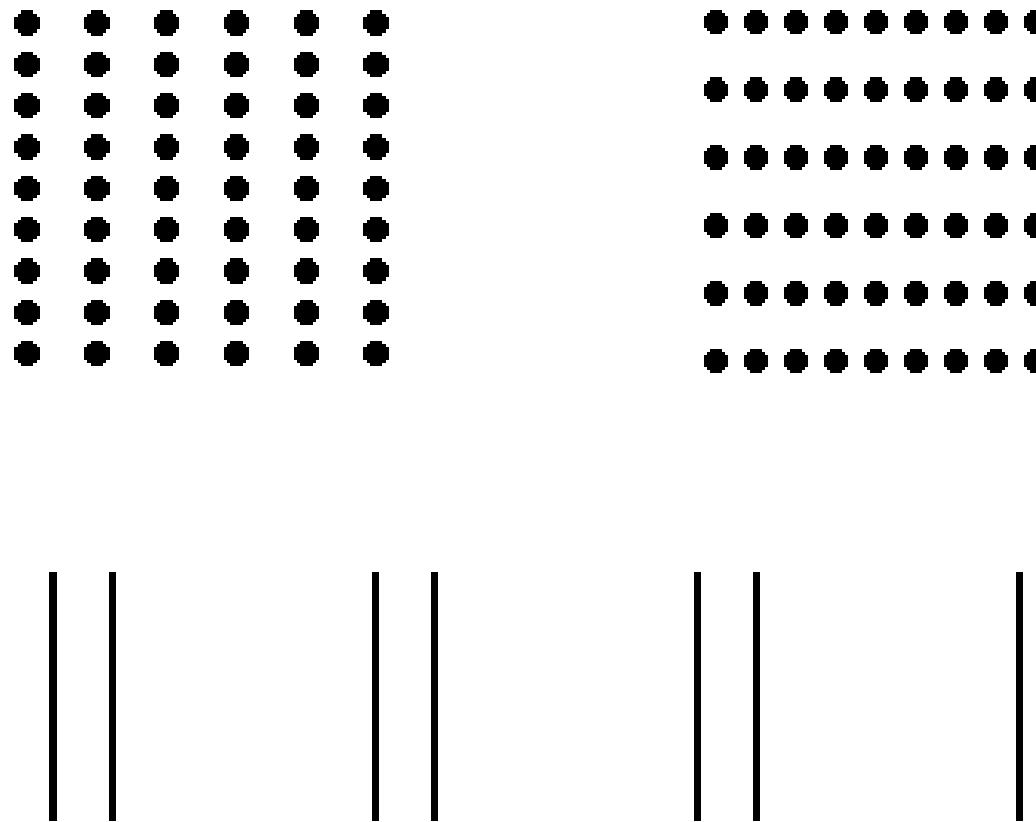


Fig. 1.32: Features which are close together are associated. (From [Cha15])

(b) Similarity

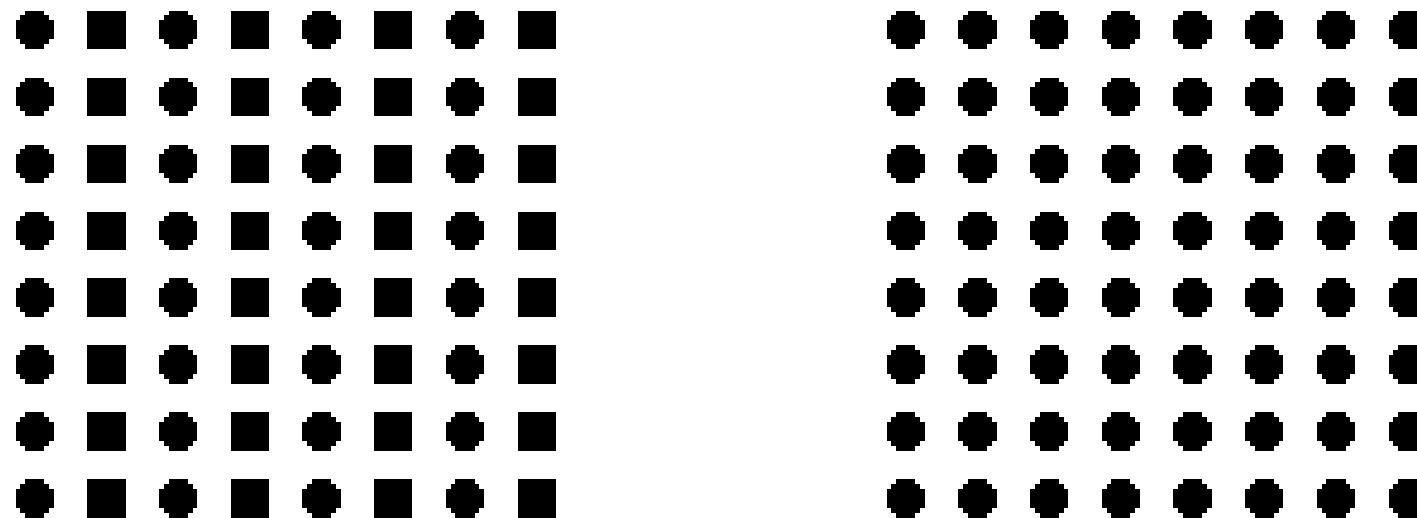


Fig. 1.33: Features which look similar are associated. (From [Cha15])

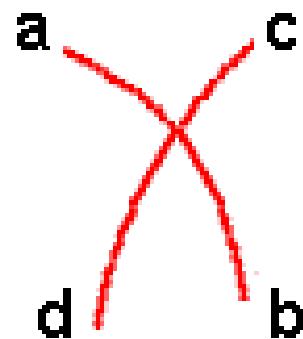
(c) Good continuation

Fig. 1.34: Contours based on smooth continuity are preferred to abrupt changes of direction. (From [Cha15])

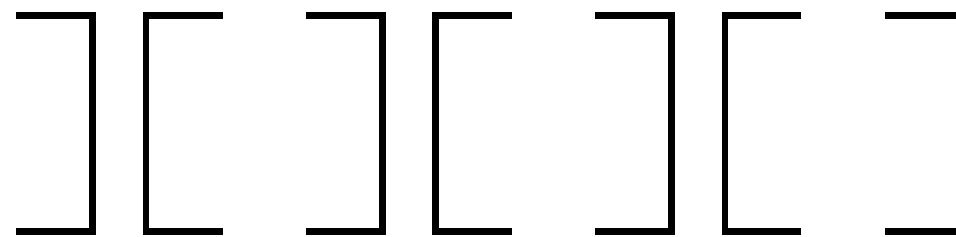
(d) **Closure**

Fig. 1.35: Interpretations of an image which produce “closed” rather than “open” figures are favored. (From [Cha15])

(e) Smallness

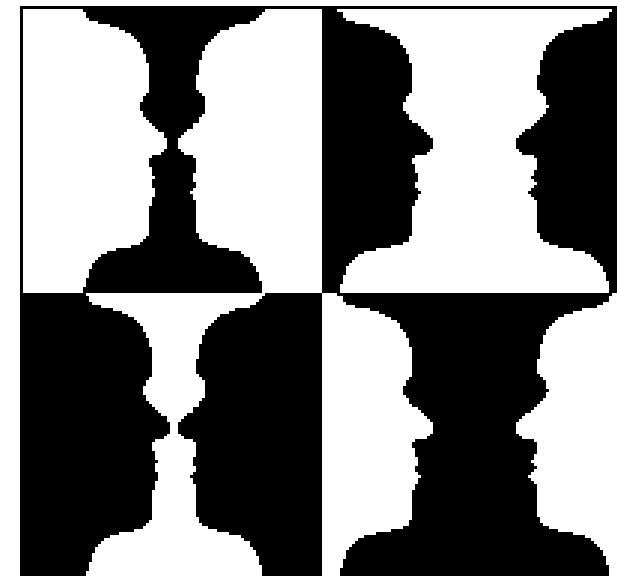
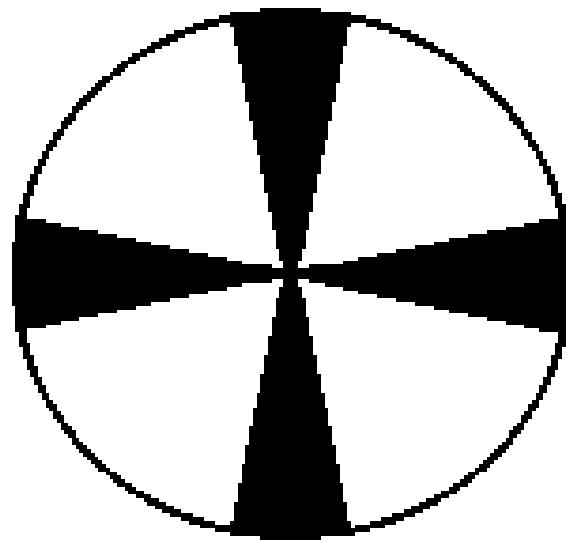


Fig. 1.36: Smaller areas tend to be seen as figures against a larger background. (From [Cha15])

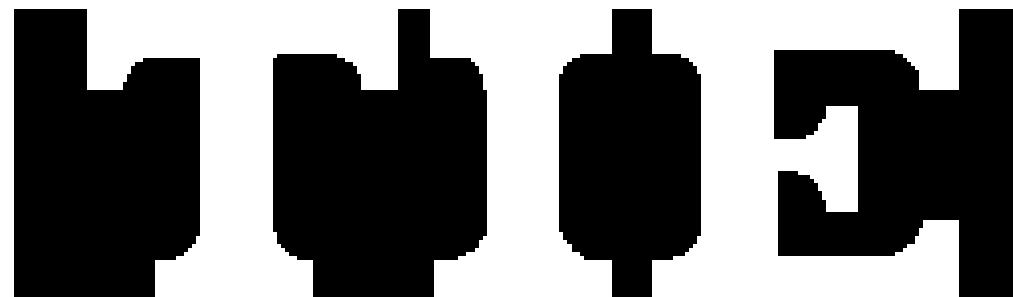
(f) Surroundedness

Fig. 1.37: Areas which can be seen as surrounded by others tend to be perceived as figures. (From [Cha15])

(g) Symmetry

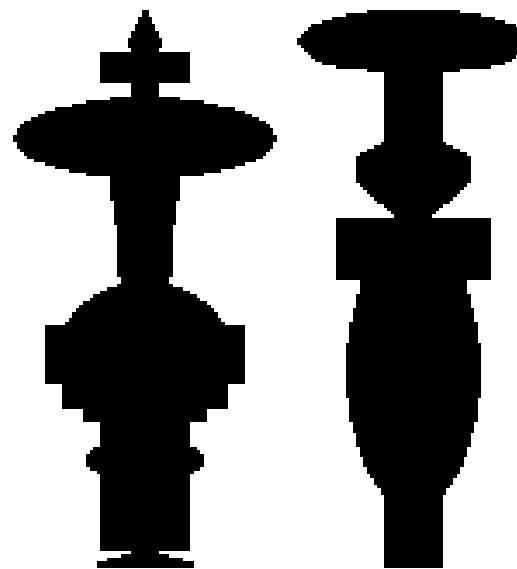


Fig. 1.38: Symmetrical areas tend to be seen as figures against asymmetrical backgrounds. (From [Cha15])

(h) Common fate

Fig. 1.39: If objects move in the same direction, humans tend to see them as a unit. (From [Uni])

All of these principles of perceptual organization serve the overarching principle of **Prägnanz**:

The simplest and most stable interpretations are favored.

1.4.3 Human Object Perception

The following figures give some insights, how humans perceive objects.



Fig. 1.40: Uncommon arrangement: Do you recognize this image? (From [Cha15])



Fig. 1.41: Motion supports figure-ground-segmentation. (From [Bac19a])

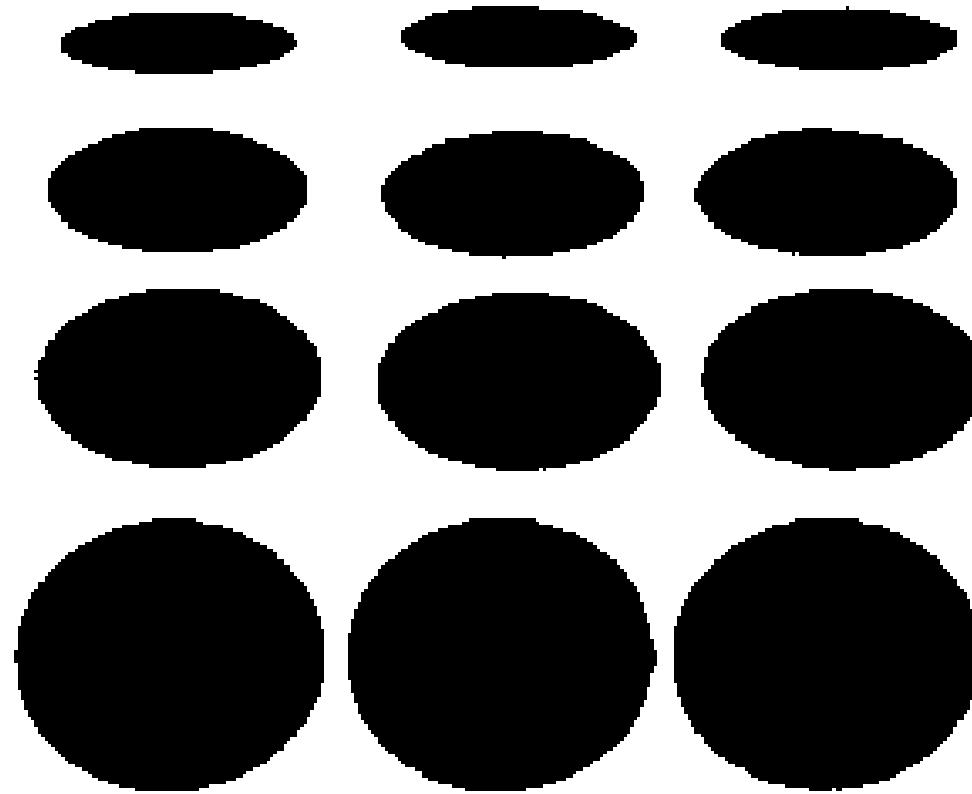


Fig. 1.42: The third dimension. (From [Cha15])

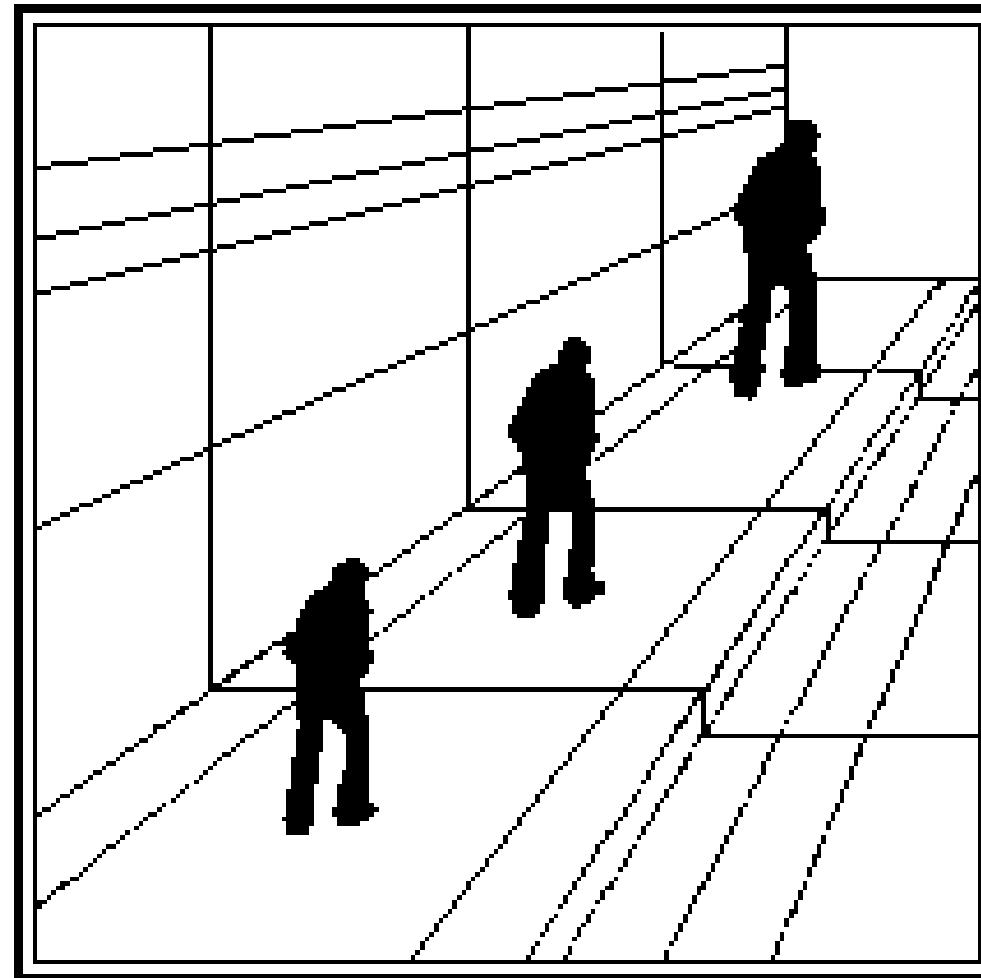


Fig. 1.43: Assumption of linear perspective. (From [Cha15])

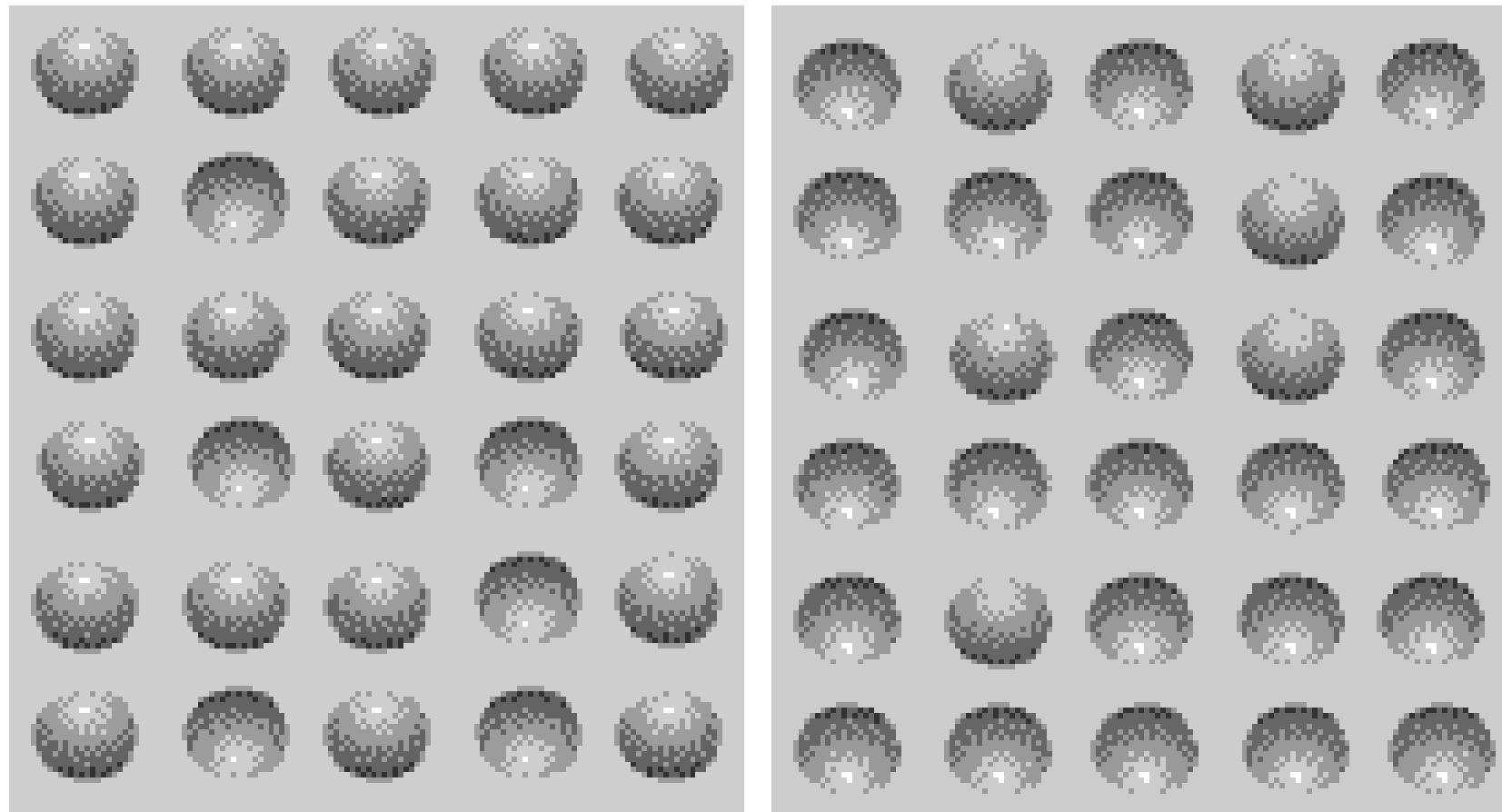


Fig. 1.44: Shadows are an important cue for depth. (From [Cha15])

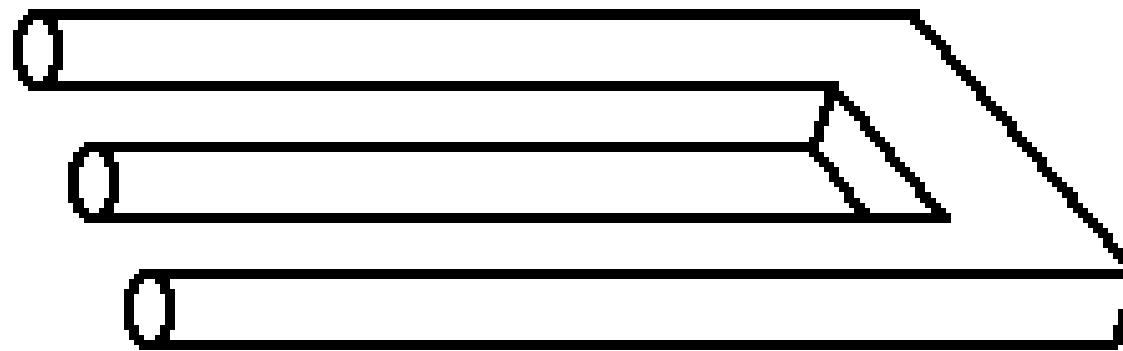


Fig. 1.45: Impossible objects. (From [Cha15])

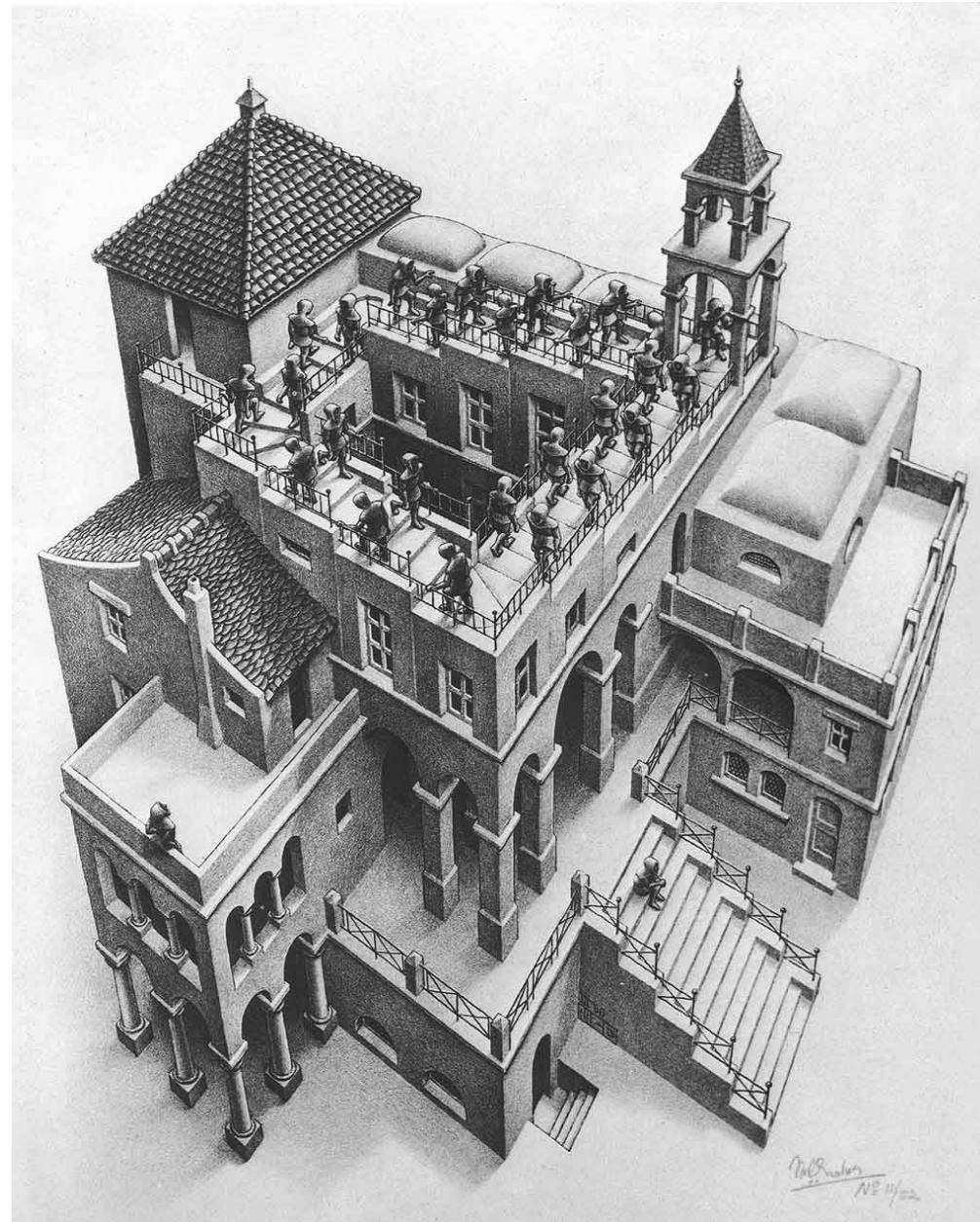


Fig. 1.46: Maurits Cornelis Escher: Ascending and Descending (Lithography). (From [Esc60])

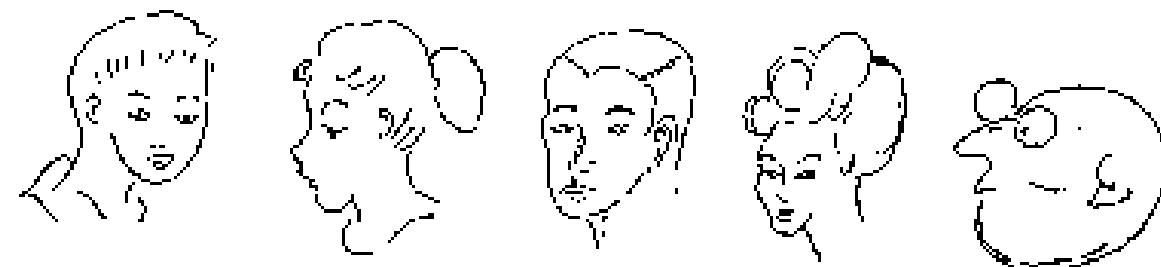


Fig. 1.47: Ambiguous images: importance of the situational context I. (From [Cha15])

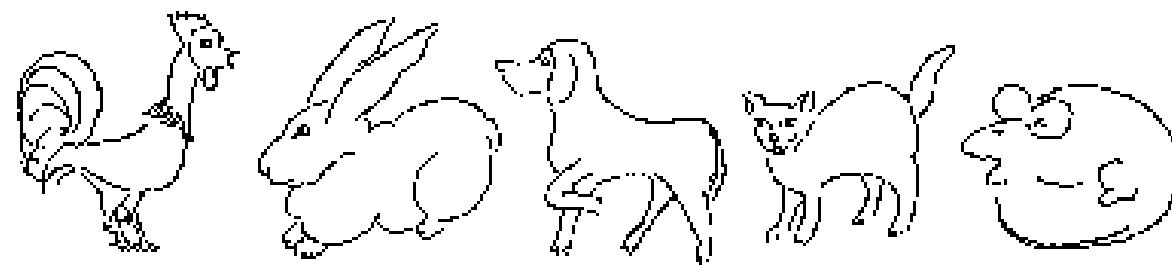


Fig. 1.48: Ambiguous images: importance of the situational context II. (From [Cha15])



Fig. 1.49: Ambiguous images: importance of the situational context III. (From [Cha15])



Fig. 1.50: Importance of perceptual set. (From [Cha15])

YELLOW	BLUE	ORANGE
BLACK	RED	GREEN
PURPLE	YELLOW	RED
ORANGE	GREEN	BLACK
BLUE	RED	PURPLE
GREEN	BLUE	ORANGE

Fig. 1.51: Significance of colors for perception. (Adapted from [Cha15])

1.4.4 Computer vision vs. biological vision

Cognitive Science (“Kognitionswissenschaft”) investigates vision in biological systems:

- Empirical models which adequately describe biological vision
- Describes vision as a computational system

Computer vision research is interested in biological vision:

- Biological vision systems have solved problems which are not yet solved in computer vision
- They provide ideas for technical solutions
- Technical requirements for vision systems often match requirements for biological vision

Hint: Mimicking biological vision does not necessarily provide the best solution for a technical problem.

1.5 History of Computer Vision

Some milestones of the way towards a new discipline:

A Vision of Computer Vision

“... eyes and ears for the computer”
Selfridge 1955

First image enhancement and image processing applications

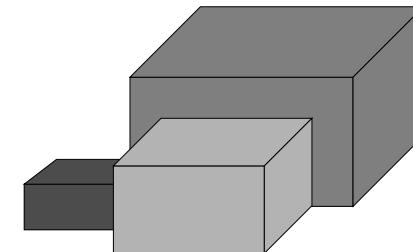
Space missions, aerial image processing

Character recognition



Blocksworld, restricted domains

Roberts 1965:



Natural scenes with motion

Nagel 1979

Visual agents, active vision

Bajcsy 1988

1.6 Fundamental Steps in DIP

An image may be defined as a two dimensional function $f(x, y)$, where

- x and y are **spatial coordinates** (plane coordinates), and
- the amplitude of f at any pair of coordinates (x, y) is called the **intensity** or **gray level** of the image at that point.

Hint: x , y , and f are finite, discrete quantities
→ **digital image**

A digital image is composed of a finite number of elements (**picture elements**, **pels**, or **pixels**).

Regarding the terms in section 1.1, there is no general agreement, where image processing stops and other related areas, such as image analysis and image understanding, start.

It is widely accepted to see computer vision as a knowledge-based process (see fig. 1.52).

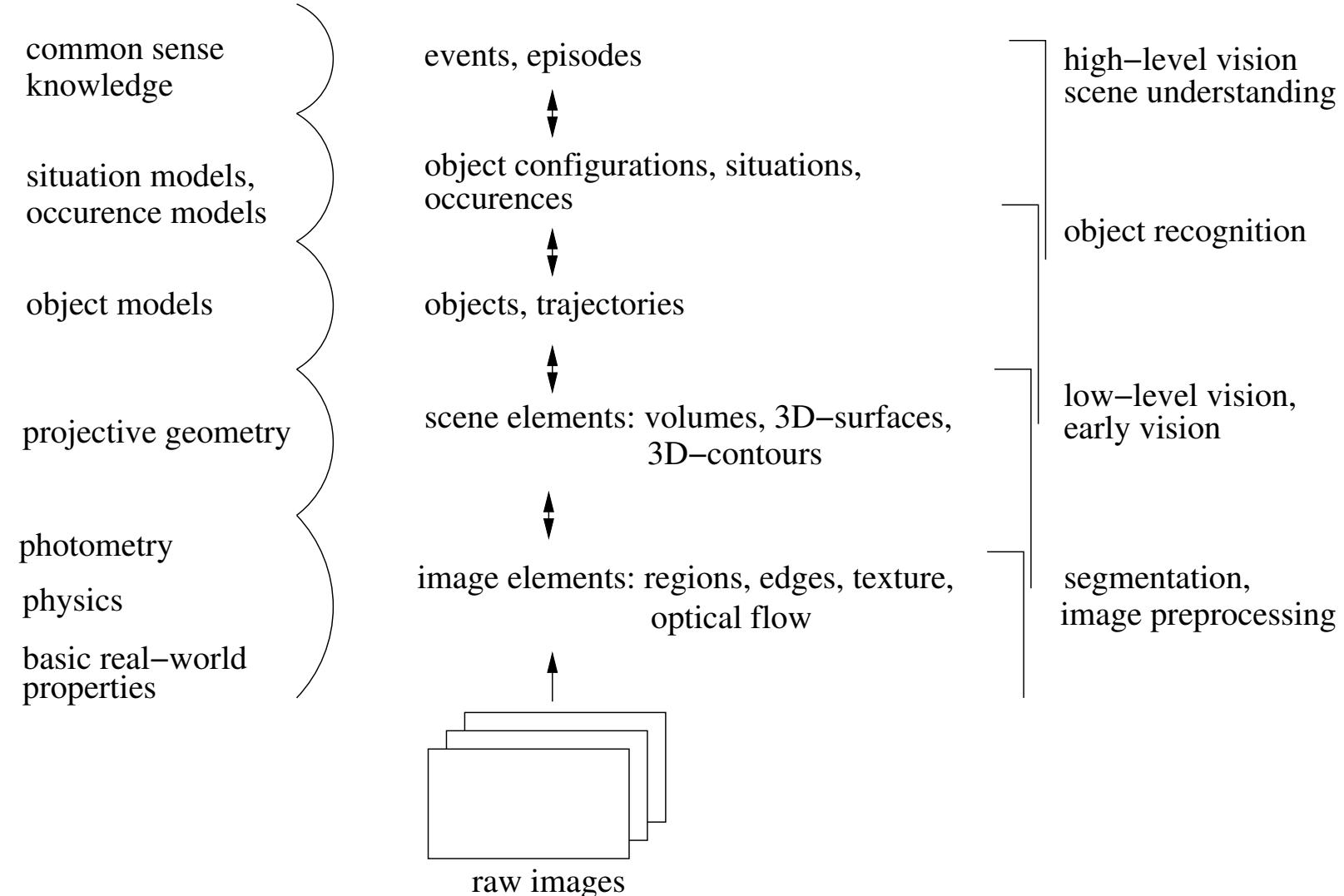


Fig. 1.52: Computer vision as a knowledge-based process.

It is useful to introduce different abstraction layers for the description of computer vision systems:

- **implementation level** What computer hardware is used?
 What programming language is used?
- **algorithmic level** How is the relevant information represented?
 What algorithms are used to process the information?
- **knowledge level** What knowledge or information enters a process?
 What knowledge is obtained by a process?
 What are the laws and constraints which determine the behavior of a process?

Fig. 1.53 summarizes the fundamental processing steps in computer vision systems.

The figure does not imply that every process is applied to an image, but to convey an idea of all the methodologies that can be applied to images for different purposes and possibly with different objectives.

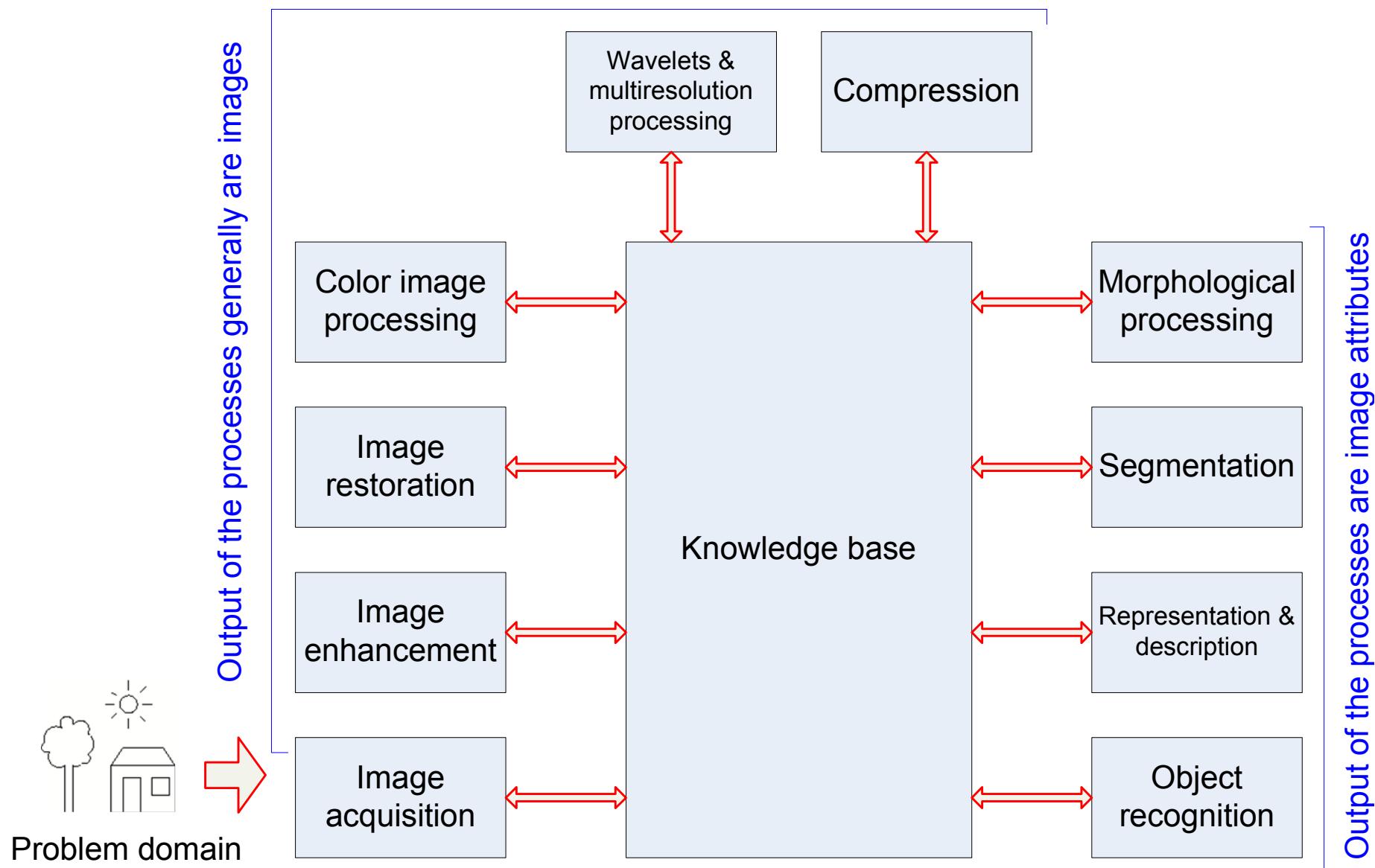


Fig. 1.53: Fundamental steps in digital image processing. (After [GW18])

The next figure depicts the main components of a general purpose image processing system.

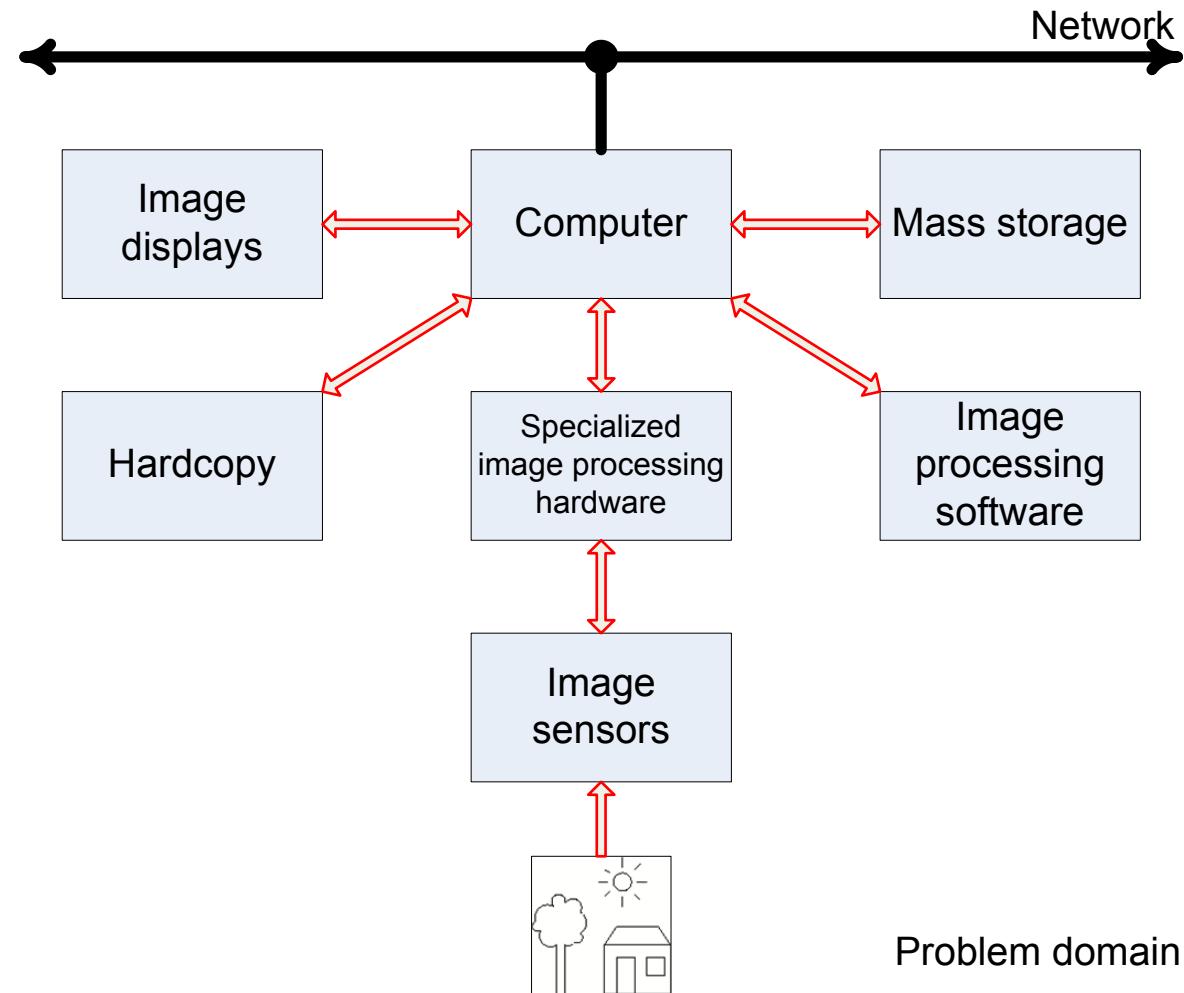


Fig. 1.54: Components of a general purpose DIP system. (After [GW18])

2 Image Formation and Image Models

Contents

2.1 Electromagnetic Spectrum	2-1
2.2 Camera Models	2-5
2.2.1 Perspective Projection	2-6
2.2.2 Thin Lens Model	2-8
2.3 Image Formation	2-12
2.3.1 Formation of Natural Images	2-14
2.3.2 Intensity of sensor signals	2-15
2.3.3 Simple image formation model	2-18
2.4 Image Sampling and Quantization	2-21
2.4.1 Basic concepts in Sampling and quantization	2-21
2.4.2 Representing digital images	2-24
2.4.3 Aliasing and Moiré Patterns	2-32
Periodic Patterns - Spatial Frequencies in Digital Image Processing	2-34

2.1 Electromagnetic Spectrum

Consideration of the electromagnetic spectrum in more detail (see fig. 1.8).

Observation: A beam of sunlight passing through a glass prism is not white but consists of a continuous spectrum of colors ranging from violet at one end to red at the other (Sir Isaac Newton, 1666) (see fig. 2.1).

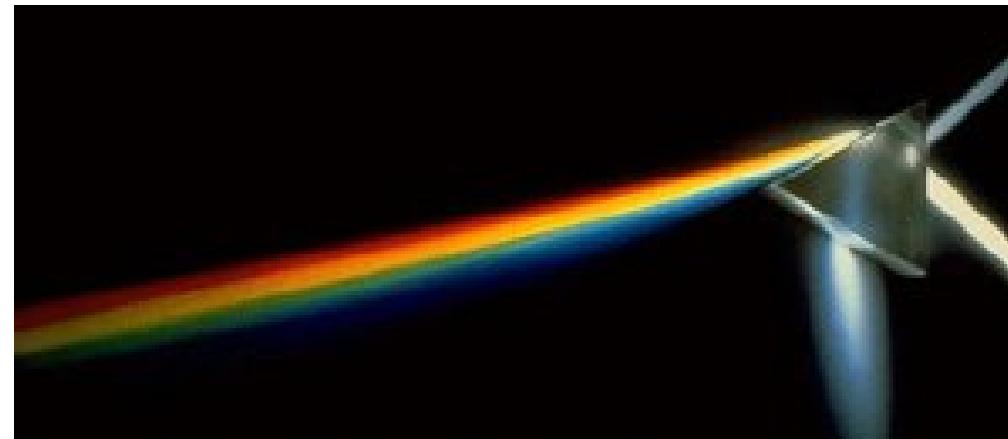


Fig. 2.1: Visible Spectrum.

Notice: The range of the visible spectrum represents only a very small portion of the entire electromagnetic spectrum.

The electromagnetic spectrum can be expressed in terms of three quantities:

- wavelength λ
- frequency f , and
- energy E .

Wavelength and frequency are related by the expression

$$\lambda = \frac{c}{f} \tag{2.1}$$

where c: speed of light, ($c = 2,998 \times 10^8 ms^{-1}$)

energy and frequency by

$$E = hf \tag{2.2}$$

where h: Planck's constant ($h = 6,6 \times 10^{-34} Js$)

Interpretation: Ultraviolet waves have more energy than infrared ones.

Visible spectrum:

- **Light:** particular type of electromagnetic radiation that can be perceived by humans
- Range: $\sim 0,43 \mu m - \sim 0,78 \mu m$
- **Achromatic or monochromatic light:**
 - ▷ light void of color
 - ▷ The only attribute of such light is its **intensity**
 - ▷ The **gray level** describes the monochromatic intensity (it ranges from black to grays to white)
- **Chromatic light:**
 - ▷ It consists of a part of the visible spectrum.
 - ▷ Its quality can be described with three basic quantities:
 - **Radiance:** total amount of energy emitted from a light source
 - **Luminance:** Amount of energy an observer perceives from a light source
 - **Brightness:** Subjective descriptor of light perception

Usage of frequency bands:

A band of the electromagnetic spectrum can be used for imaging when a sensor can be developed that is capable of detecting energy radiated by that band.

Hint: To “see” an object requires that the wavelength of an electromagnetic wave is of the same size as or smaller than the object.

Example 2.1 Detection of a water molecule

Diameter of molecule: $\sim 10^{-10} m$

Source required which is capable of emitting in the X-ray region.

2.2 Camera Models

This section deals with the basic geometry of projection of 3D points, curves, and surfaces onto a 2D surface, the **image plane**.

Hint: We use a right-handed coordinate system for the camera, with the x-axis as the horizontal direction and the y-axis as the vertical direction. The optical axis is the negative z-axis.

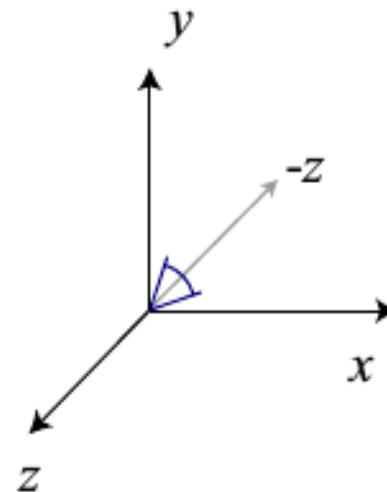


Fig. 2.2: Camera coordinate system.

Hint: We use $P(X, Y, Z)$ to denote 3D points and $p(x, y)$ to denote 2D points.

2.2.1 Perspective Projection

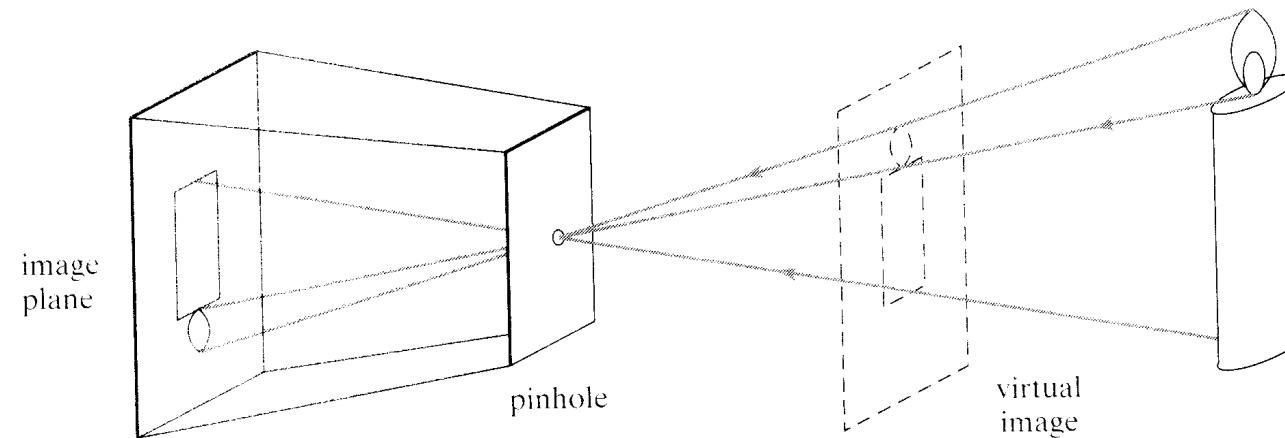


Fig. 2.3: The pinhole camera model.

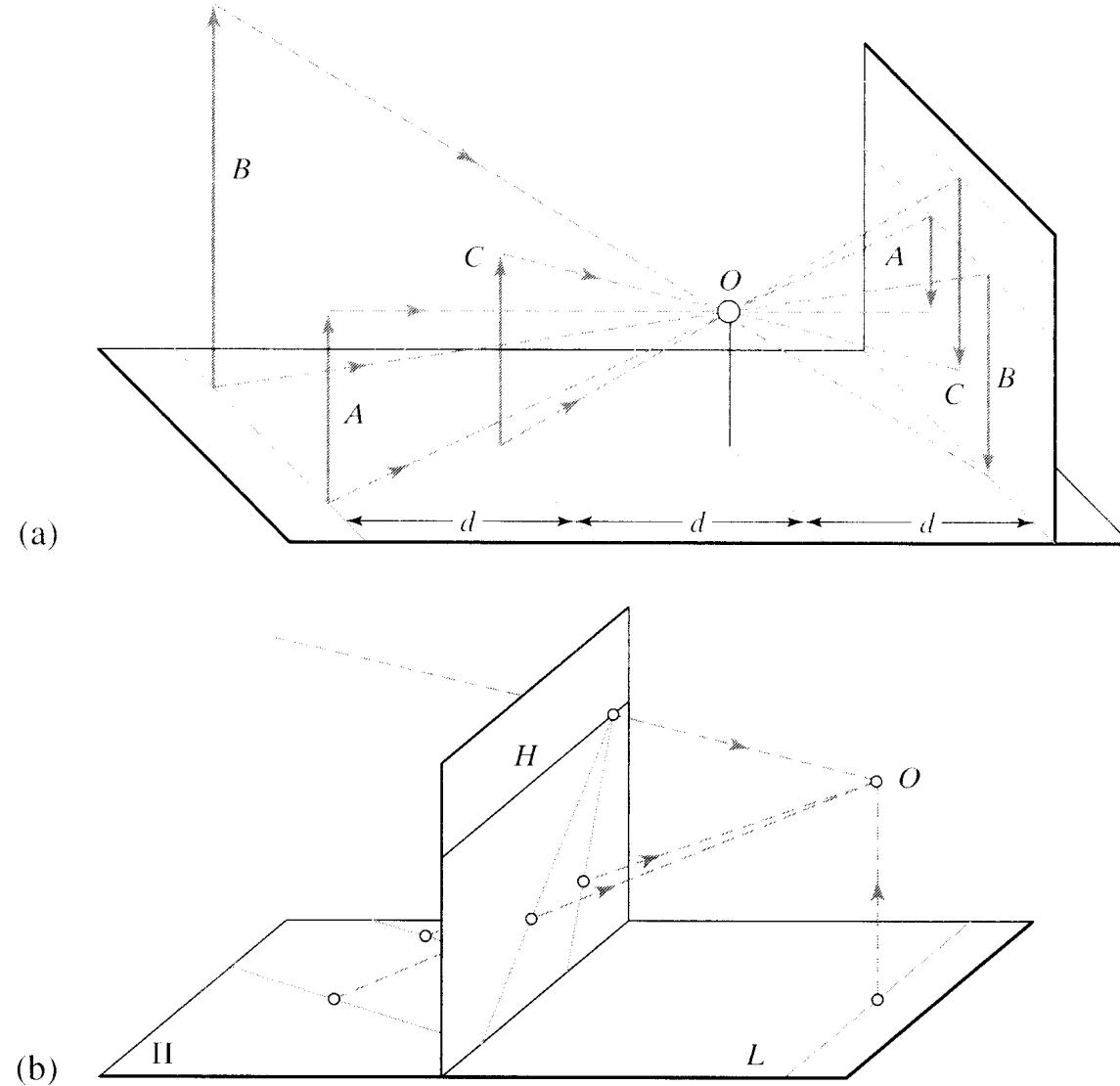


Fig. 2.4: Perspective effects.

2.2.2 Thin Lens Model

If the hole in a pinhole camera is replaced by a lens, the features of the projection can be improved.

Fig. 2.5 shows the projection in the case of a thin convex lens where R is the radius of the spherical surfaces and the refraction index $n = 1$.

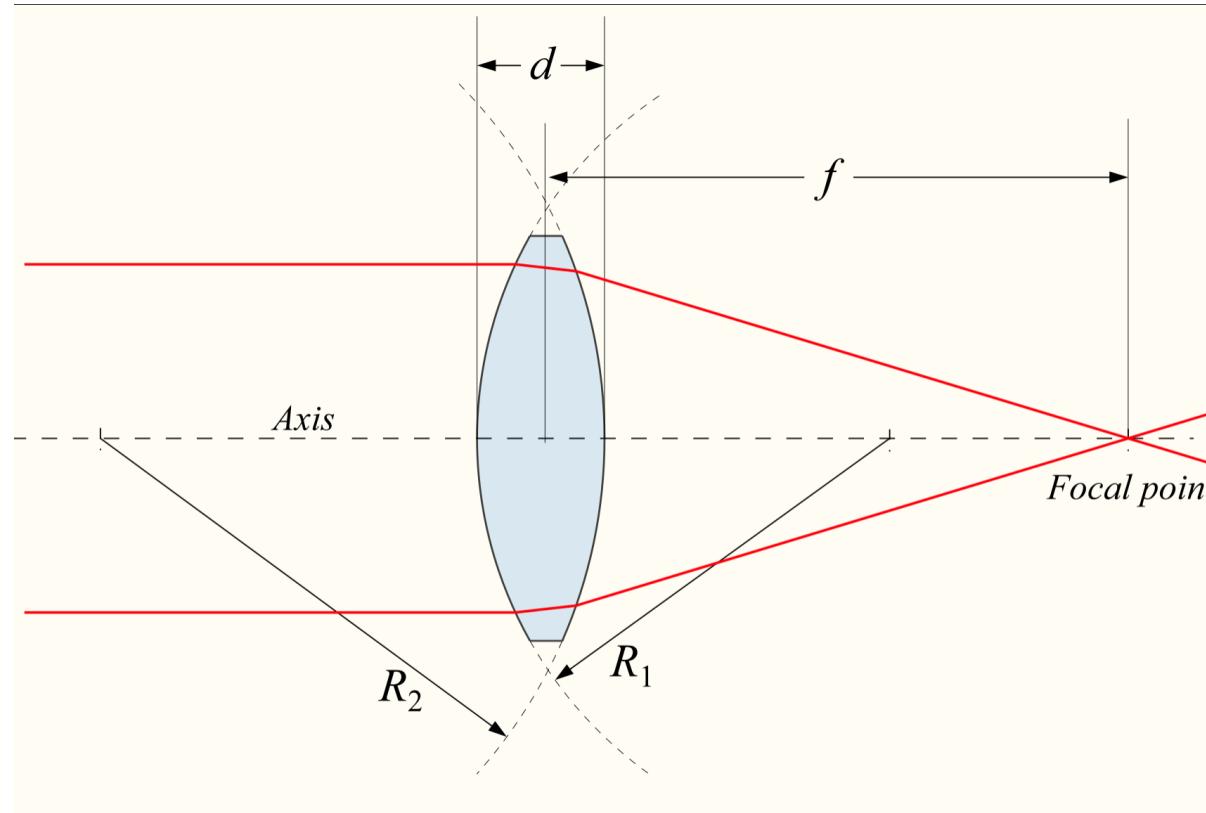


Fig. 2.5: A collimated light beam converges at a point behind the lens.

A collimated light beam converges at a point behind the lens, the **focal point** F . The distance from the center of the lens to focal point is called **focal length** f .

Fig. 2.6 shows the mapping of a point P onto a point P' in the image plane.

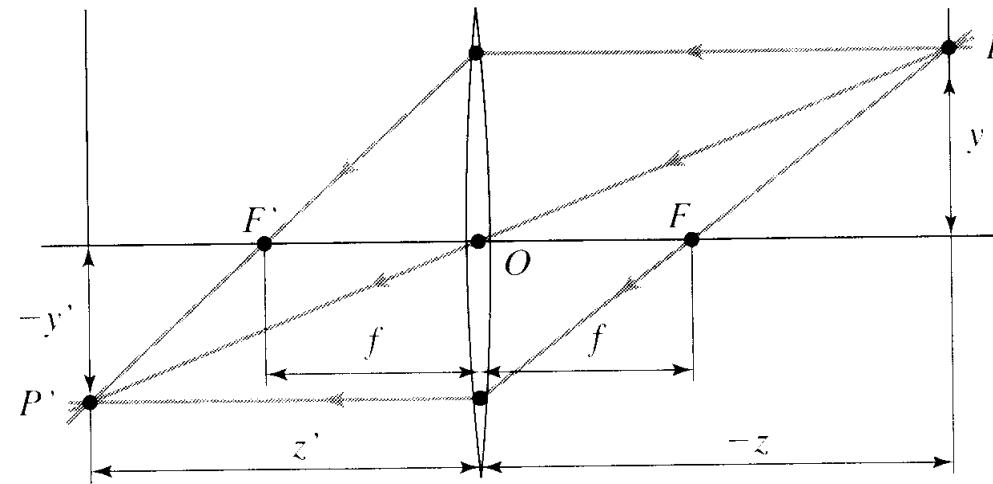


Fig. 2.6: Rays passing through a thin lens.

In case of a magnification factor of 1, the **thin lens equation** holds:

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f} \quad (2.3)$$

where

$$f = \frac{R}{2(n - 1)} \quad (2.4)$$

- Hint:**
- The points P and P' relate to the pinhole perspective projection if $z' = f'$.
 - Only points located at $-z$ from O are only in sharp focus when the image plane is located at z' from O .
 - In practice: Points within some range of distance are in acceptable focus (**Depth of focus**).

2.3 Image Formation

Images can be generated by various processes:

- Illumination of surfaces, measurement of reflections
→ natural images
- Illumination of translucent material, measurement of irradiation
- Measurement of heat (infrared) radiation
- X-ray of material, computation of density maps
- Measurement of any feature by means of a sensor array

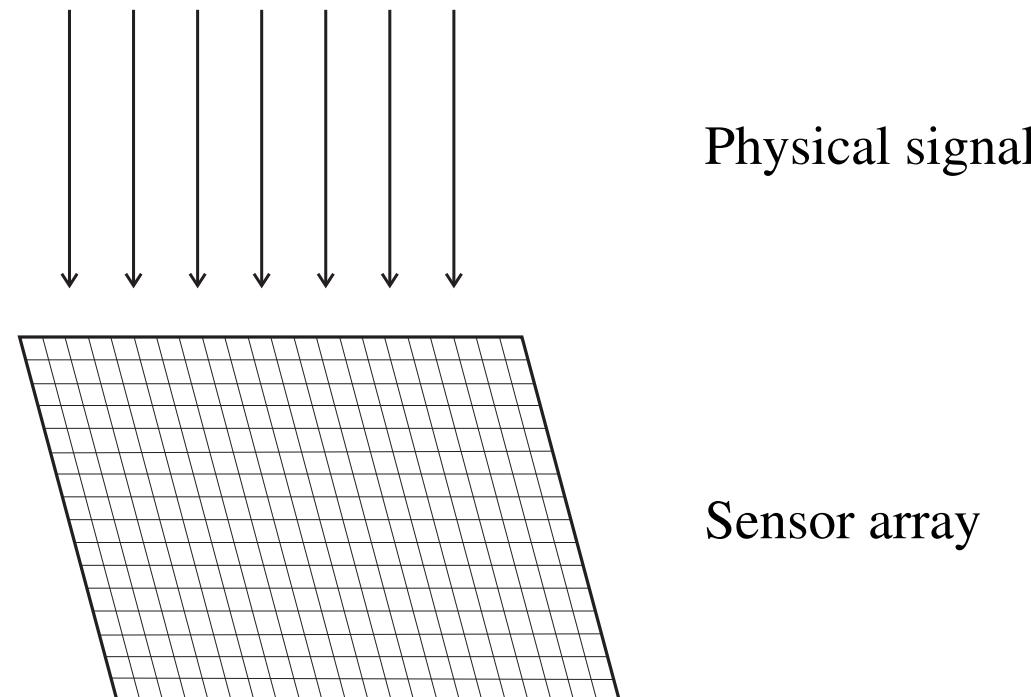


Fig. 2.7: Image formation.

2.3.1 Formation of Natural Images

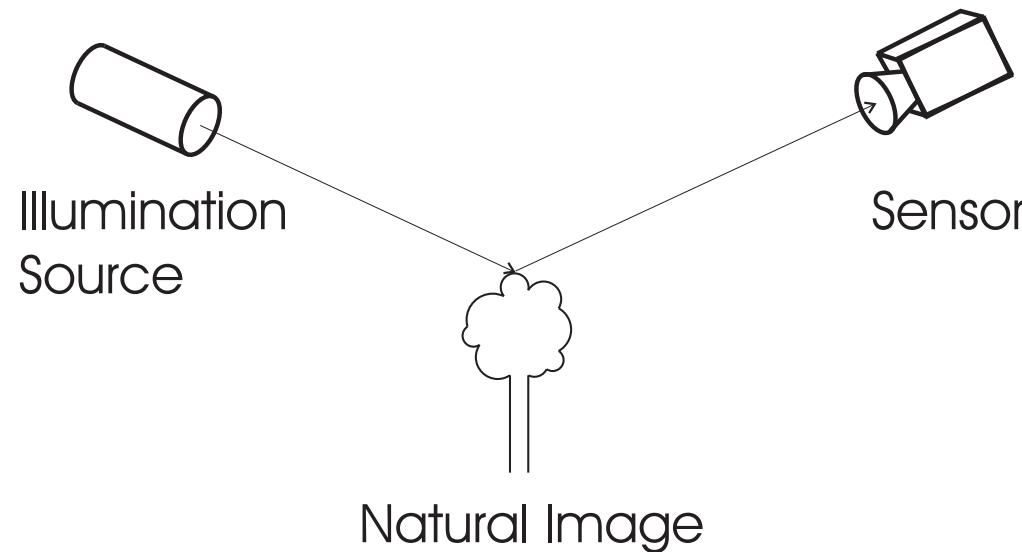


Fig. 2.8: Formation of natural images.

The intensity (brightness) of a pixel depends on

- Illumination (spectral energy, secondary illumination),
- Object surface properties (reflectivity),
- Sensor properties,
- Geometry of light source, object and sensor constellation (angles, distances), and
- Transparency of irradiated medium.

2.3.2 Intensity of sensor signals

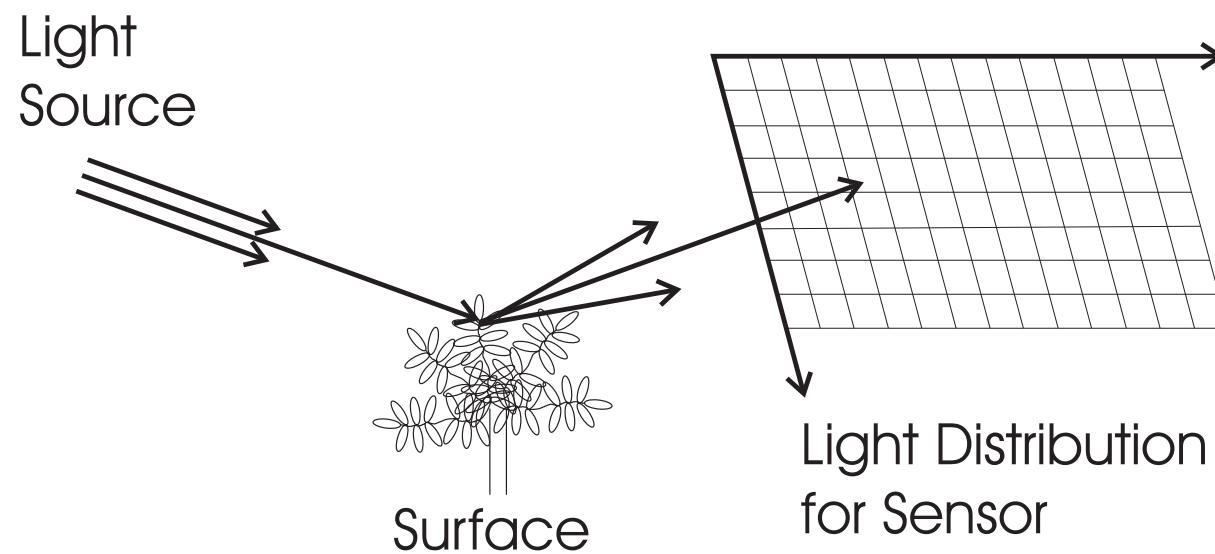


Fig. 2.9: Image function.

The intensities of sensor signals depend on the

- location x, y on the sensor plane,
- instance of time t ,
- frequency of incoming light wave λ , and
- spectral sensitivity of sensor.

$$f(x, y, t) = \int_0^{\infty} C(x, y, t, \lambda) S(\lambda) d\lambda \quad (2.5)$$

where

$C(\dots)$: spectral energy distribution

$S(\dots)$: sensitivity function of sensor

Sensors with different spectral sensitivities generate **multispectral images**:

$$f_1(x, y, t) = \int_0^{\infty} C(x, y, t, \lambda) S_1(\lambda) d\lambda \quad (2.6)$$

$$f_2(x, y, t) = \int_0^{\infty} C(x, y, t, \lambda) S_2(\lambda) d\lambda \quad (2.7)$$

$$f_3(x, y, t) = \int_0^{\infty} C(x, y, t, \lambda) S_3(\lambda) d\lambda \quad (2.8)$$

Example 2.2 RGB-Image

For an RGB-sensor $\vec{\mathbf{S}} = [S'_r(\lambda), S'_g(\lambda), S'_b(\lambda)]$ where the function $S'_i(\lambda)$ with $i \in [R, G, B]$ specify the sensors response characteristics to light in the red, green, and blue part of the spectrum.

R(red) : 650 nm center frequency

G(green) : 530 nm center frequency

B(blue) : 410 nm center frequency

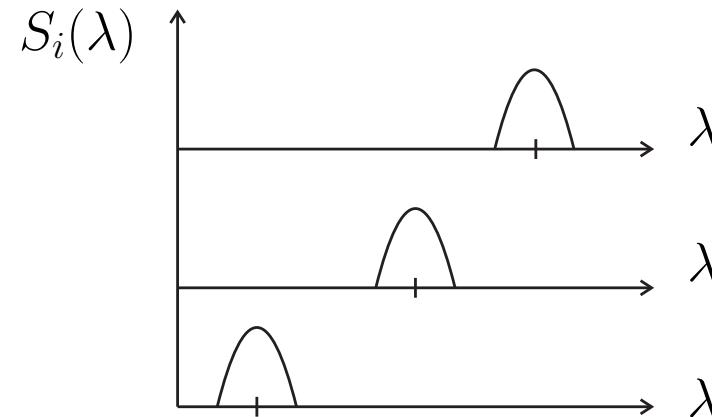


Fig. 2.10: Spectral sensitivities of a sensor.

2.3.3 Simple image formation model

As introduced in sec. 1.6:

An image is denoted by a 2D function $f(x, y)$.

An image can be generated from a physical process, therefore $f(x, y)$ must be nonzero and finite.

$$0 < f(x, y) < \infty \quad (2.9)$$

The image function can be characterized by two components:

- amount of source **illumination** ($i(x, y)$)
- amount of illumination reflected by objects of a scene (**reflectance**, $r(x, y)$)

The product of the two functions results in $f(x, y)$

$$f(x, y) = i(x, y) \cdot r(x, y) \quad (2.10)$$

where

$$0 \leq i(x, y) \leq \infty$$

$$0 \leq r(x, y) \leq 1$$

with

$r = 0$ **total absorption**

$r = 1$ **total reflection**

Illumination	Sun on a clear day	$\sim 90000 \text{ } lm \text{ } m^{-2}$
	Cloudy day	$\sim 10000 \text{ } lm \text{ } m^{-2}$
	Full moon on a clear evening	$\sim 0,1 \text{ } lm \text{ } m^{-2}$
	Typical office illumination	$\sim 1000 \text{ } lm \text{ } m^{-2}$
Reflectance	Black velvet	0,01
	Stainless steel	0,65
	Flat-white wallpaint	0,80
	Silver-plated metal	0,90
	Snow	0,93

Table 2.1: Some typical values of illumination and reflectance. (From [GW18])

2.4 Image Sampling and Quantization

2.4.1 Basic concepts in Sampling and quantization

For the creation of an digital image the continuously sensed data has to be converted into a digital form.

Two processes:

- **sampling** (digitizing the coordinate values) and
- **quantization** (digitizing the amplitude)

These processes are illustrated in fig. 2.11.

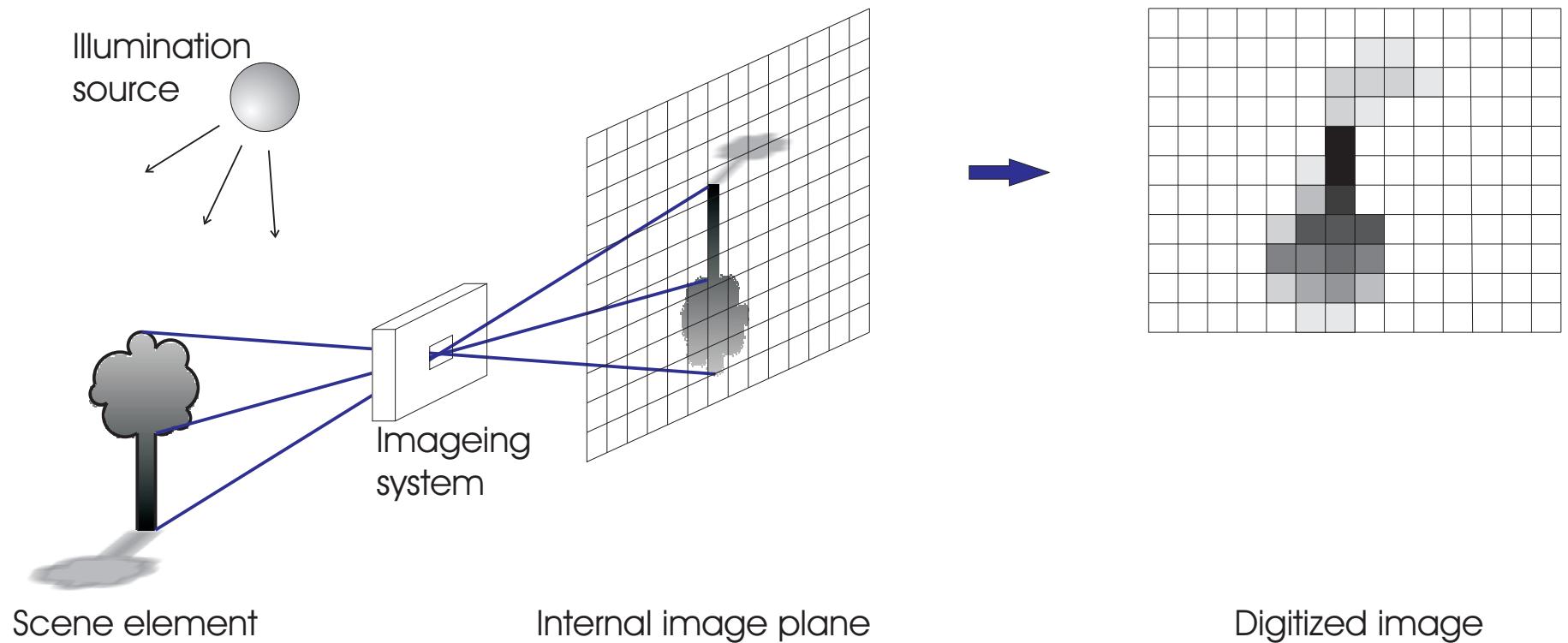


Fig. 2.11: Example of an digital image acquisition process.

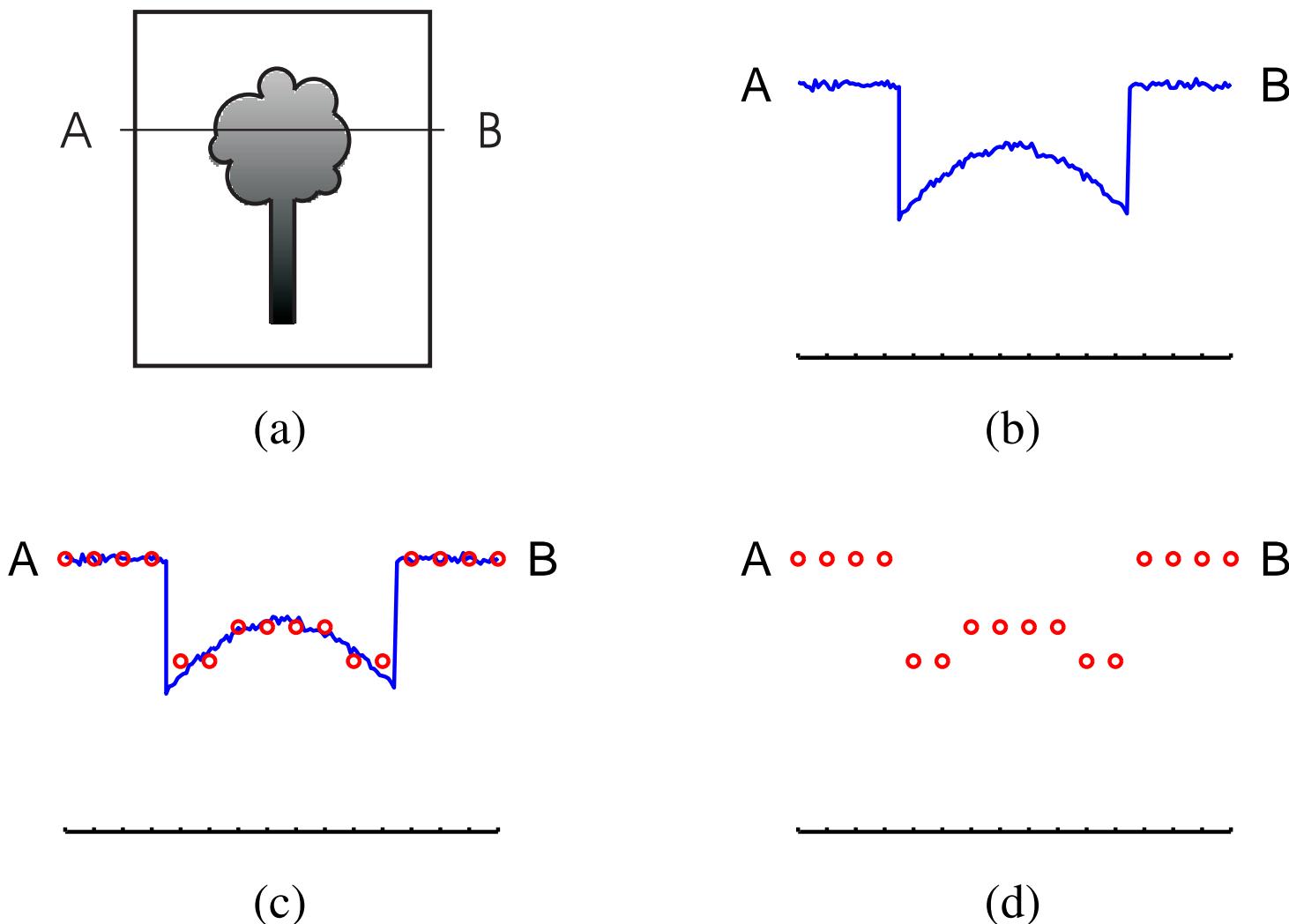


Fig. 2.12: Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line. (Adapted from [GW18])

2.4.2 Representing digital images

Assumption: The resulting digital image has M rows and N columns.

It consists of a matrix of real numbers.

Fig. 2.13 shows the coordinate conventions used in this course.

A complete $M \times N$ digital image can now be written in the following compact matrix form:

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \cdots & f(M - 1, N - 1) \end{bmatrix} \quad (2.11)$$

Hint: Due to processing, storage, and sampling hardware considerations, the number of gray levels typically is an integer power of

$$L = 2^k \quad (2.12)$$

where L : number of discrete gray levels allowed for each pixel

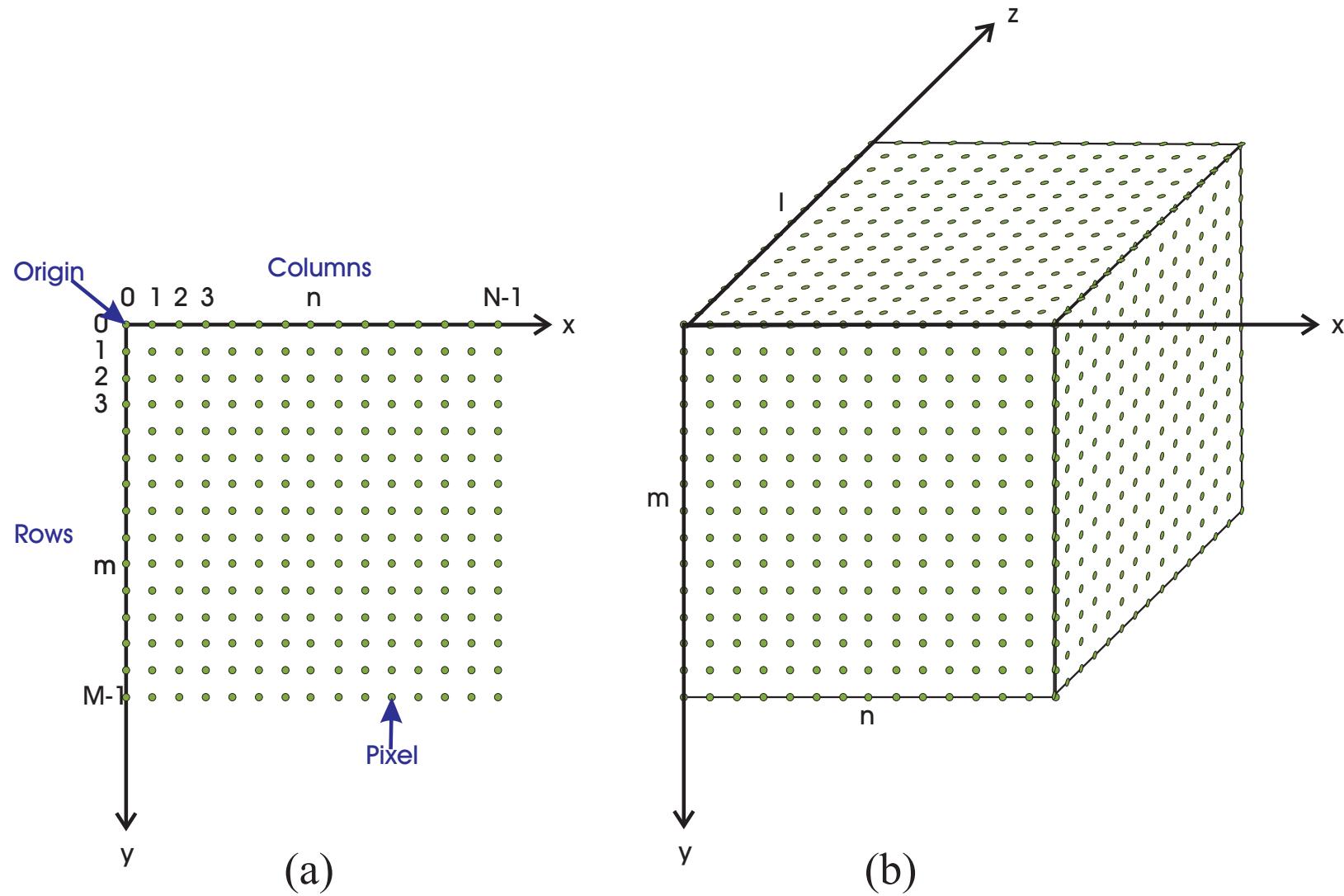


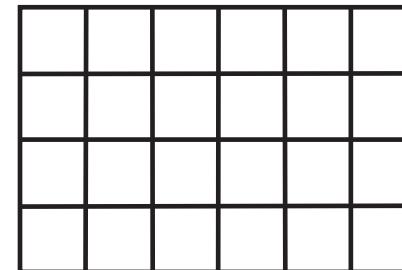
Fig. 2.13: Coordinate conventions in this script. (a) 2D image. (b) 3D image.

The number of bits b , required to store a digitized image is

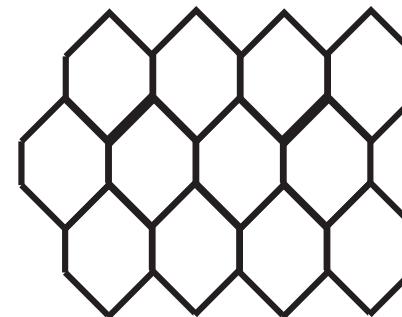
$$b = M \times N \times k \quad (2.13)$$

For the sampling (spatial quantization) exist different grinds.

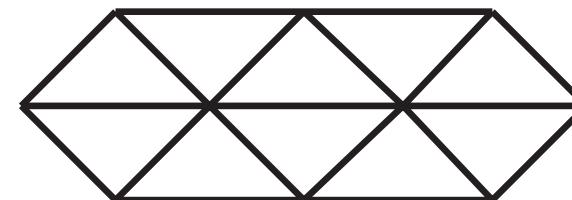
Rectangular grid



Hexagonal grid



Triangular grid



✓

Fig. 2.14: The three possible regular 2D grids.

Typical effects of varying the number of samples in a digital image:

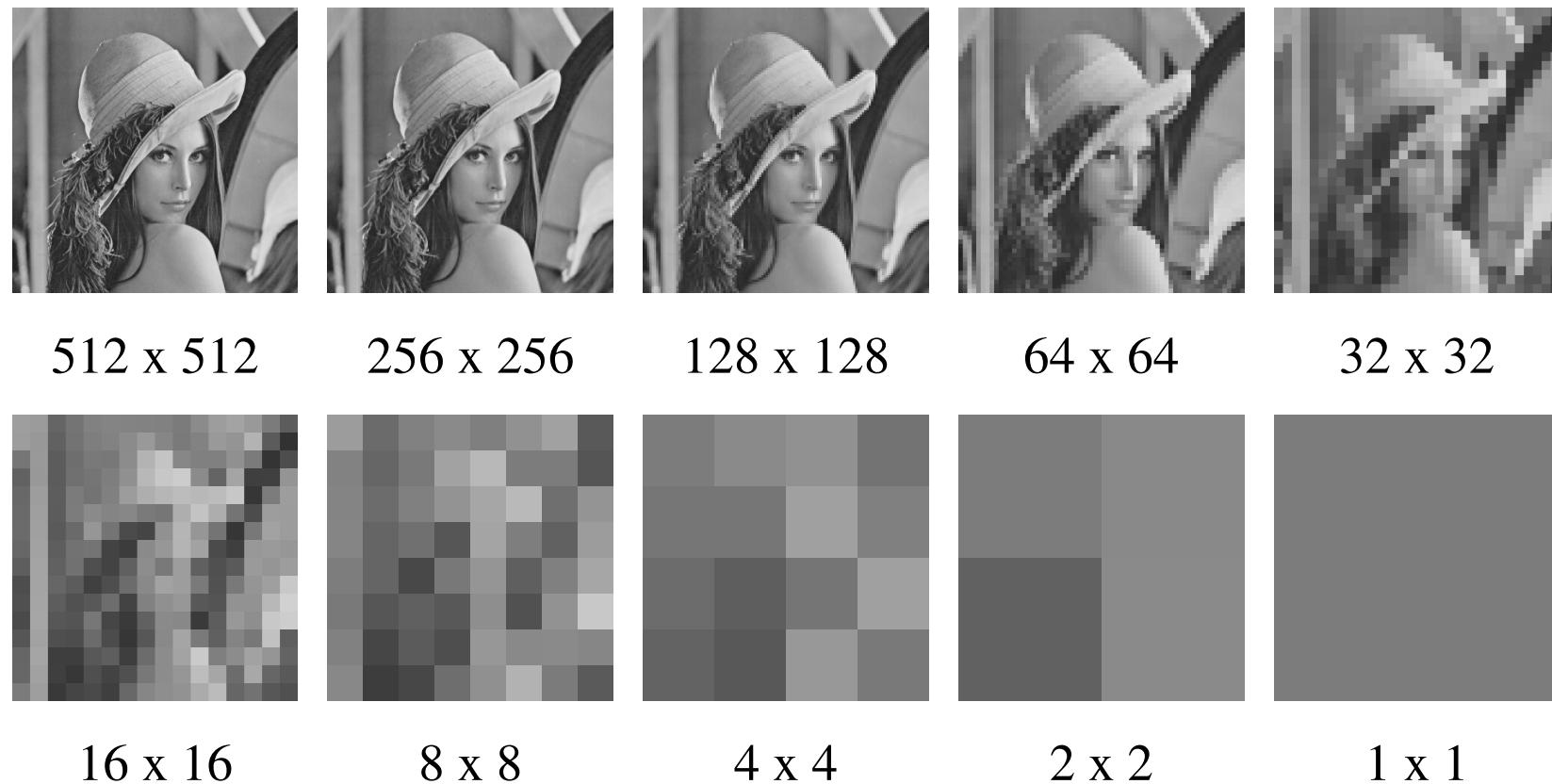


Fig. 2.15: A 512x512, 8-bit image subsampled down to size 1x1. The number of allowable gray levels was kept at 256. The images are resampled into 512x512 pixels.

Typical effects of varying the number of gray levels in a digital image:



Fig. 2.16: A 512x512 image displayed in different gray levels, while keeping the spatial resolution constant.

Quantization and image quality



Fig. 2.17: Image zooming and averaging. (a) 512x512, 8-bit image. (b) 64x64, 8-bit image. (c) same as (b), but averaged.

2.4.3 Aliasing and Moiré Patterns

The reconstruction of an digitized image is only possible, if the “variability” of the image function is restricted.

The sampling theorem by Claude Shannon states for 2D functions:

If the image function is sampled at a rate equal to or greater than twice its highest frequency, it is possible to recover completely the original function from its sample.

Hint: The principal approach for reducing the alias effects caused by undersampling on an image is to reduce its high-frequency components.

However, aliasing is always present in a sampled image.

The effect can be studied in fig. 2.18.

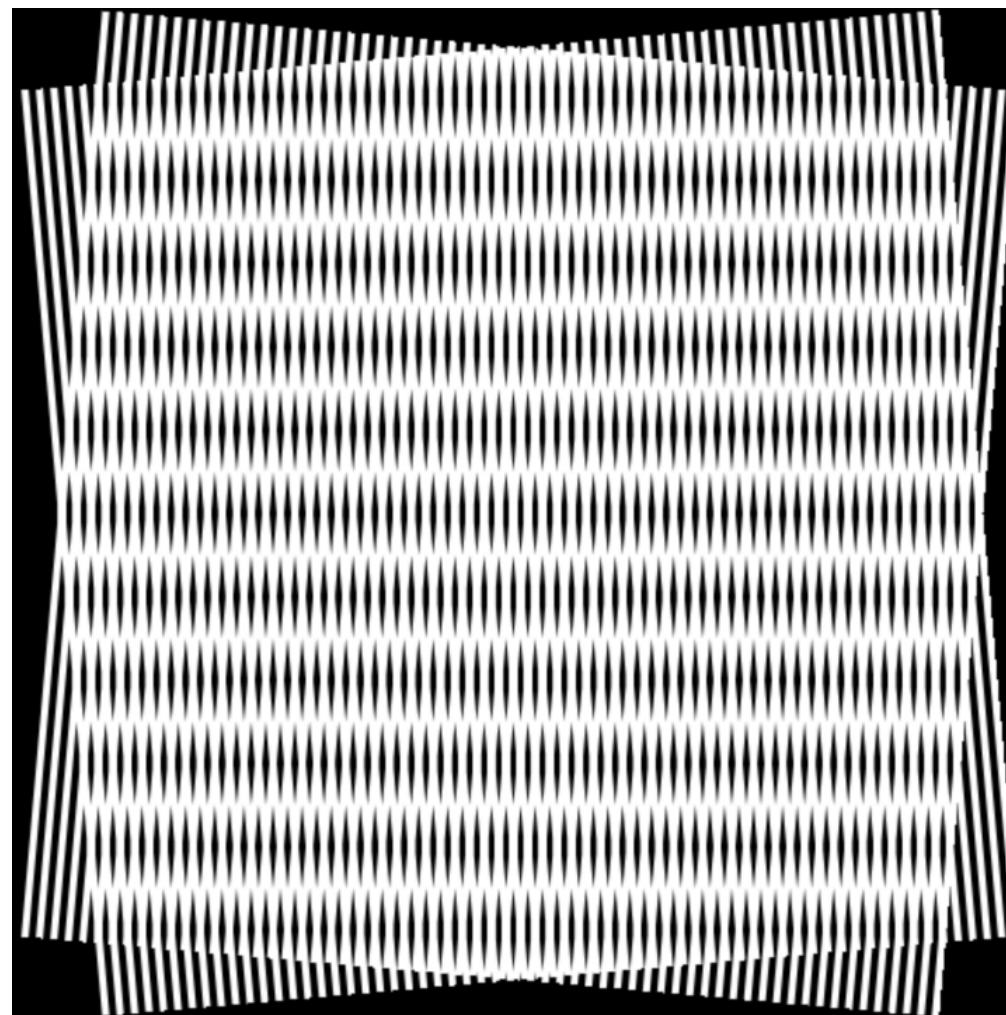
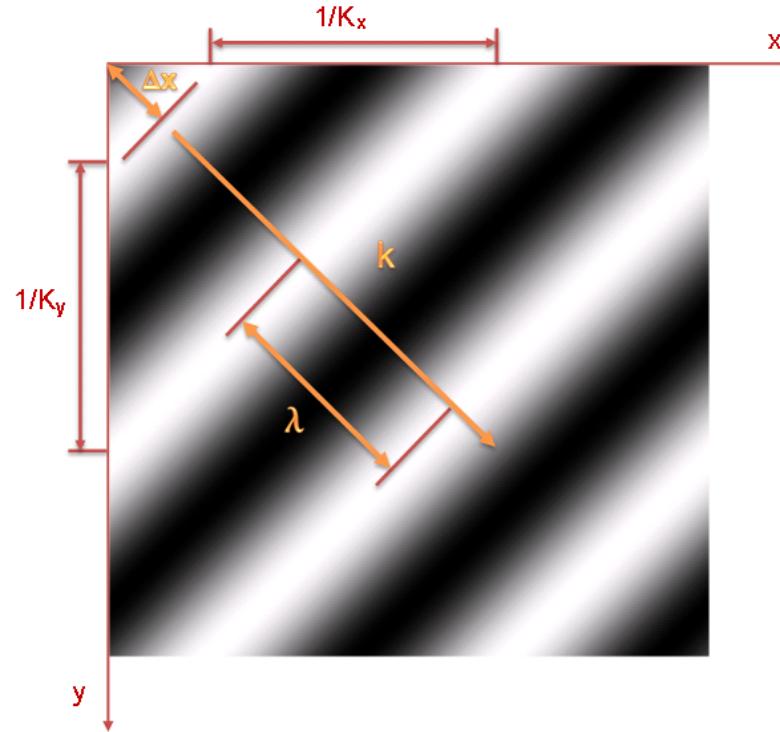


Fig. 2.18: Illustration of the Moiré pattern effect. (From [GW18])

Periodic Patterns - Spatial Frequencies in Digital Image Processing

Describing periodic patterns using the **wavenumber vector**.



- * The wavenumber vector $\vec{k} = [k_x, k_y]^T$
- * The wavelength $\lambda = 1/|\vec{k}|$
- * The phase angle $\phi = 2\pi \frac{\Delta x}{\lambda}$

- * The periodic function

$$f(\vec{\mathbf{x}}) = a \cdot \cos(2\pi \vec{\mathbf{k}}^T \vec{\mathbf{x}} - \phi), \quad a = \frac{255}{2} = 127.5$$

$$f(x, y) = a + a \cdot \cos(2\pi (k_x x + k_y y) - \phi)$$

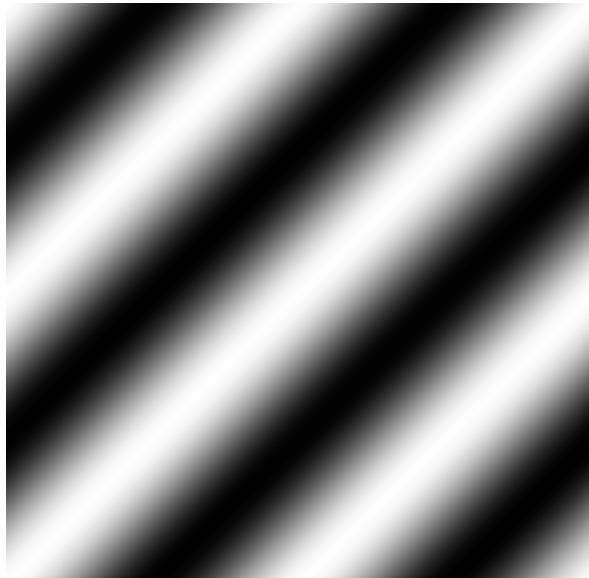
- * The sampling interval should be: $r \leq \frac{1}{2}\lambda$

What happens if?

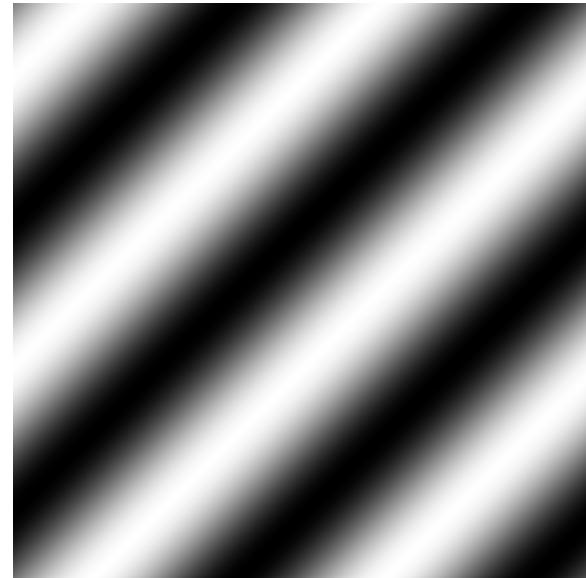
- * The phase angle $\phi = \{0, \frac{\pi}{2}, \pi\}$
- * $\{k_x > k_y\}$ or $\{k_x < k_y\}$
- * The sampling interval $r > \frac{1}{2}\lambda$
- * The periodic function $f(x, y) = 0 + a \cdot \cos(2\pi (k_x x + k_y y) - \phi)$

Answer:

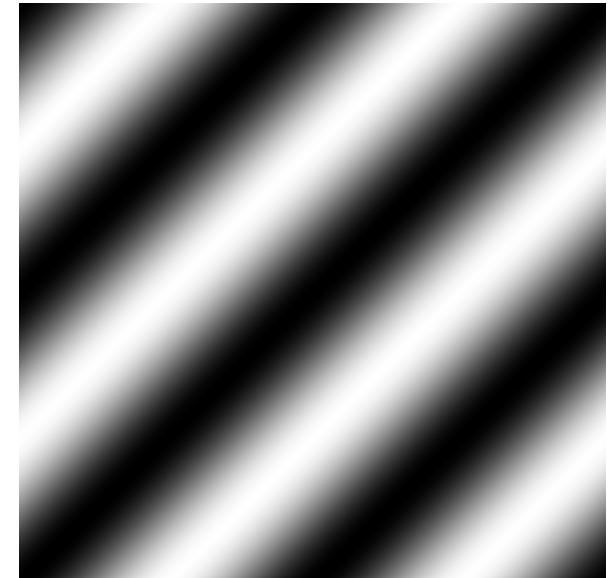
- * The phase angle $\phi = \{0, \frac{\pi}{2}, \pi\}$



(a)



(b)



(c)

Fig. 2.19: Change of phase angle ϕ (a) $\phi = 0$ (b) $\phi = \frac{\pi}{2}$ (c) $\phi = \pi$.

- * $\{k_x > k_y\}$ or $\{k_x < k_y\}$
- * The sampling interval $r > \frac{1}{2}\lambda$

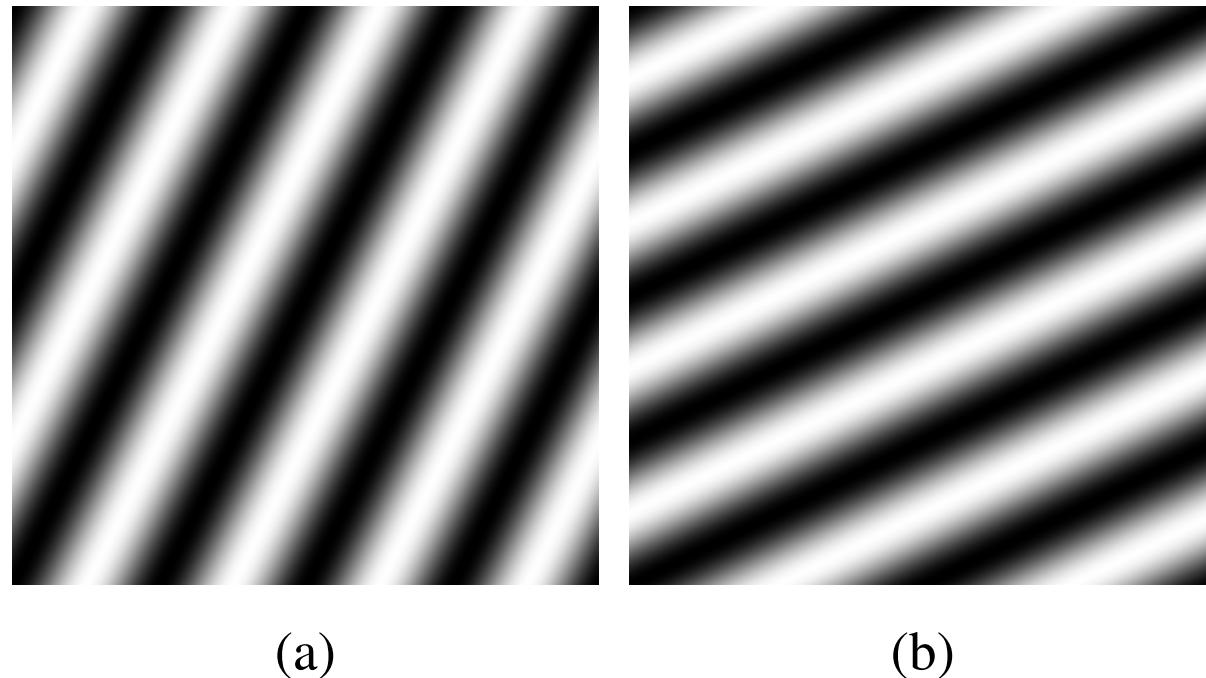
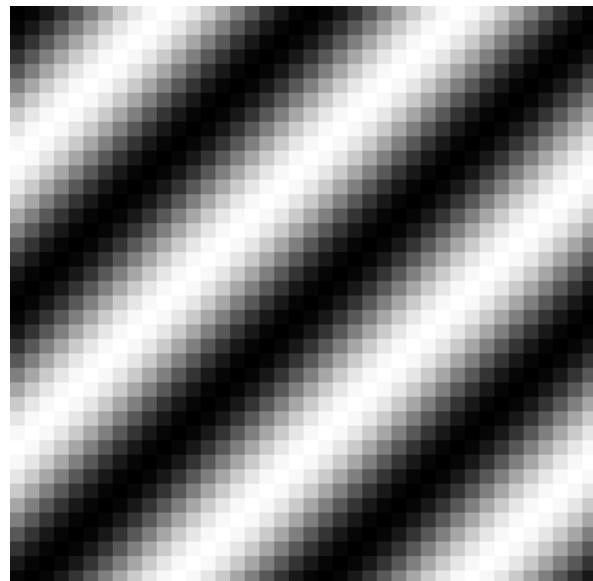
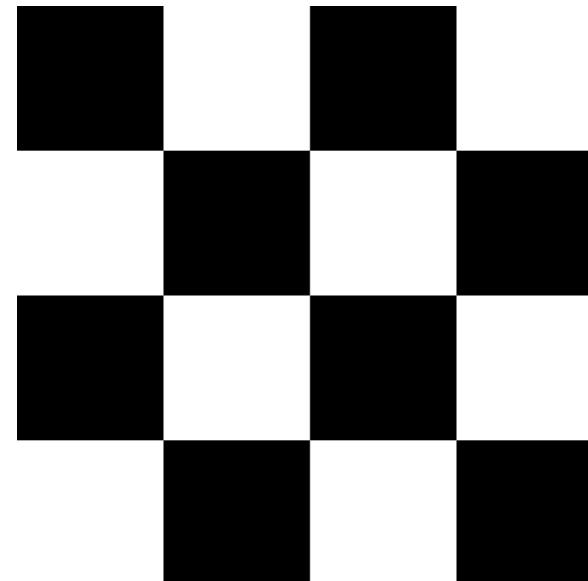


Fig. 2.20: Change of the wavenumber vectors $\{K_x, K_y\}$ (a) $k_x > k_y$ (b) $k_x < k_y$.

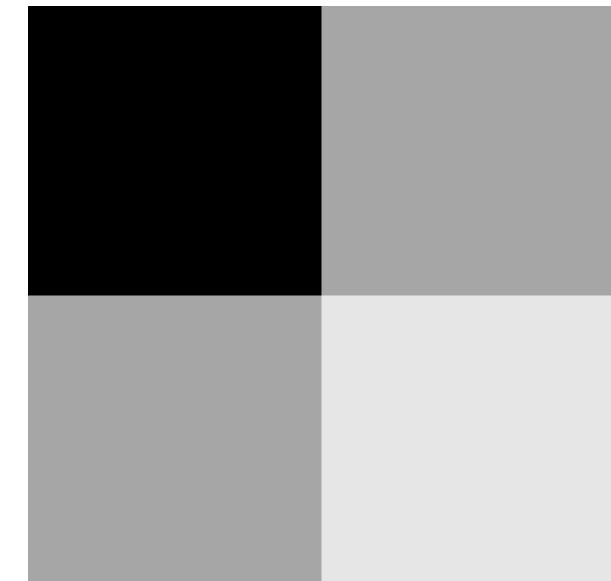
* The periodic function $f(x, y) = 0 + a \cdot \cos(2\pi (k_x x + k_y y) - \phi)$



(a)



(b)



(b)

Fig. 2.21: Change of the sampling interval r (a) $r < \frac{1}{2}\lambda$ (b) $r = \frac{1}{2}\lambda$ (c) $r > \frac{1}{2}\lambda$.

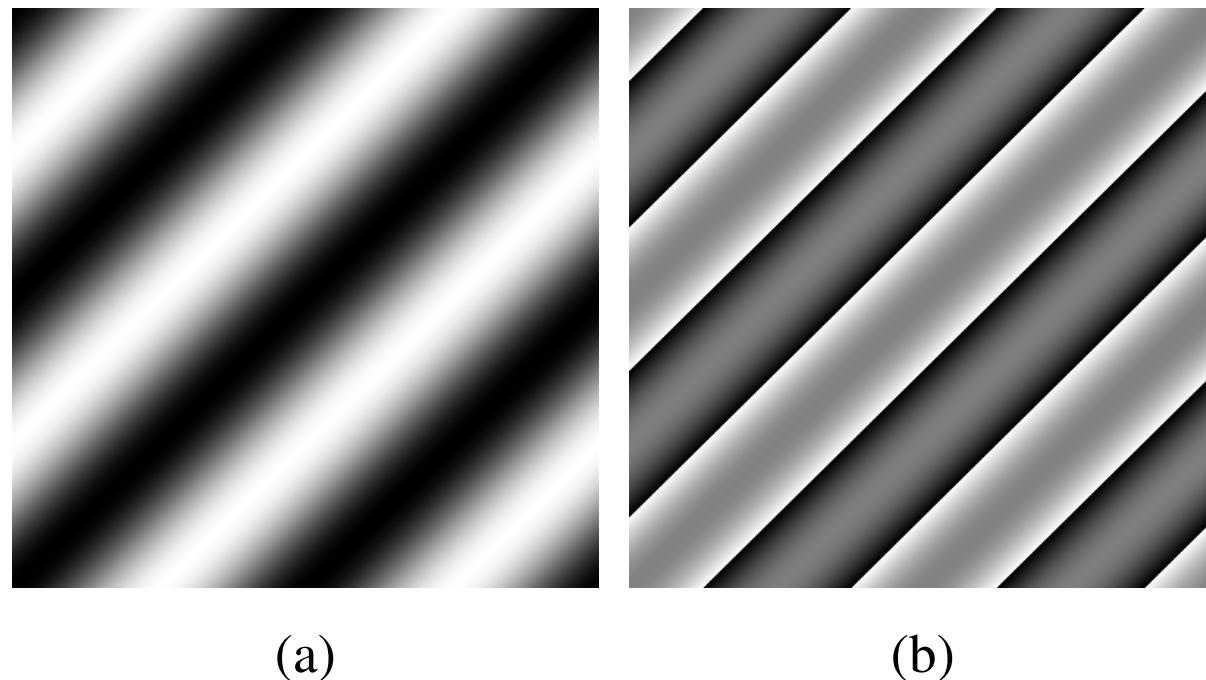


Fig. 2.22: A DC shift is required for the visualization of the function. (a) $a = 127.5$ (b) $a = 0$.

3 Image Enhancement in the Spatial Domain

Contents

3.1 Basics	3-2
3.2 Gray-Level Transformation Functions	3-3
3.2.1 Linear Transformations	3-8
3.2.2 Logarithmic Transformations	3-9
3.2.3 Power-Law Transformations	3-11
3.2.4 Piecewise-Linear Functions	3-15
Contrast stretching	3-15
Gray-level slicing	3-17
Bit-plane slicing	3-18
3.3 Histogram Processing	3-20
3.3.1 Histogram Equalization	3-23
3.3.2 Local Enhancement	3-43
3.3.3 Using Histogram Statistics for Image Enhancement	3-44

3.4 Spatial Filtering	3-52
3.5 Smoothing Spatial Filters	3-60
3.5.1 Linear Smoothing Filters	3-61
3.5.2 Ranking Filters	3-75
3.6 Sharpening spatial filters	3-79
3.6.1 Differentiation of Images	3-80
3.6.2 Second-Order Derivatives - The Laplacian	3-85
3.6.3 First-Order Derivatives - The Gradient	3-100

Image enhancement: Processing of an image so that the result is more suitable than the original image for a specific application.

Two broad categories of image enhancement approaches:

- Spatial domain methods (see chapter 3)
- Frequency domain methods (see chapter 4)

The term ‘spatial domain’:

- refers to the image plane itself and
- all methods in this category are based on direct manipulation of the pixels in an image.

3.1 Basics

Spatial domain methods are denoted by

$$g(x, y) = T [f(x, y)] \quad (3.1)$$

where

$f(x, y)$: input image

$g(x, y)$: processed image

T : operator on f , defined over some neighborhood of (x, y)

Neighborhood:

- Typically: rectangular subimage area centered about a point (x, y) (see fig. 3.1)
- Other neighborhood shapes, e. g. approximations to a circle, are possible, but seldom used.

Important: The operator is only applied to pixels in the area of the image spanned by the neighborhood.

3.2 Gray-Level Transformation Functions

Simplest case of a neighborhood:

The neighborhood is of size 1 x 1 (single pixel).

→ T becomes a **gray-level** (also called **intensity** or **mapping**) **transformation function**

$$s = T(r) \quad (3.2)$$

where $r = f(x, y)$ and $s = g(x, y)$

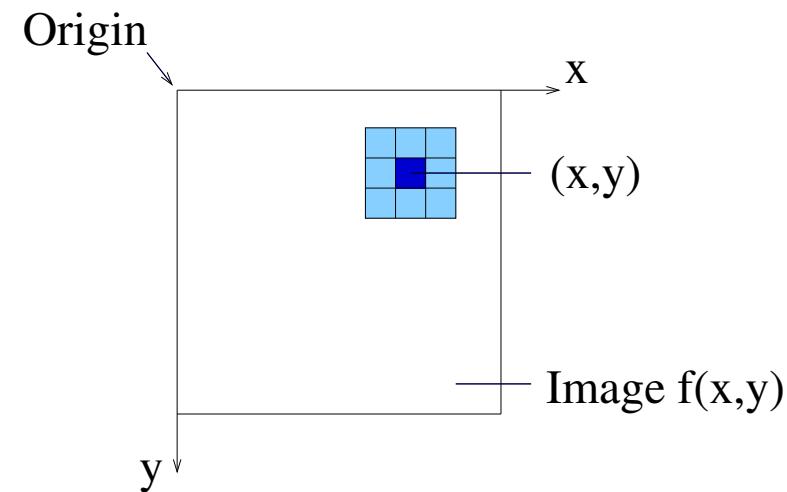
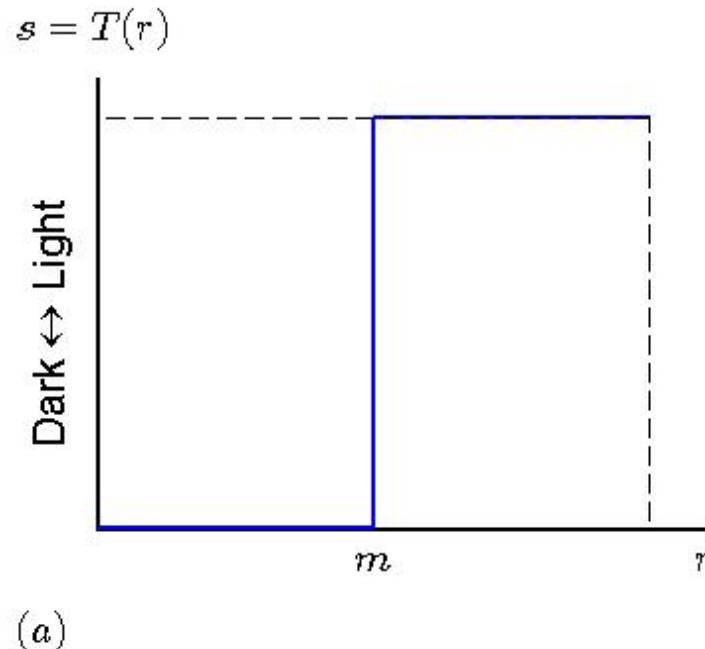
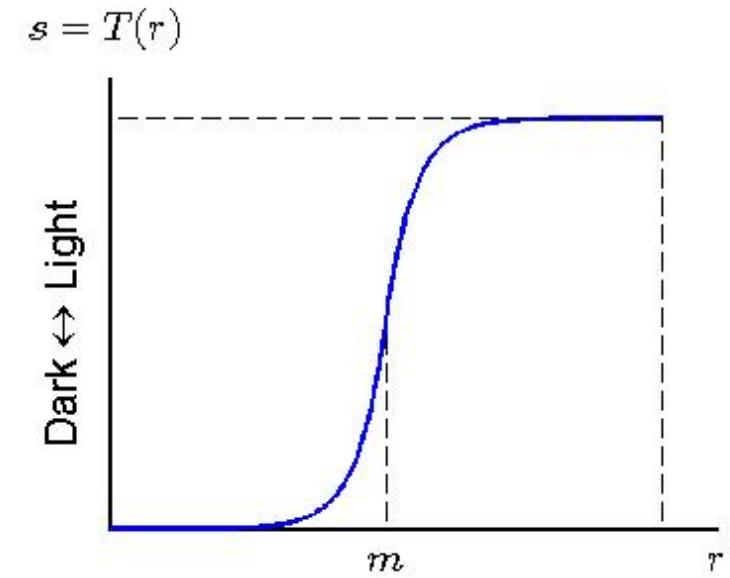


Fig. 3.1: 3×3 neighborhood about a point (x, y) in an image.

Fig. 3.2 shows gray-level transformation functions:



(a)



(b)

Fig. 3.2: Gray-level transformation functions for (a) **thresholding** and (b) **contrast enhancement**.

Hint: The techniques are also called **point processing**.

More basic grey-level transformation functions are depicted in fig. 3.3.

The values of the transformation function is typically stored in a one-dimensional array.

The mappings from r to s are implemented via **table lookups**.

Example 3.1 8-bit image

The corresponding lookup table containing the values of T will have 256 entries.

*

Three basic types of functions for image enhancement:

- linear (negative and identity transformations),
- logarithmic (log and inverse-log transformations), and
- power-law (n th power and n th root transformations).

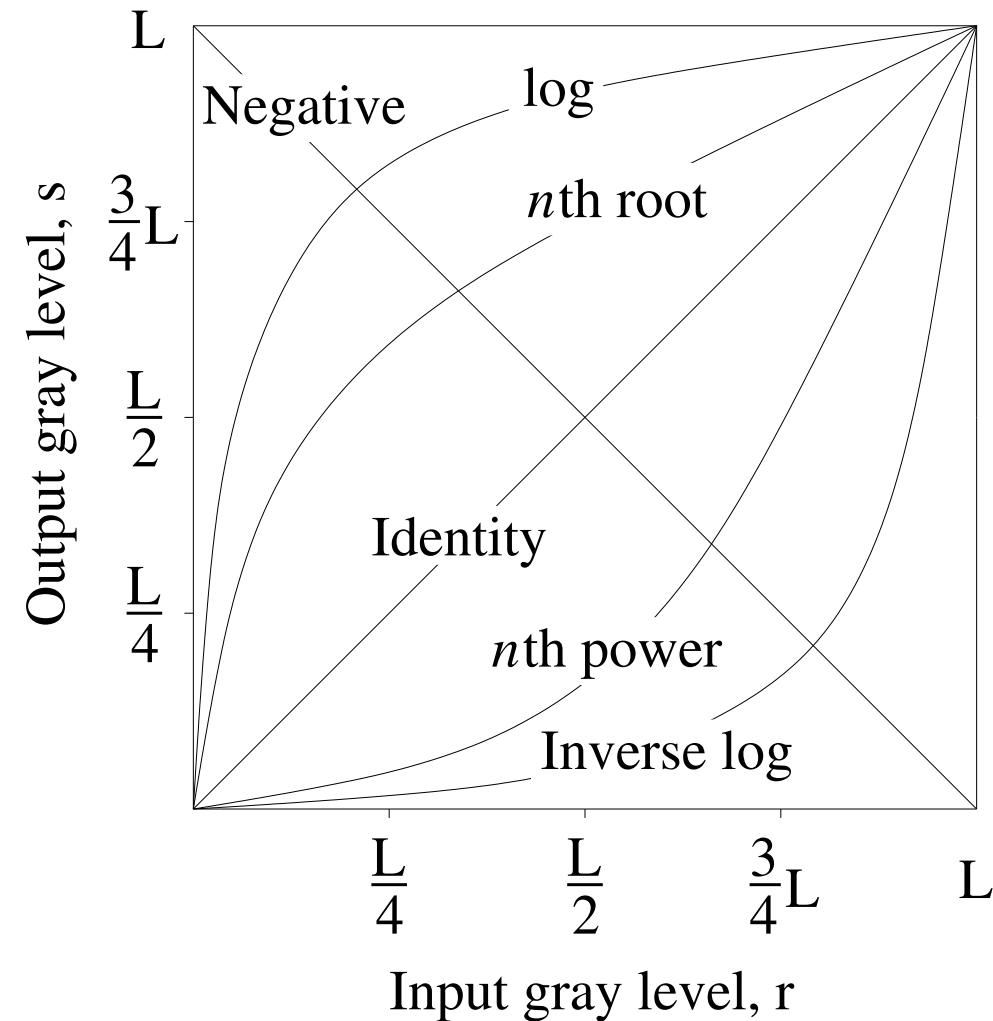


Fig. 3.3: Some basic gray-level transformation functions.

3.2.1 Linear Transformations

Assuming an image with gray levels in the range $[0, L-1]$, its negative is obtained by the operation

$$s = L - 1 - r \quad (3.3)$$



(a)



(b)

Fig. 3.4: (a) Original image. (b) Negative image obtained using the transformation in eq. (3.3).

3.2.2 Logarithmic Transformations

General form of the logarithmic transformation:

$$s = c \log(1 + r) \quad (3.4)$$

where

c : constant and

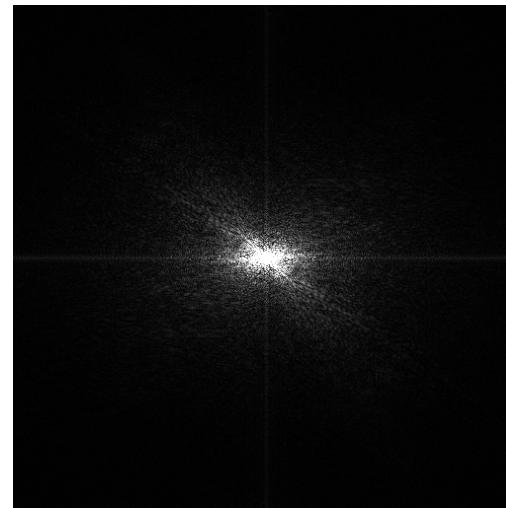
$$r \geq 0$$

- Notice:**
- The log transformation expands the values of the dark pixels, while compressing the higher level values.
 - The opposite holds for the inverse log transformation.

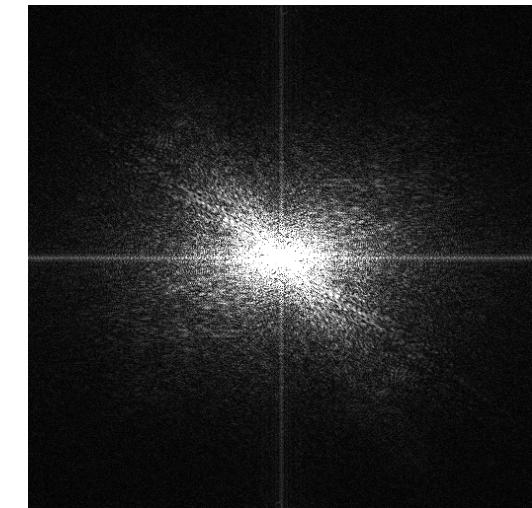
Classical application: Mapping of a Fourier Spectrum (compare sec. 4).



(a)



(b)



(c)

Fig. 3.5: (a) Original image. (b) Corresponding Fourier Spectrum. (c) Result of applying the log transformation given in eq. (3.4) with $c = 1$.

3.2.3 Power-Law Transformations

General form of the transformations:

$$s = c r^\gamma \quad (3.5)$$

where c, γ : positive constants

Plots of s versus r for various values of γ are shown in Fig. 3.6.

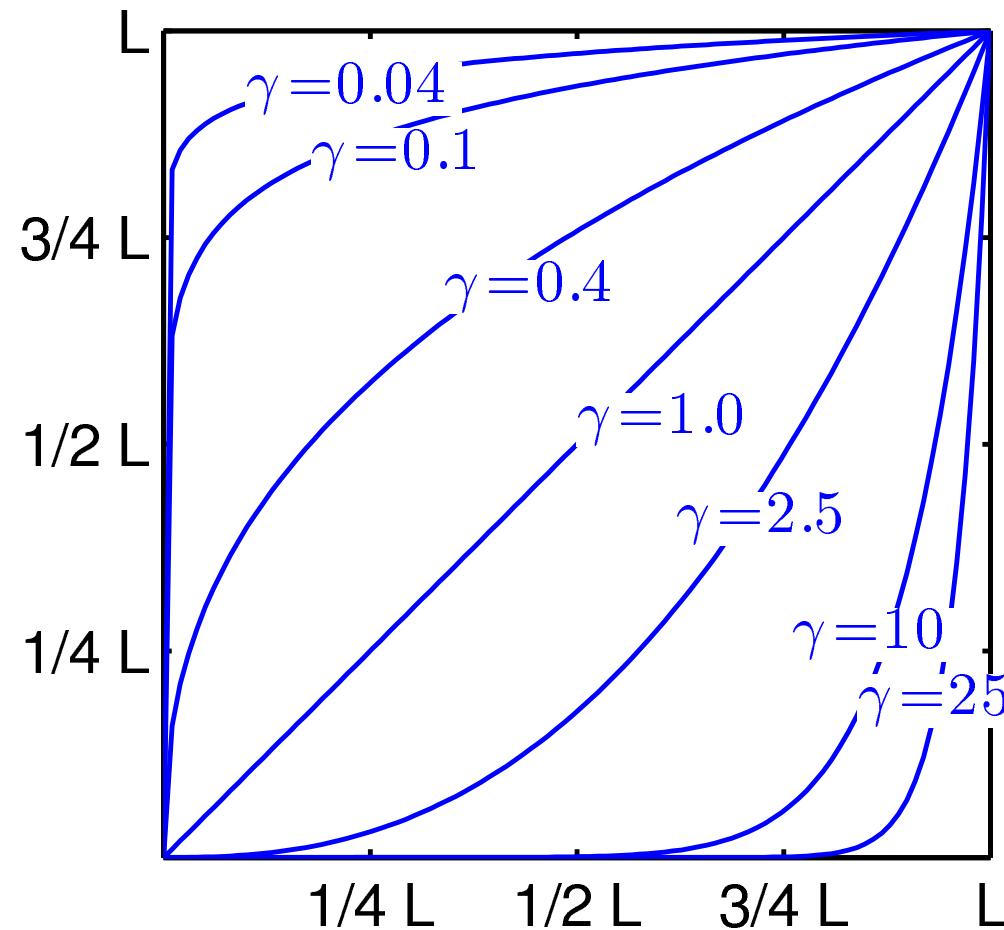


Fig. 3.6: Plots of power-law transformations for various values of γ ($c = 1$ in all cases).

Hint: Various devices, as e. g. frame grabbers, printers and displays respond according to a power law.

The exponent in eq. (3.5) is referred to as **gamma**.

The process to correct these phenomena is called **gamma correction**.

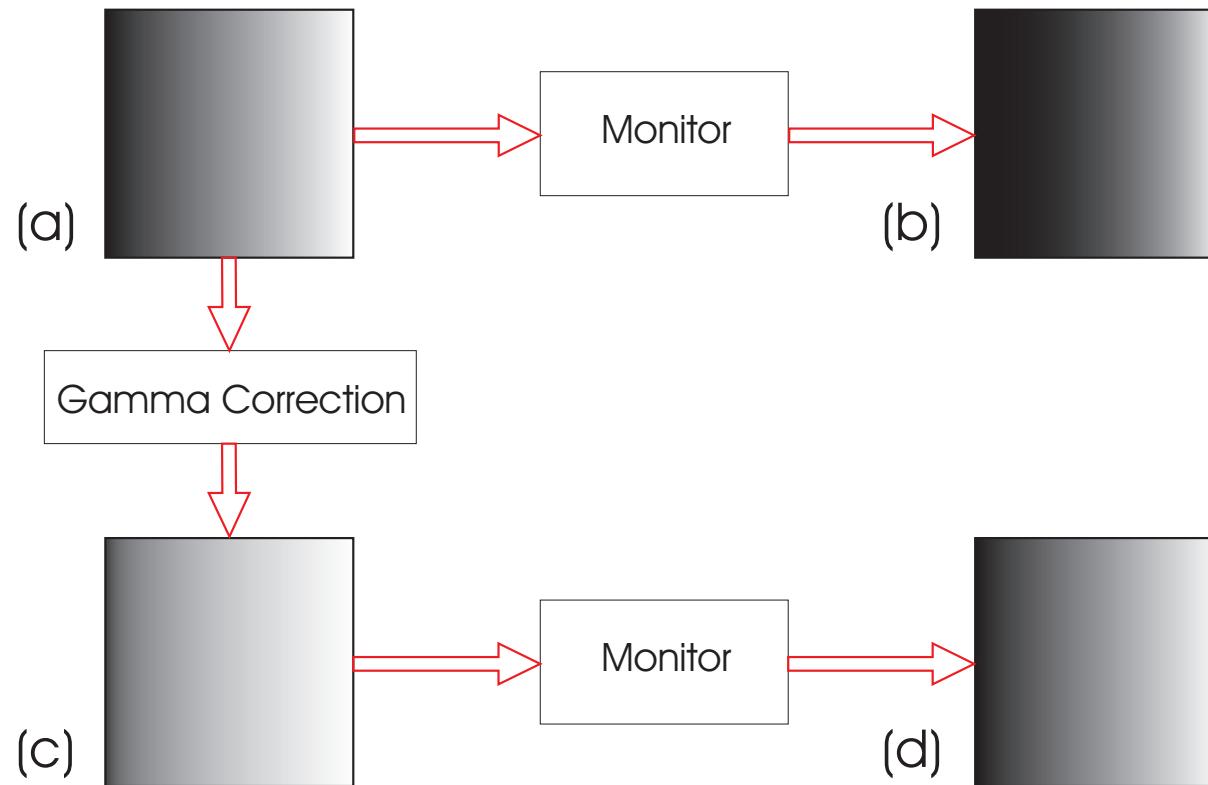


Fig. 3.7: Gamma correction of a nonlinear intensity-to-voltage response of a cathode ray tube (CRT). (a) Original image (linear-wedge gray-scale image). (b) Response of monitor to linear wedge ($s_n = r^{2,5}$). (c) Gamma-corrected wedge ($s = r^{\frac{1}{2,5}} = r^{0,4}$). (d) Output of monitor.

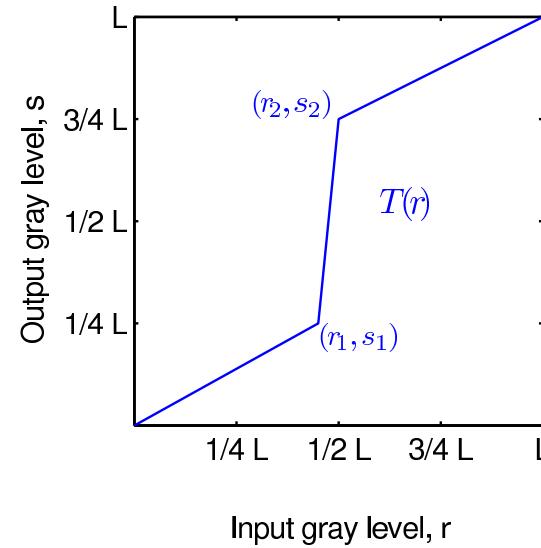
3.2.4 Piecewise-Linear Functions

Gray level transformations can also be realized as piecewise-linear transformation functions.

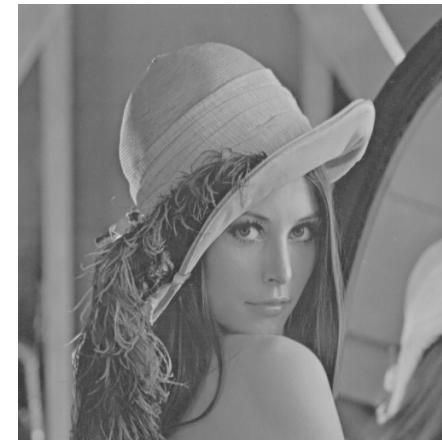
bigskip

Contrast stretching

This simple method is illustrated in fig. 3.8.



(a)



(b)



(c)



(d)

Fig. 3.8: Contrast stretching by use of a piecewise-linear transformation function. (a) Form of transformation. (From [GW18]) (b) A low contrast image. (c) Result of contrast stretching. (d) Result of thresholding.

Gray-level slicing

A specific range of gray levels in an image is highlighted. Sometimes called **Intensity-level slicing**.

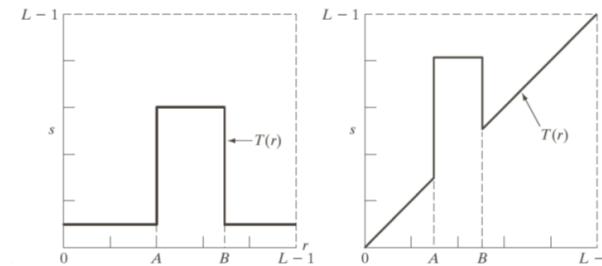


Fig. 3.9: a) Transformation highlights the range $[A,B]$, while in the transformation in (b) highlights the range $[A,B]$ and preserves other intensities. (From [GW18])

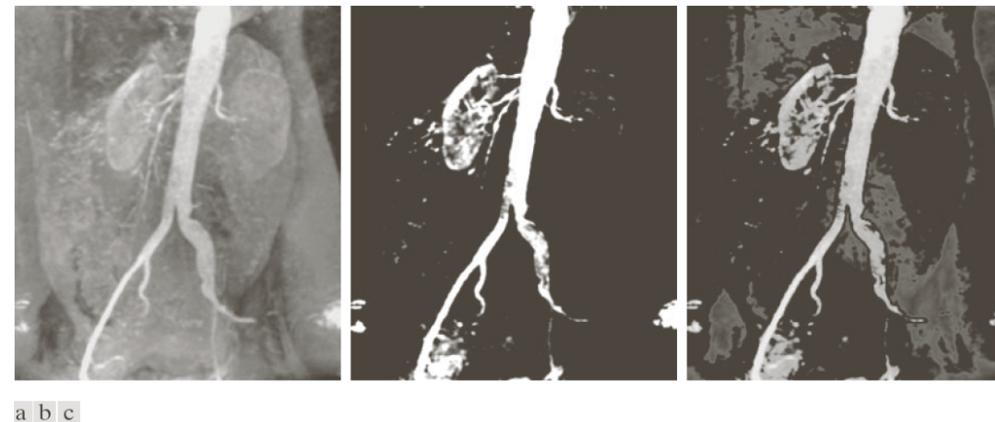


Fig. 3.10: (a) Aortic angiogram. (b) Result from transform a and (c) from transform b in the figure above. (From [GW18])

Bit-plane slicing

Highlighting the contribution made to total image appearance by specific bits (see fig. 3.11).

Useful technique for **image compression** (see section 6.3.2).

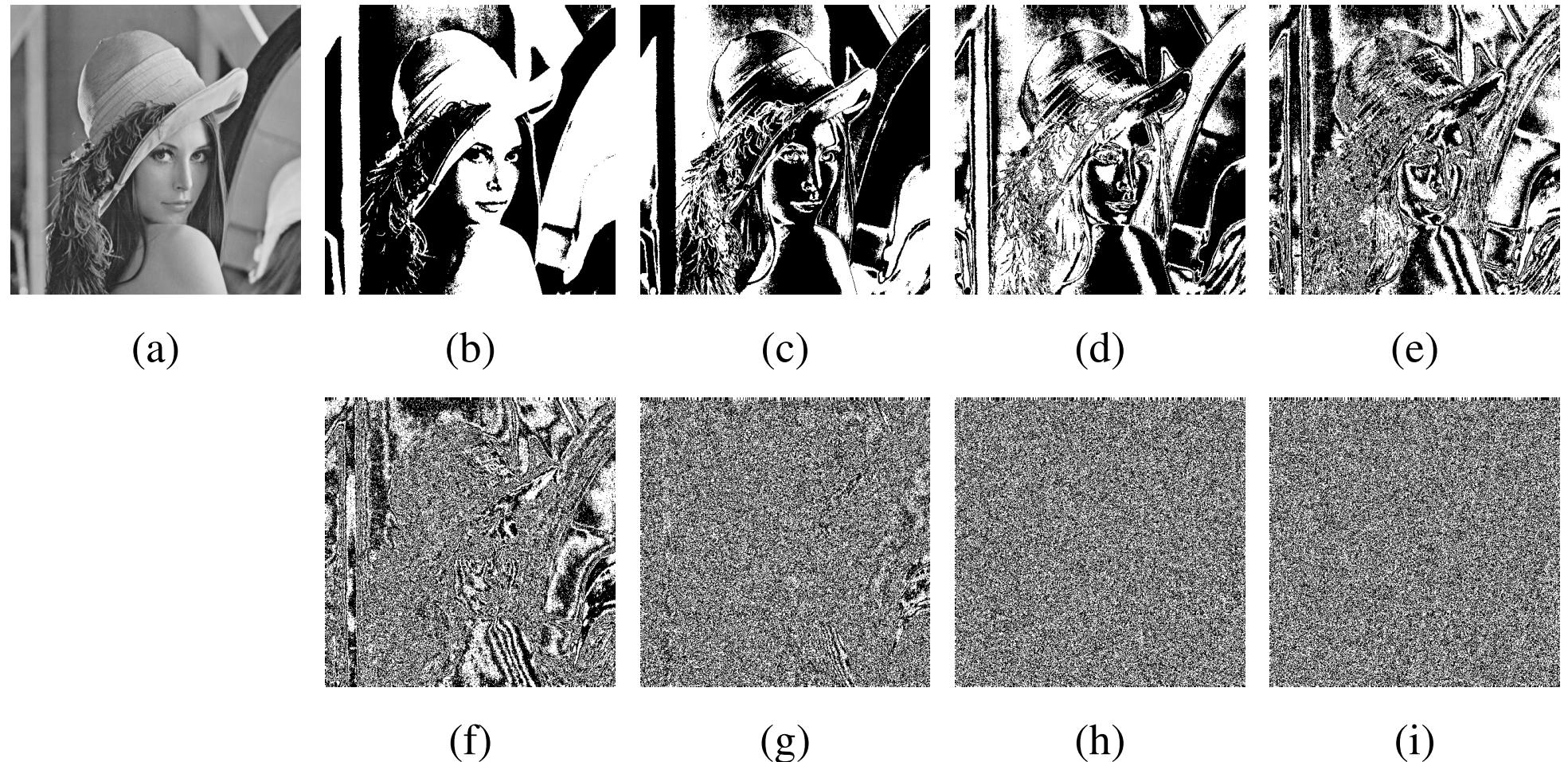


Fig. 3.11: (a) Original image. (b) to (i): bit planes 7-0 of the image in (a). It is obvious that the bit planes 0 and 1 contain nearly no image information. They can be neglected.

3.3 Histogram Processing

Histogram: Discrete function in the range [0, L-1]

$$h(r_k) = n_k \quad (3.6)$$

where

r_k : kth gray level

n_k : number of pixels having the gray level r_k

Hint: Histograms are often normalized by dividing each of its values by the total number of pixels ($M N$) in the image.

Normalized histogram¹:

$$p(r_k) = \frac{n_k}{M N} \quad (3.7)$$

for $k = 0, 1, 2, \dots, L - 1$

¹ $p(r_k)$: estimate of the probability of occurrence of intensity level r_k in an image.

Applications of histograms:

- Useful image statistics,
- Image compression, and
- Image segmentation.

Advantages of histograms:

- Easy to calculate in software
- Economic hardware implementations possible
→ popular tool for real-time image processing.

Fig. 3.12 shows histograms for four basic gray-level characteristics².

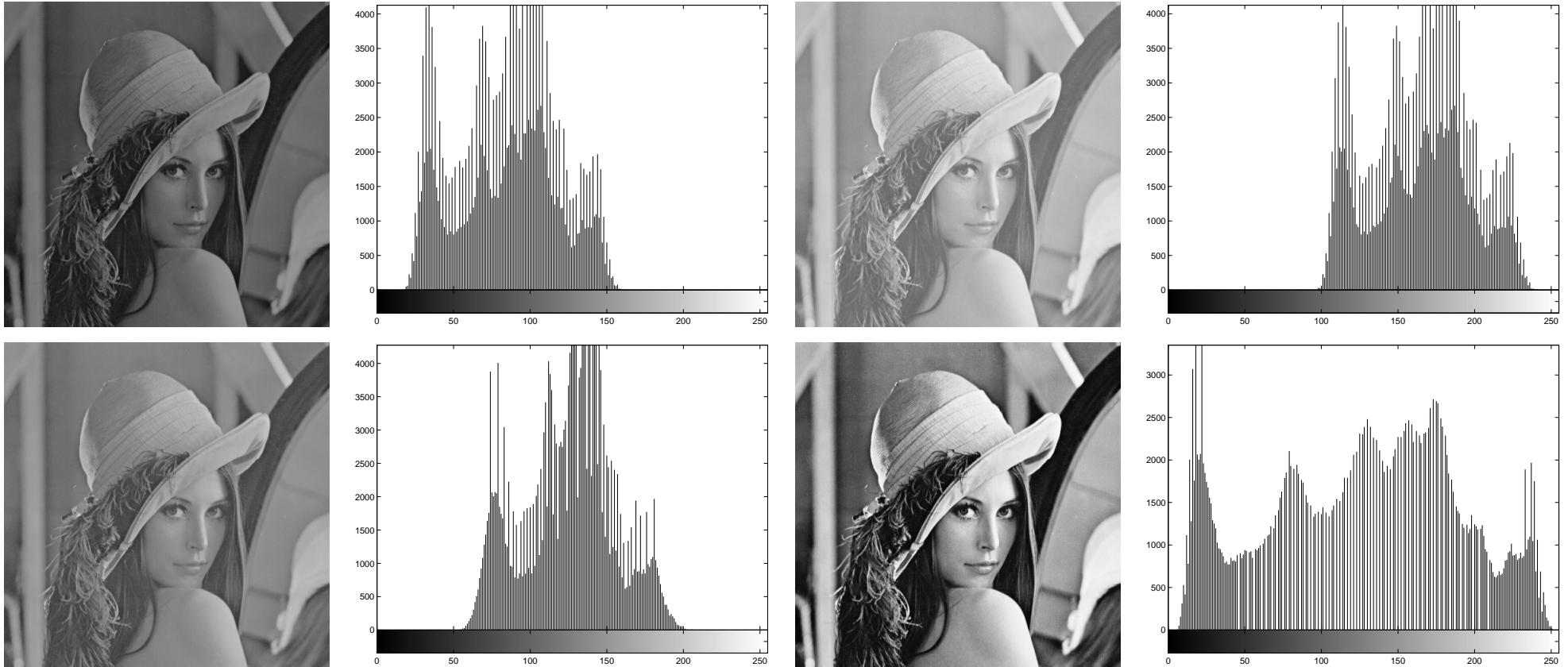


Fig. 3.12: Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

² Histograms may be viewed graphically as plots of $h(r_K) = n_k$ versus r_k .

3.3.1 Histogram Equalization

Histogram equalization allows a considerable enhancement of image quality.

Assumptions:

- Continuous intensity values
- r represents the gray levels of an image
- r has been normalized to the interval $[0, L-1]$, with
 - $r = 0 \rightarrow$ black
 - $r = L - 1 \rightarrow$ white

- Intensity mapping T according to eq. (3.2):

$$s = T(r); 0 \leq r \leq L - 1 \quad (3.8)$$

$T(r)$ should satisfy the following assumptions:

1. $T(r)$ is monotonically³, ⁴ increasing in the interval $0 \leq r \leq L - 1$; and
2. $0 \leq T(r) \leq L - 1$ for $0 \leq r \leq L - 1$.

Hint: • Assumption (1) prevents artifacts.

• Assumption (1') is required to ensure that the inverse transformation will exist.

$$r = T^{-1}(s); 0 \leq s \leq L - 1 \quad (3.9)$$

- Assumption (2) generates that the output gray levels will be in the same range as the input levels.

³ Monotonically increasing: $T(r_2) \geq T(r_1)$ for $r_2 \geq r_1$.

⁴ Required in some cases: Assumption (1'): $T(r)$ is a strictly monotonically increasing function: $T(r_2) > T(r_1)$ for $r_2 > r_1$.

Fig. 3.13 gives an example of a transformation which complies with the assumptions.

Example 3.2 Strictly Monotonically Increasing Gray-Level Transformation Function.

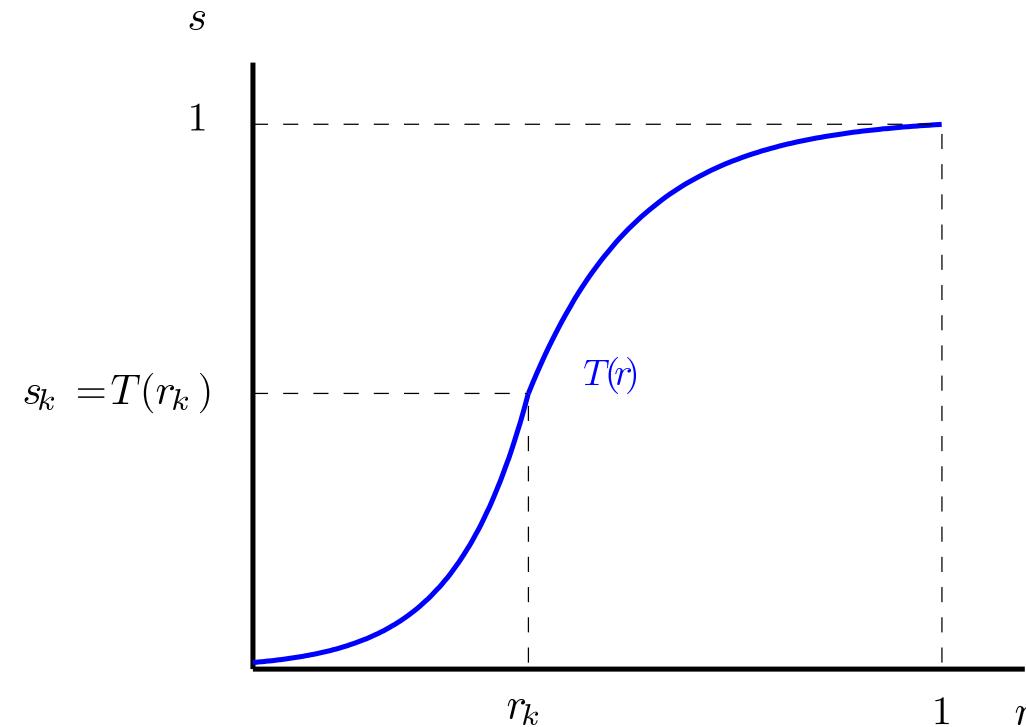


Fig. 3.13:

New idea: Gray levels of an image can be interpreted as **random variables** in the interval $[0, L-1]$.

Exkurs 3.1 Statistics in DIP

Interpretation of digital image processing as a process of measuring statistical values which is distorted by noise and systematic errors.

Numerous statistical values in computer vision:

- The image sensor superimposes the measured radiance with electrical noise.
- When the illumination intensity is low, no continuous light flux, but single photons arriving in random intervals are registered.
- The observed process or the observed object can reveal statistical features.

Two error categories:

Statistical error: This error describes the distribution of the measured values when repeating the process of measurement (see fig. 3.14(a)).

Systematic error: It is often caused by unknown or uncontrollable parameters influencing the process of measurement (e. g. calibration error, see fig 3.14(b)).

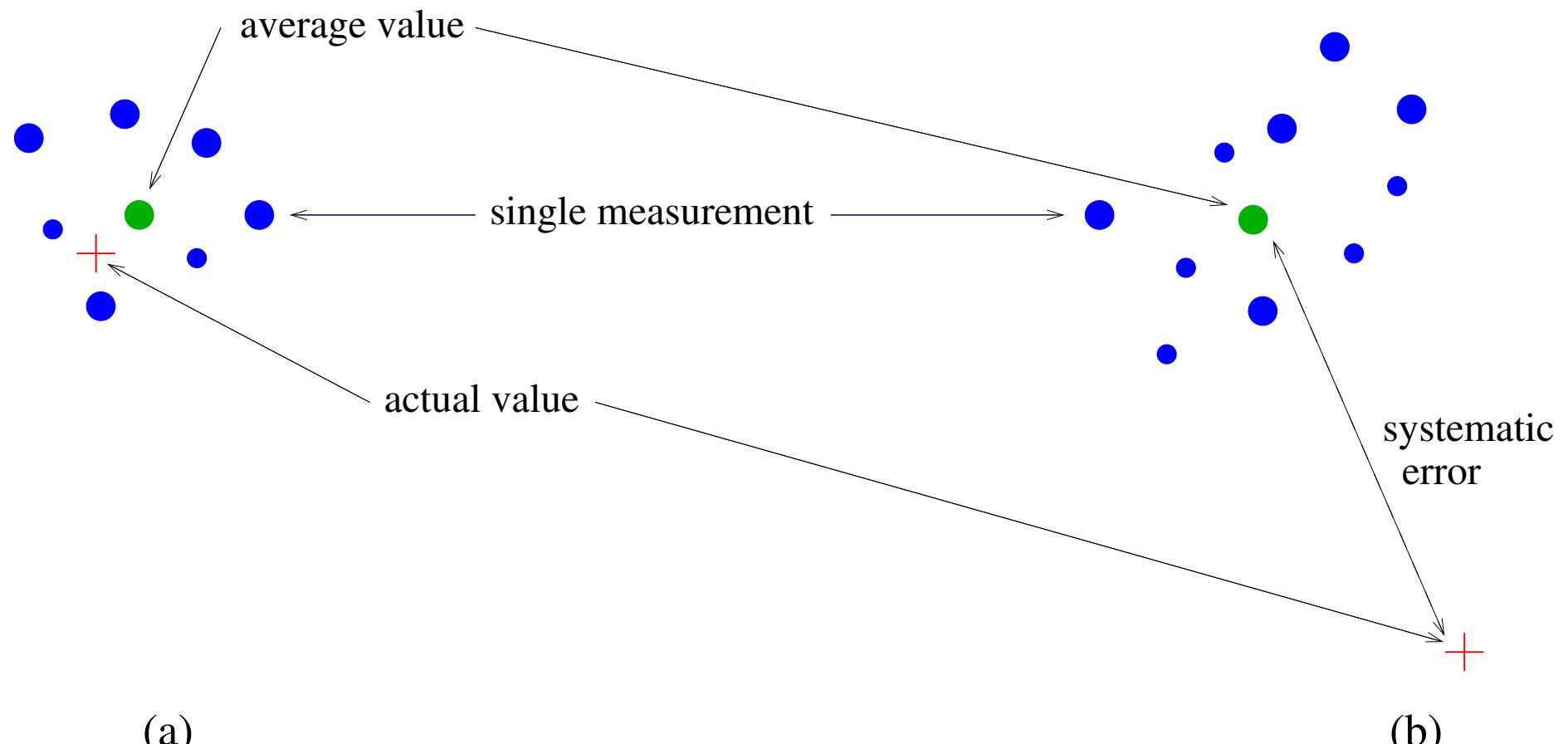


Fig. 3.14: Illustration of (a) statistical and (b) systematic errors.

Fundamental descriptor of a random variable: Its **probability density function (PDF)**.

Now:

- $p_r(r)$: Probability density function of the input image and
- $p_s(s)$: Probability density function of the output image⁵.

If

- $p_r(r)$ and $T(r)$ are known and
- $T(r)$ is continuous and differentiable over range of gray level values,

the PDF of the transformed image can be calculated using

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right|. \quad (3.10)$$

⁵ The subscripts s and r indicate that p_r and p_s are different functions.

Particular important transformation function:

$$s = T(r) = (L - 1) \int_0^r p_r(w) dw \quad (3.11)$$

where w: dummy variable of integration

Right side of eq.:

Cumulative distribution function (CDF) of random variable r.

Note: Function (3.11) satisfies the assumptions (1) and (2):

- Assumption (1) is satisfied (function is both single valued and monotonically increasing) because PDFs are always positive and the integral of a function is the area under the function.
- When upper limit in this equation is $r = (L - 1)$, the integral evaluates to 1⁶ and assumption (2) is satisfied also.

⁶ Area under a PDF curve always is 1.

Calculating the derivative of eq. (3.10) with respect to its upper limit yields:

$$\begin{aligned}\frac{ds}{dr} &= \frac{d(T(r))}{dr} \\ &= (L - 1) \frac{d}{dr} \left(\int_0^r p_r(w) dw \right) \\ &= (L - 1) p_r(r)\end{aligned}\tag{3.12}$$

Substituting eq. (3.12) into eq. (3.10):

$$\begin{aligned} p_s(s) &= p_r(r) \left| \frac{dr}{ds} \right| \\ &= p_r(r) \left| \frac{1}{(L-1)p_r(r)} \right| \\ \text{Note: Probability values are always positive.} \\ &= \frac{1}{L-1} \quad 0 \leq s \leq L-1 \tag{3.13} \\ \rightarrow \text{uniform probability density function.} \end{aligned}$$

Interpretation: Eq. (3.13) shows that the resulting $p_s(s)$ is always uniform, independently of the form of $p_r(r)$.

Fig. 3.15 illustrates these concepts.

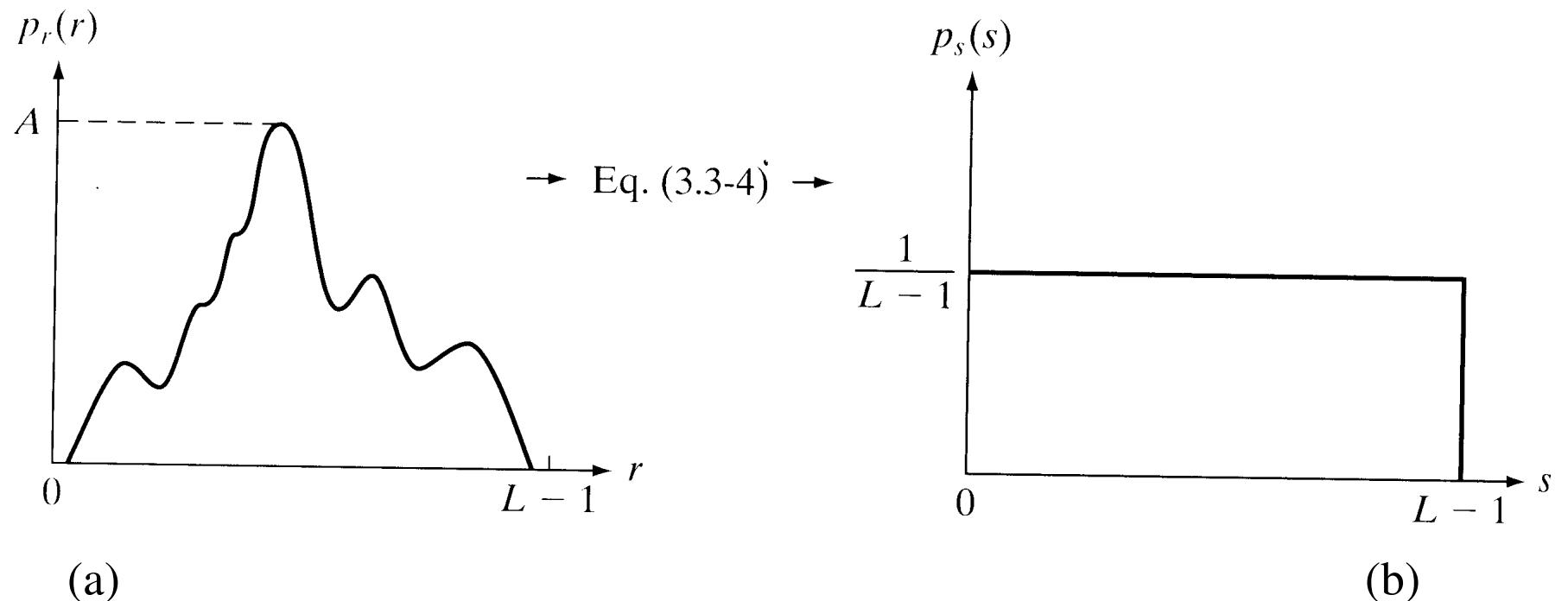


Fig. 3.15: (a) Arbitrary PDF. (b) Result of applying the transformation in eq. (3.11) to all intensity levels r .
(From [GW18])

Example 3.3 Histogram Equalization: Continuous Functions

Given: PDF of the continuous intensity values in an image:

$$p_r(r) = \begin{cases} \frac{2r}{(L-1)^2} & \text{for } 0 \leq r \leq L-1 \\ 0 & \text{otherwise} \end{cases}$$

From eq. (3.11):

$$\begin{aligned} s = T(r) &= (L-1) \int_0^r p_r(r) dr \\ &= \frac{2}{L-1} \int_0^r r dr \\ &= \frac{r^2}{L-1} \end{aligned}$$

- Assumption:**
- Image with $L = 10$
 - Pixel at location (x,y): Intensity $r(x, y) = 3$

$$\rightarrow s = T(r) = \frac{r^2}{9} = 1$$

Verification:

$$\begin{aligned}
 p_s(s) &= p_r(r) \left| \frac{dr}{ds} \right| = \frac{2r}{(L-1)^2} \left| \left[\frac{ds}{dr} \right]^{-1} \right| \\
 &= \frac{2r}{(L-1)^2} \left| \left[\frac{d}{dr} \frac{r^2}{L-1} \right]^{-1} \right| \\
 &= \frac{2r}{(L-1)^2} \left| \frac{L-1}{2r} \right| \\
 &= \frac{1}{L-1}
 \end{aligned}$$

Now: Return to discrete values and functions

→ Use of probabilities (histogram values) and summations instead of probability density functions and integrals.

From eq. (3.7):

$$p(r_k) = \frac{r_k}{MN} \quad k = 0, 1, 2, \dots, L - 1$$

Discrete formulation of eq. (3.11):

$$\begin{aligned} s_k &= T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j) \\ &= \frac{L - 1}{MN} \sum_{j=0}^k n_j \quad k = 0, 1, 2, \dots, L - 1 \end{aligned} \tag{3.14}$$

Interpretation: An output image is obtained by mapping each pixel with level r_k in the input image into a corresponding pixel with level s_k in the output image using eq. (3.14):

→ **histogram equalization or histogram linearization.**

Example 3.4 Histogram Equalization: Discrete Functions

Given: 3-bit image ($k=3$) of size 64×64 pixels ($MN = 4096$).

The intensity distribution is given in table 3.1:

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

Table 3.1: Intensity distribution and histogram values for an example image. (From [GW18])

The corresponding histogram is sketched in fig. 3.16(a).

Application of eq. (3.14) yields:

$$s_0 = T(r_0) = 7 \sum_{j=0}^0 p_r(r_j) = 7 p_r(r_0) = 1.33,$$

$$s_1 = T(r_1) = 7 \sum_{j=0}^1 p_r(r_j) = 7 p_r(r_0) + 7 p_r(r_1) = 3.08,$$

and $s_2 = 4.55$, $s_3 = 5.67$, $s_4 = 6.23$, $s_5 = 6.65$, $s_6 = 6.86$, and $s_7 = 7.00$.

The staircase shape of the transformation function is shown in fig. 3.16(b).

As the s values contain fractions, they have to be rounded to the nearest integer.

$$s_0 = 1.33 \rightarrow 1 \quad s_4 = 6.23 \rightarrow 6$$

$$s_1 = 3.08 \rightarrow 3 \quad s_5 = 6.65 \rightarrow 7$$

$$s_2 = 4.55 \rightarrow 5 \quad s_6 = 6.86 \rightarrow 7$$

$$s_3 = 5.67 \rightarrow 6 \quad s_7 = 7.00 \rightarrow 7$$

Note: There are only five distinct intensity values, resulting in only five values of $p_s(s)$.

Fig. 3.16(c) shows the equalized histogram.

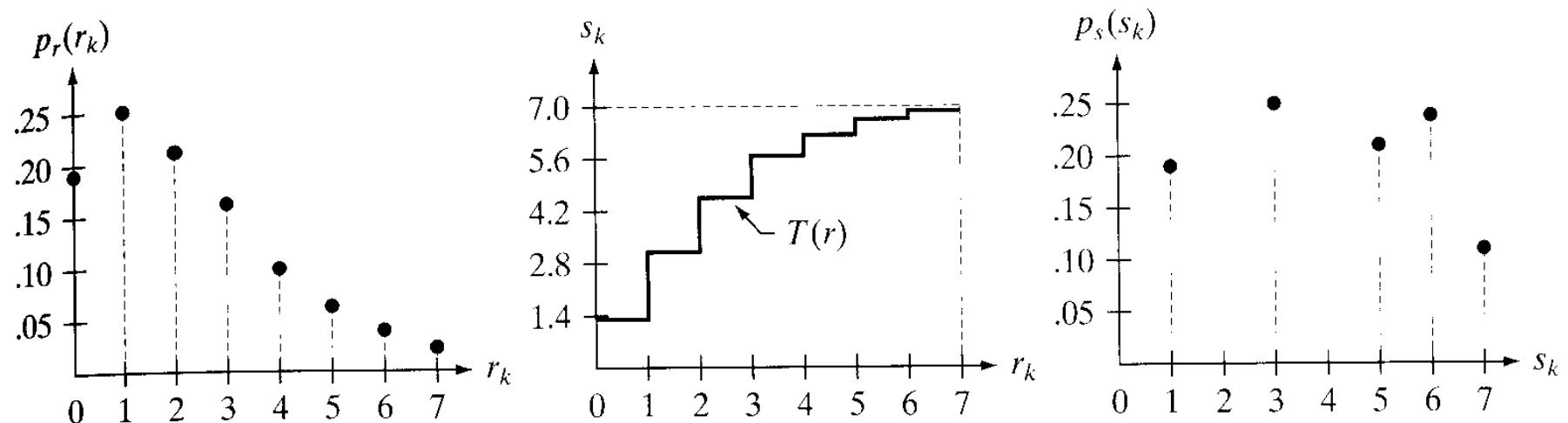
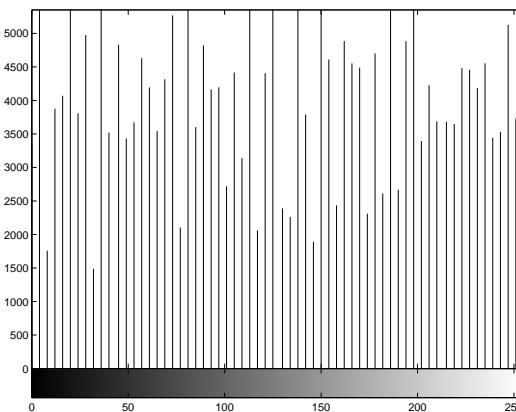
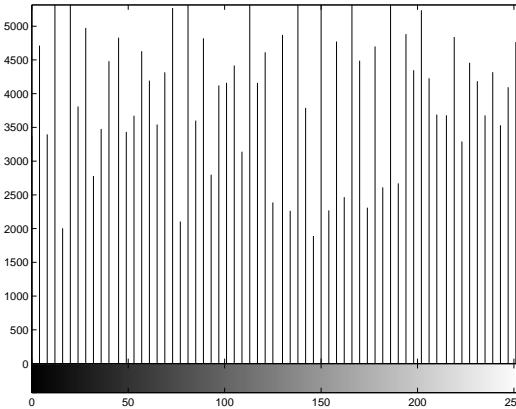
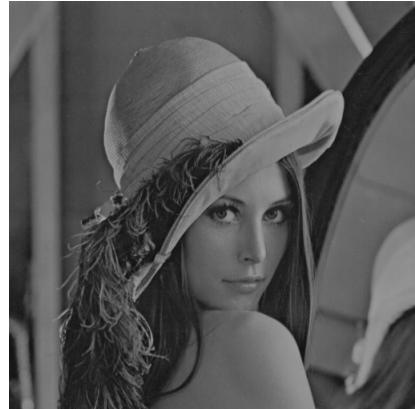


Fig. 3.16: Illustration of histogram equalization. (From [GW18])

Interpretation: Due to quantization effects perfectly flat histograms are rare in practical applications of histogram equalization.

Figs. 3.17(a) and 3.18(a) show the four images from fig. 3.12, and figs. 3.17(b) and 3.18(b) show the result of performing histogram equalization on each of these images.



(a)

(b)

(c)

Fig. 3.17: (a) Images from fig. 3.12. (b) Results of histogram equalization. (c) Corresponding histograms.

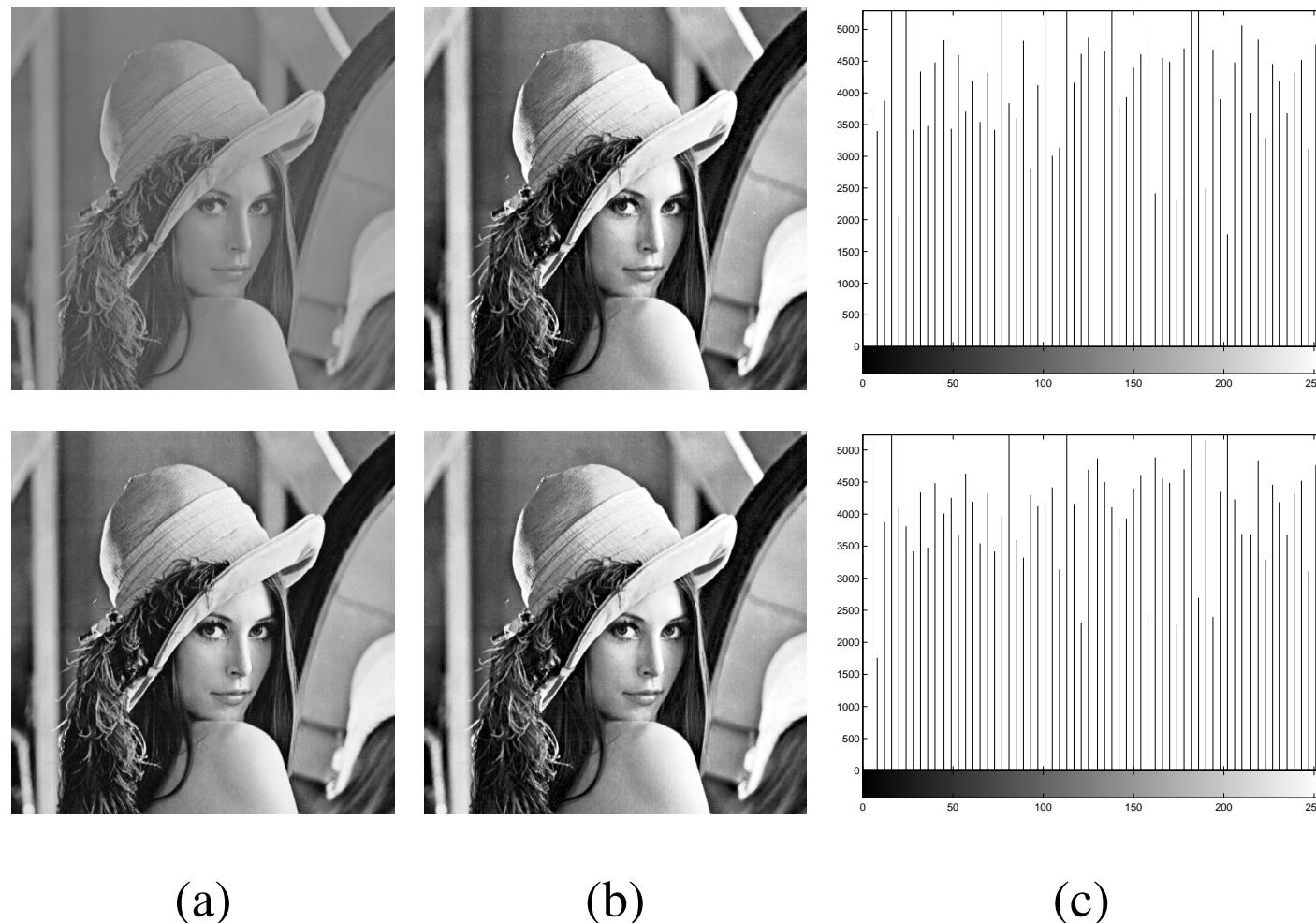


Fig. 3.18: (a) Images from fig. 3.12. (b) Results of histogram equalization. (c) Corresponding histograms.

3.3.2 Local Enhancement

While histogram equalization is a global histogram processing method, now an approach for the enhancement of areas in the image is described.

Idea: Histogram equalization applied to a square or rectangular neighborhood.

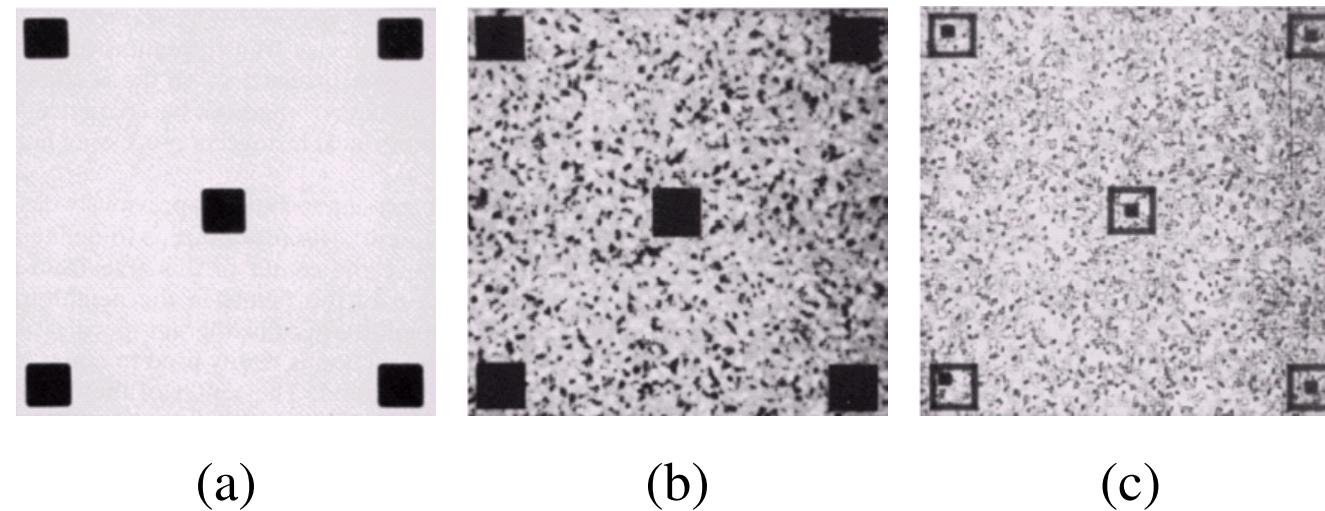


Fig. 3.19: (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using 7×7 neighborhood about each pixel. (From [GW18])

Advantages:

- Small image structures can be revealed.
- The enhancement of noise can be reduced.

3.3.3 Using Histogram Statistics for Image Enhancement

Instead of using the histogram directly, statistical parameters which can be computed from the histogram can be applied in order to enhance the image.

- Assumptions:**
- r is a discrete random variable representing gray levels in the range $[0, L - 1]$.
 - $p(r_i)$ denotes the normalized histogram component corresponding to the i th value of r .

Interpretation: $p(r_i)$: estimate of the probability of occurrence of gray-level r_i

Now: **Moments** will be introduced as new statistical parameters.

Definition: *n*th moment of r about its mean

$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i) \quad (3.15)$$

where the mean value m of r (average gray level) is

$$m = \sum_{i=0}^{L-1} r_i p(r_i) \quad (3.16)$$

With eq. (3.15) and (3.16) follow:

$$\mu_0(r) = 1$$

$$\mu_1(r) = 0$$

Second moment:

$$\mu_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i) \quad (3.17)$$

- Interpretation:**
- The second moment is the **variance** $\sigma^2(r)$ of r .
 - The square root of the variance is defined as **standard deviation**.

The impact of moments for image enhancement can be increased when calculating the first and second moment (according to eq. (3.16) and (3.17)) for a local regional subimage S_{xy} only.

Mean value of the pixels in S_{xy} :

$$m_{S_{xy}} = \sum_{(s,t) \in S_{xy}} r_{s,t} \cdot p(r_{s,t}) \quad (3.18)$$

Variance of the pixels in S_{xy} :

$$\sigma_{S_{xy}}^2 = \sum_{(s,t) \in S_{xy}} [r_{s,t} - m_{S_{xy}}]^2 p(r_{s,t}) \quad (3.19)$$

Example 3.5 Enhancement of Dark Background Structure

Fig. 3.20 shows an SEM image of a tungsten filament wrapped around a support.

A second filament can be detected on the right side of the image, which is much darker.

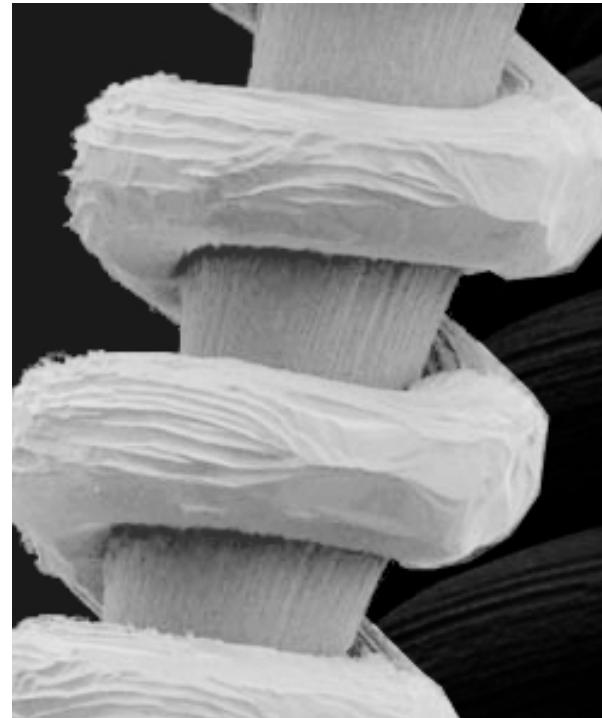


Fig. 3.20: SEM image of a tungsten filament and support, magnified approximately 130x. (From [GW18])

Idea: Comparison of the local average gray level $m_{S_{xy}}$ at a point (x, y) to the average image gray-level (global mean) M_G .

A pixel at point (x, y) is a candidate for enhancement, if

$$m_{S_{xy}} \leq k_0 M_G$$

where $k_0 \in Q^+$ and $k_0 \leq 1.0$

A pixel is furthermore a candidate if its local contrast is low. This can be derived by

$$\sigma_{S_{xy}} \leq k_2 D_G$$

where D_G : global standard deviation

$k_2 \in Q^+$ and

$k_2 < 1.0$ for dark areas

$k_2 \geq 1.0$ for area to be enhanced.

At last the lowest value of contrast has to be defined, which is still acceptable:

$$k_1 D_G \leq \sigma_{S_{xy}}$$

with $k_1 < k_2$

Summary of methods:

$$g(x, y) = \begin{cases} E \cdot f(x, y) & \text{if } m_{S_{xy}} \leq k_0 M_G \text{ and } k_1 D_G \leq \sigma_{S_{xy}} \leq k_2 \cdot D_G \\ f(x, y) & \text{otherwise} \end{cases}$$

The parameter for the next images are the following:

- $E = 4, 0$
- $k_0 = 0, 4$
- $k_1 = 0, 02$
- $k_2 = 0, 4$

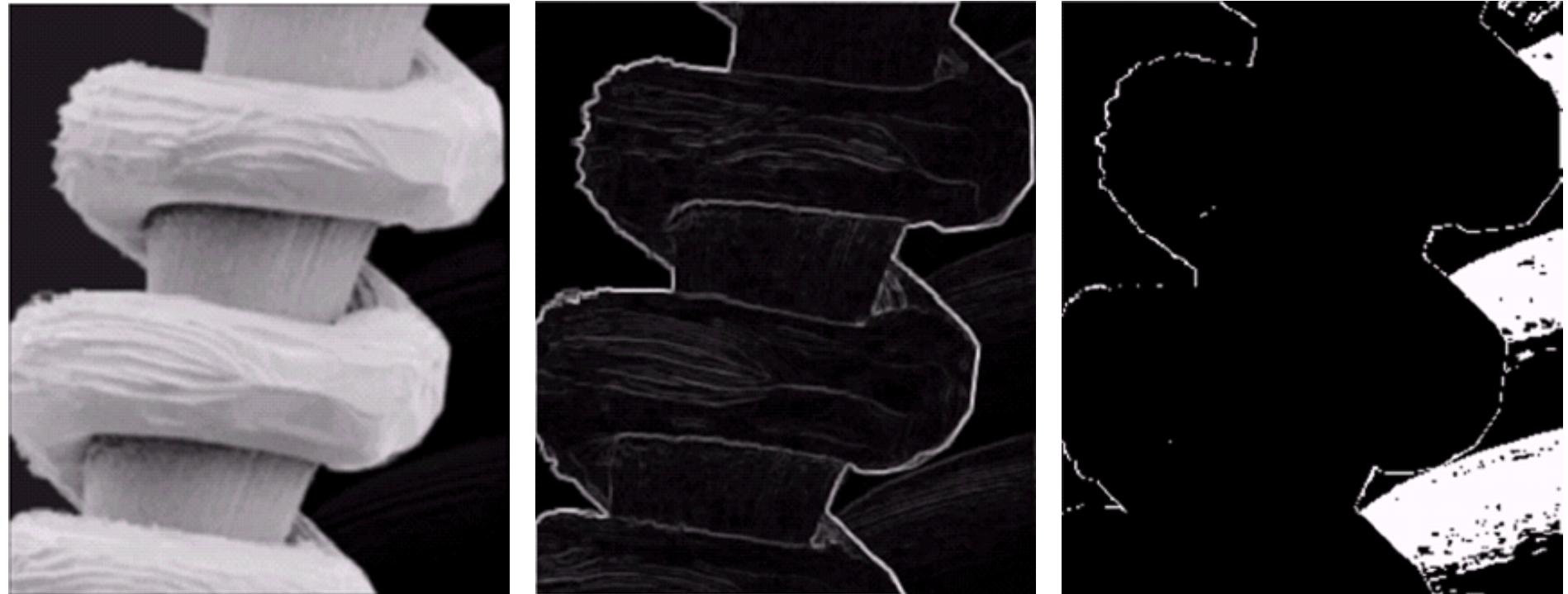


Fig. 3.21: Intermediate result while enhancing the image in fig. 3.20. (a) Image formed from all means using eq. (3.18). (b) Image formed from all standard deviations obtained from eq. (3.19). (c) Image formed from all multiplication constants used to produce the enhanced image shown in fig. 3.20. (From [GW18])

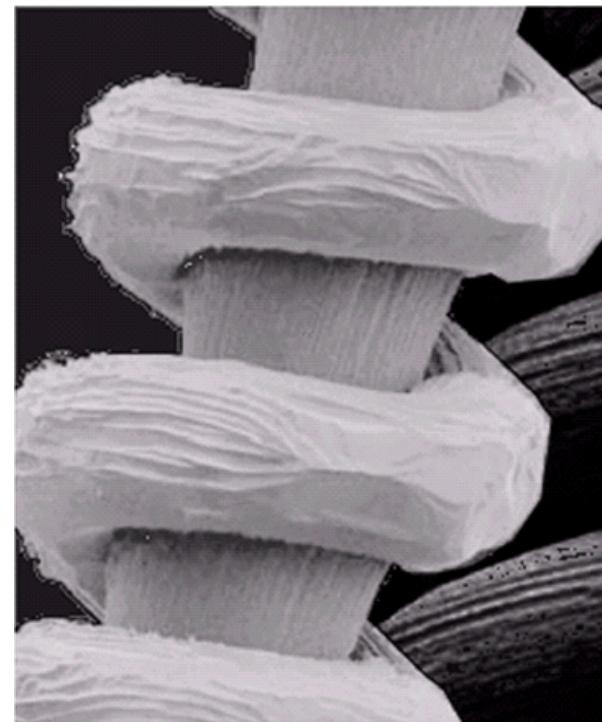


Fig. 3.22: Enhanced SEM image. (From [GW18])

*

3.4 Spatial Filtering

The gray-level transformation functions in Section 3.2 use the simplest case of a neighborhood: one pixel.

Larger neighborhoods allow considerably more flexibility.

General approach:

Use of a function of the values of f in a predefined neighborhood of (x, y) to determine the value of g at (x, y) .

One principal approach:

Use of so-called **masks** (also referred to as):

- **filters**,
- **kernels**,
- **templates**, or
- **windows**.

The values of the mask, referred to as coefficients rather than pixels, determine the nature of the process, e. g.

- Image smoothing (see Section 3.5) and
- Image sharpening (see Section 3.6).

Basically, a mask is a small 2D array, e. g. 3×3 as it is depicted in fig. 3.1.

Hint: The concept of filtering has its root in the use of the **Fourier transform** for signal processing in the frequency domain.

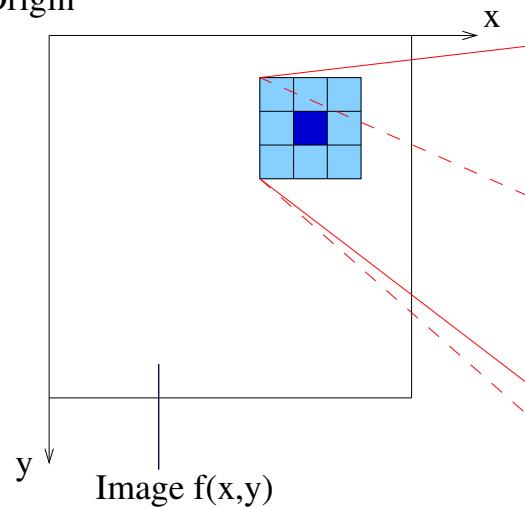
Here, the term **spatial filtering** is used to differentiate this type of process from the one in the frequency domain.

Mechanics of spatial filtering (see fig. 3.23)

- Simply moving the filter mask from point to point in an image.
- At each point (x, y) , the response of the filter application to the image is calculated.
- For **linear spatial filtering**:

The response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask.

Origin



$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	$w(0,0)$	$w(0,1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

Mask coefficients showing
coordinate arrangement

$f(x-1,y-1)$	$f(x-1,y)$	$f(x-1,y+1)$
$f(x,y-1)$	$f(x,y)$	$f(x,y+1)$
$f(x+1,y-1)$	$f(x+1,y)$	$f(x+1,y+1)$

Pixels of image
section under mask

Fig. 3.23: Mechanics of linear spatial filtering, using a 3×3 mask.

Example 3.6 Linear Filtering with 3x3 Mask

Using the names in fig. 3.23, the result R of linear filtering with the filter mask at a point (x, y) in the image is

$$\begin{aligned} R = & w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + w(-1, +1)f(x - 1, y + 1) \\ & + w(0, -1)f(x, y - 1) + w(0, 0)f(x, y) + w(0, +1)f(x, y + 1) \\ & + w(1, -1)f(x + 1, y - 1) + w(1, 0)f(x + 1, y) + w(1, +1)f(x + 1, y + 1) \end{aligned}$$

*

Hint: Mask usually have odd sizes:

For a mask of size $m \times n$ is assumed that $m = 2a+1$ and $n = 2b+1$, where $a, b \in N$.

In general, linear filtering is described by the expression:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (3.20)$$

Different notation of the result of the filtering process:

$$\begin{aligned} R &= w_1 z_1 + w_2 z_2 + \dots + w_{mn} z_{mn} \\ &= \sum_{i=1}^{mn} w_i z_i \end{aligned}$$

where

w_i : mask coefficients

z_i : values of the image gray levels

The process of linear filtering given in eq. (3.20) is similar to a frequency domain concept called **convolution** (see sec. 4).

For this reason, linear spatial filtering is often referred to as “convolving a mask with an image”.

The filter masks are called similarly **convolution masks** or **convolution kernels**.

Example 3.7 Alternative notation: Linear filtering with 3x3 mask

Fig. 3.24 shows another representation of a general 3x3 spatial filter mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

(3.21)

Fig. 3.24: 3x3 filter mask.

The result R can be written as

$$\begin{aligned} R &= w_1z_1 + w_2z_2 + \dots + w_9z_9 \\ &= \sum_{i=1}^9 w_i z_i \end{aligned} \tag{3.22}$$

Hint: Nonlinear spatial filters also operate on neighborhoods, but the filtering operation is based conditionally on the values of the pixels in the neighborhood under consideration and not explicitly use coefficients in the sum-of-manner.

Important: When implementing neighborhood operations it is necessary to consider what happens when the center of the filter approaches the border of the image.

Different solutions are possible:

- Simplest method: limitation of operation so that mask center will always be at a distance no less than $\frac{n-1}{2}$ pixels from the border.
- Processing of pixels near the border with a partial filter mask.
- Adding rows and columns of 0's (or another constant gray level).
- Padding by replicating rows or columns.

3.5 Smoothing Spatial Filters

Application of **smoothing filters**:

- **Blurring:**
 - Removal of small details from an image
 - Bridging of small gaps in lines or curves
- **Noise reduction**

Some background information of noise in the DIP context can be found in appendix A1.

3.5.1 Linear Smoothing Filters

Idea: Calculation of the average of the pixels contained in the neighborhood of the filter mask. The process reduces the "sharp" transitions in gray levels.

Advantage: Because random noise typically consists of such sharp transitions, smoothing reduces obviously noise and irrelevant details of an image.

Disadvantage: Edges, which are desirable features of an image and characterized by sharp transitions in gray levels are blurred as well.

The standard average of an image can be obtained with a filter mask according to fig. 3.25(a).

Using eq. (3.21) follows:

$$R = \frac{1}{9} \sum_{i=1}^9 z_i \quad (3.23)$$

Hint: It is computationally more efficient to have coefficients valued 1 than 1/9.

A $m \times n$ mask always have a normalizing constant multiplier equal to $\frac{1}{m \cdot n}$ (here: 1/9).

The filter mask in fig. 3.25(b) yields a so-called weighted average, giving more importance (weight) to some pixels at the expense of others.

Note: The sum of all coefficients is 16 (a power of 2), which is attractive for computer implementation.

$$\frac{1}{9} \times \begin{array}{|c|c|} \hline 1 & 1 \\ \hline 1 & 1 \\ \hline 1 & 1 \\ \hline \end{array}$$

(a)

$\frac{1}{16} \times$	1	2	1
	2	4	2
	1	2	1

(b)

Fig. 3.25: Two 3x3 smoothing filter masks

The **Gaussian smoothing filter** is another operator that uses a kernel that represents the shape of a Gaussian hump.

The continuous version of the Gaussian distribution in 1D has the form:

$$w(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (3.24)$$

where σ : standard deviation of the distribution.

Assumption: The distribution has the mean of zero (i. e. it is centered about the line $x=0$).

The distribution is illustrated in fig. 3.26.

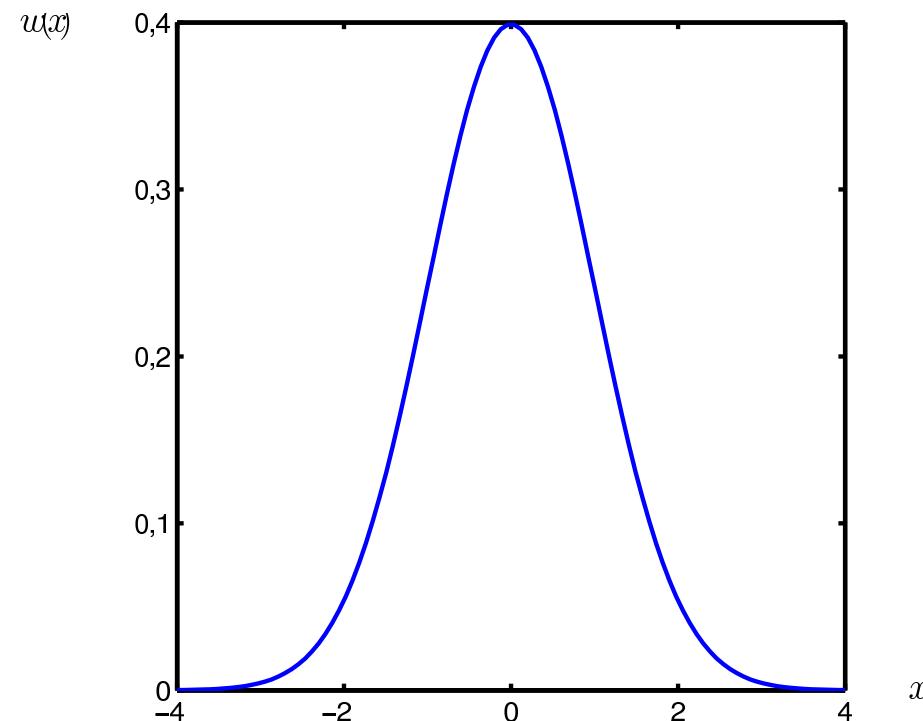


Fig. 3.26: Gaussian distribution with mean 0 and $\sigma = 1$.

In 2D, an isotropic (i. e. circularly symmetric) Gaussian has the form:

$$w(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2}\frac{x^2+y^2}{2\sigma^2}} \quad (3.25)$$

This distribution is shown in fig. 3.27.

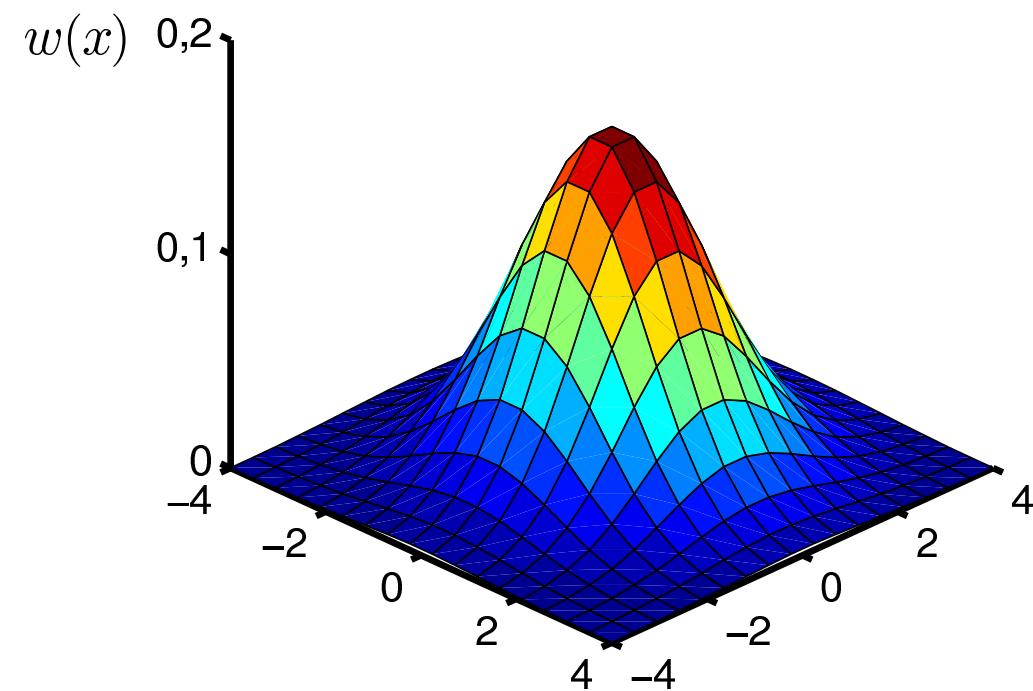


Fig. 3.27: Gaussian distribution with mean $(0, 0)$ and $\sigma = 1$.

Idea: Use of 2D distribution as a **point-spread function**.

Now: Discrete approximation to the Gaussian function.

In theory: Gaussian distribution is non-zero everywhere
→ infinitely large convolution mask required.

In practice: Gaussian distribution is effectively zero more than about three standard deviations from the mean
→ truncation of mask at this point.

Fig. 3.28 shows a suitable integer valued convolution mask.

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

Fig. 3.28: Approximation to Gaussian function with $\sigma = 1.4$

Hint: The convolution can be performed fairly quick since the eq. (3.25) is separable into x and y components.

Thus the convolution can be performed by first convolving with a 1D Gaussian in the x direction, and then convolving with another 1D Gaussian in the y direction.

Fig. 3.29 shows the 1D component mask that can be used to produce the full mask shown in fig. 3.28.

1.3	3.2	3.8	3.2	1.3
-----	-----	-----	-----	-----

Fig. 3.29: One of the pair of 1D convolution masks used to calculate a full 2D mask.

Important: The Gaussian is the only completely circularly symmetric operator which can be decomposed in such a way.

Hint: Gaussian smoothing has been attributed with some amount of biological plausibility, e. g. some cells in the visual pathways of the brain have an approximately Gaussian response.

With reference to eq. (3.20), the general implementation for filtering an $M \times N$ image with a weighted averaging filter of size $m \times n$ is given by the expression

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)} \quad (3.26)$$

Hint: The constant in the denominator is typically applied to all pixels of the output image *after* the filtering process is completed.

Fig. 3.30 to 3.32 show the effects of smoothing as a function of the filter size.



Fig. 3.30: Smoothing filters using Gaussian filter masks of different sizes. (a) Original image. (b)-(f) Results of mask sizes $n = 3, 5, 9, 15$ and 35 , respectively.

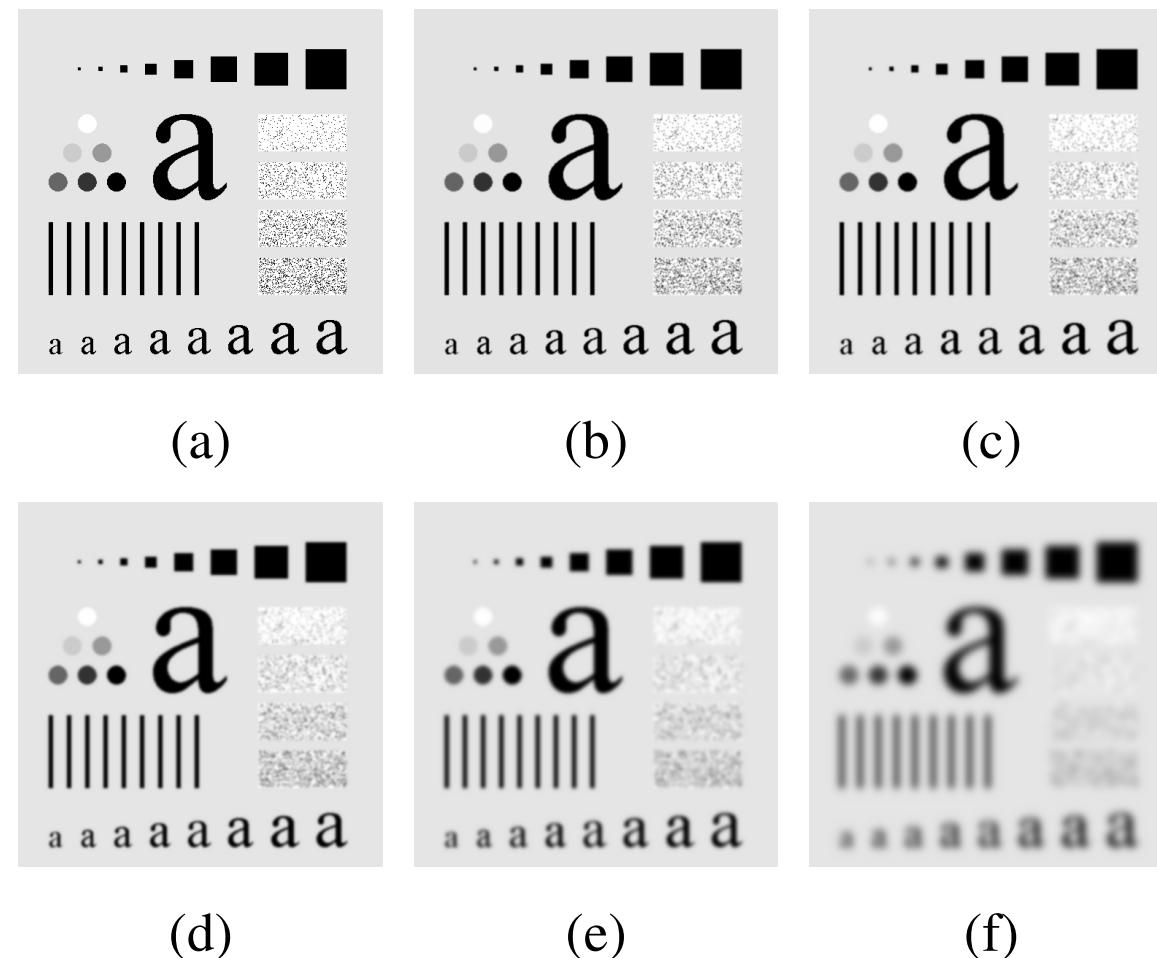


Fig. 3.31: (a) Original image. (b)-(f) Results of smoothing with Gaussian filter masks as used in fig. 3.30.
(From [GW18])

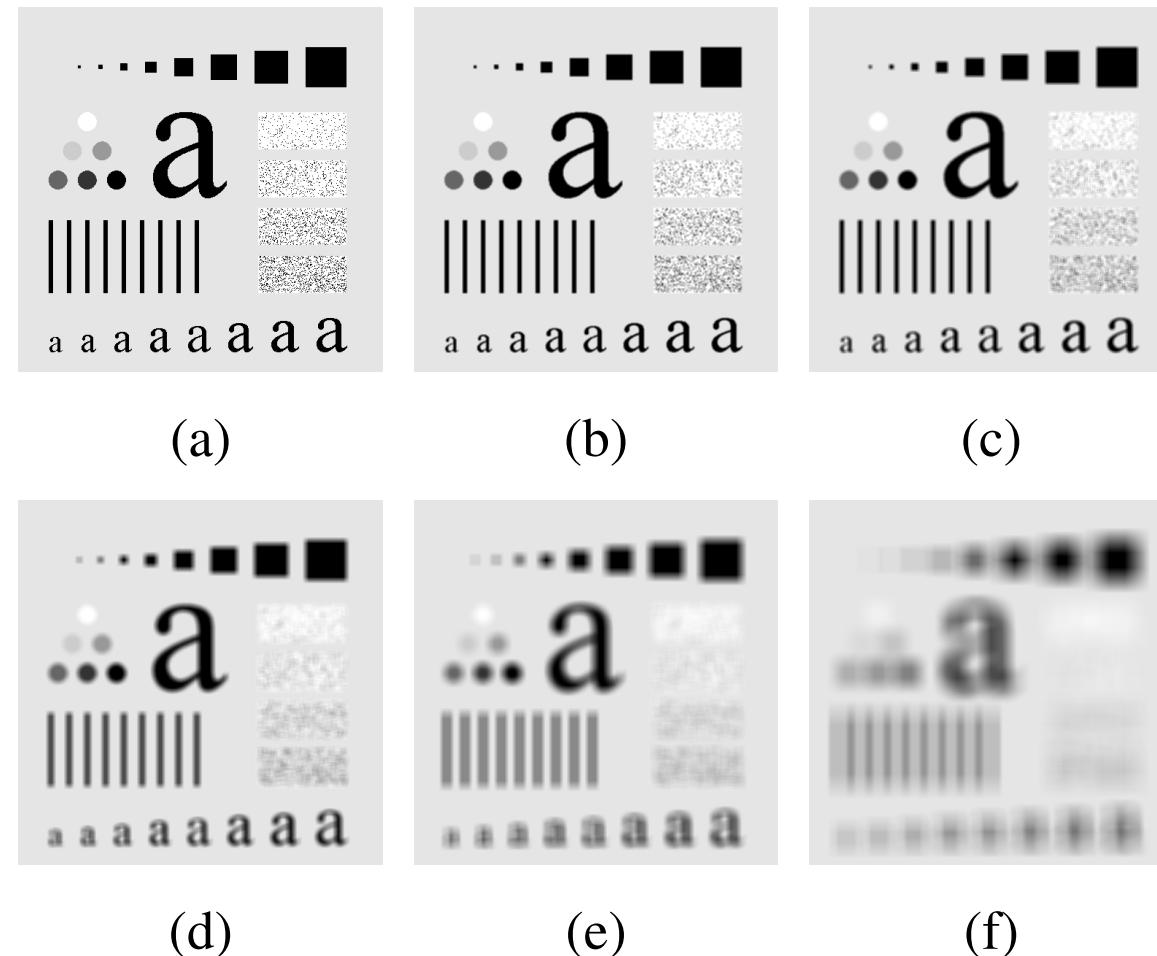


Fig. 3.32: Smoothing filters using square averaging filter masks of different sizes (compare fig. 3.25(a)).
(a) Original image. (b)-(f) Results of mask sizes $n = 3, 5, 9, 15$ and 35 , respectively. (From [GW18])

The use of smoothing filters for blurring an image for the purpose getting a gross representation of objects of interest can be explained with fig. 3.33

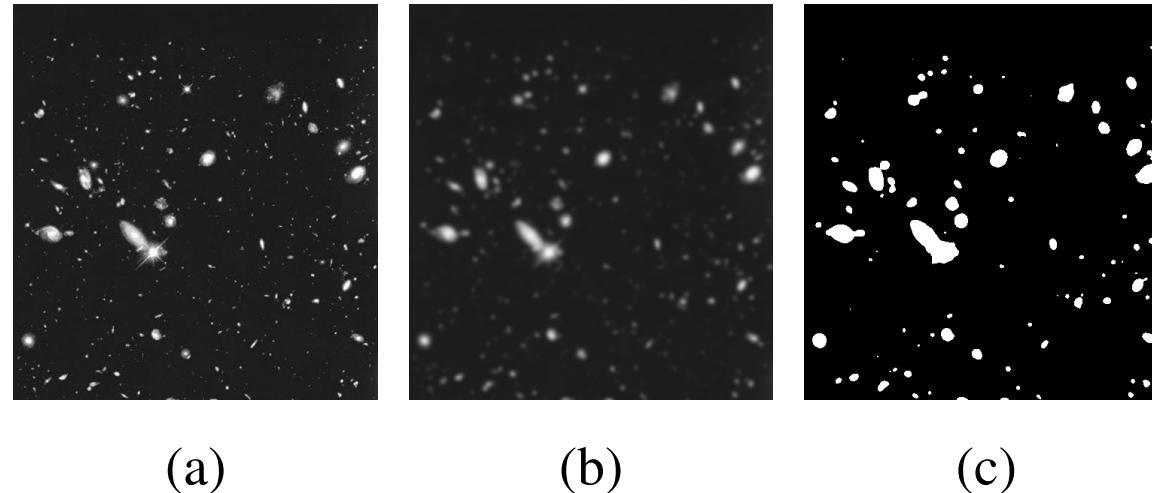


Fig. 3.33: (a) Image from the Hubble Space Telescope. (b) Image processed by a 15×15 averaging mask. (c) Result of thresholding (b). (From [GW18])

3.5.2 Ranking Filters

Ranking filters (as well referred to as **order-statistics filters**) are non-linear spatial filters whose response is based on ranking the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.

Best-known example: **median filter**

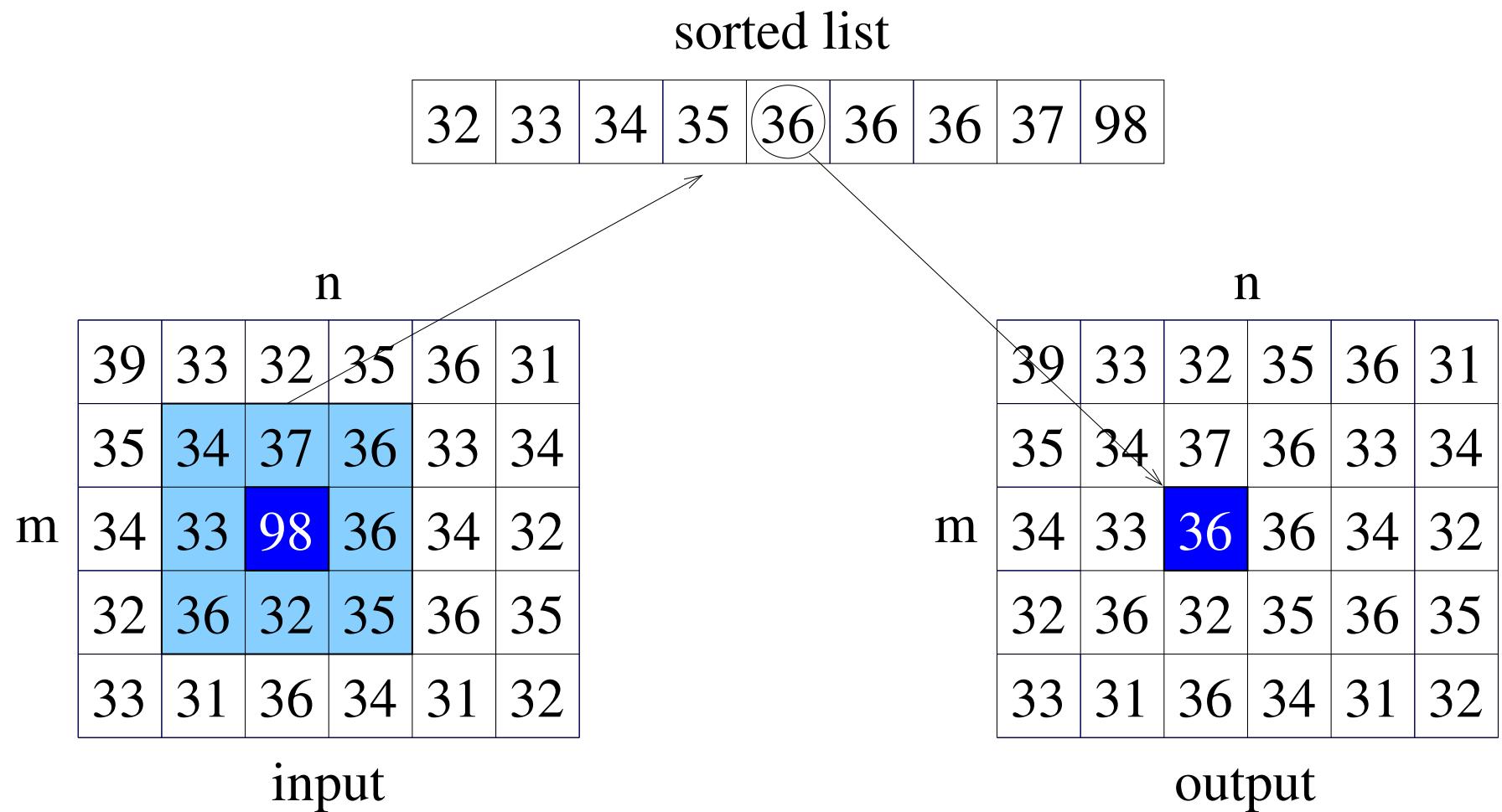
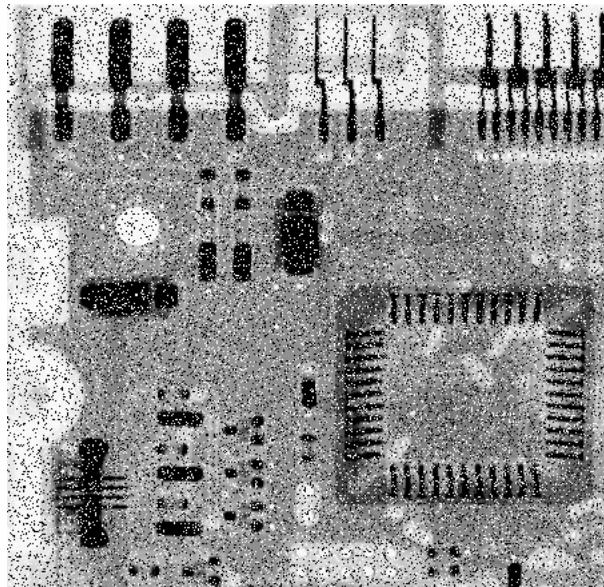
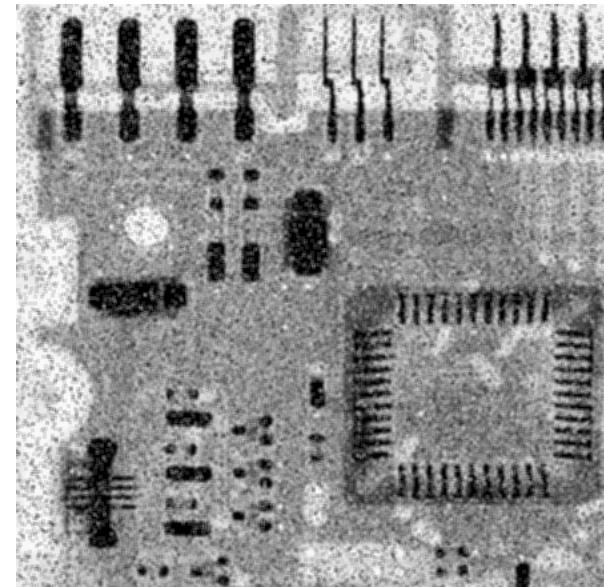


Fig. 3.34: Illustration of the node operation of a 3×3 median filter. (Adapted from [Jäh12])

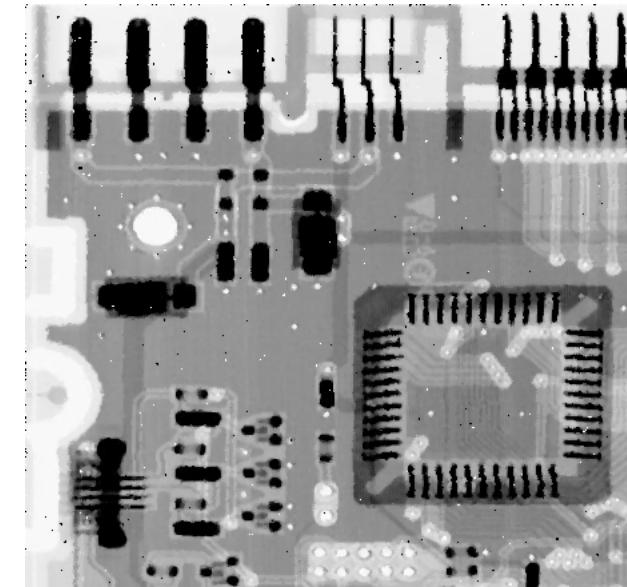
- Advantages:**
- Median filters are particularly effective in the presence of **impulse noise** (salt-and-pepper noise, see appendix A.1).
 - They provide excellent noise-reduction capabilities with considerably less blurring than linear smoothing filters of the same size.



(a)



(b)

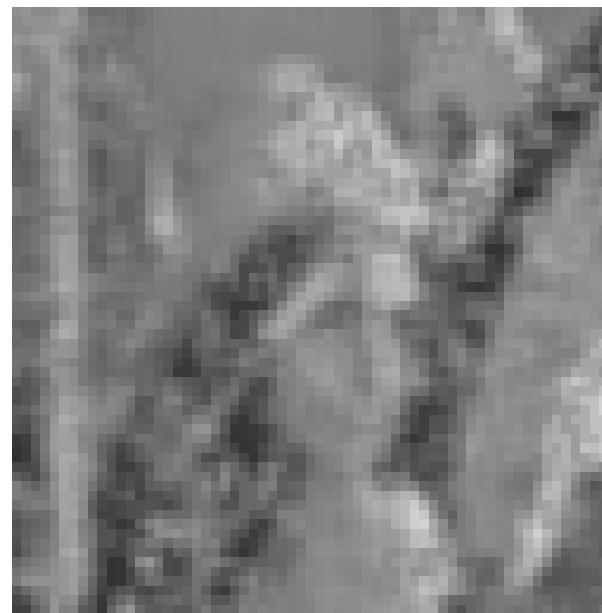


(c)

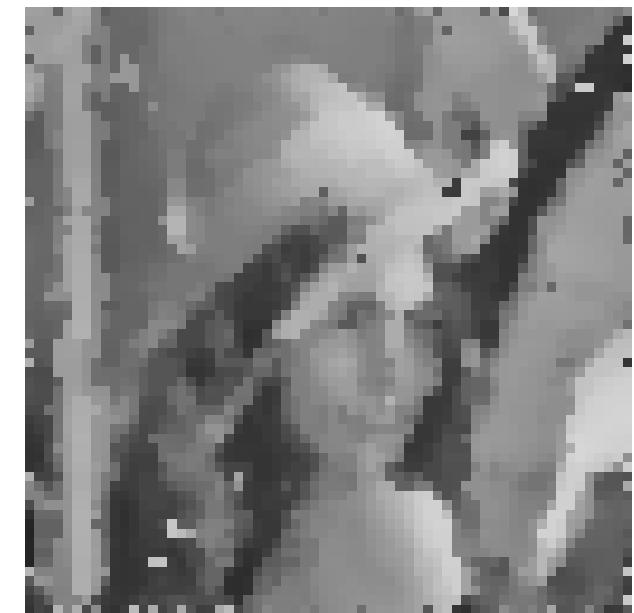
Fig. 3.35: (a) X-ray image of a circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (From [GW18])



(a)



(b)



(c)

Fig. 3.36: (a) Lena, superimposed with impulse noise. (b) and (c): For methods see fig. 3.35 (b) and (c).

3.6 Sharpening spatial filters

Application of **sharpening filters**:

- Highlighting of fine details in an image.
- Enhancement of details that have been blurred, either in error or as a natural effect of a particular method of image acquisition.

Idea: Since averaging is analogous to integration, it is logical to conclude that sharpening could be accomplished by spatial differentiation.

In this chapter:

Various ways of defining and implementing operators for sharpening by digital differentiation.

Important: The strength of the response of a derivative operator is proportional to the degree of discontinuity of the point at which the operator is applied.

→ the operators will enhance edges and other discontinuities (e. g. noise) and deemphasize areas with no or slowly varying gray-level values.

3.6.1 Differentiation of Images

Now: Sharpening filters based on

- First-order derivatives and
- Second-order derivatives

of images.

Assumption: 1D derivatives

Analysis of typical image details:

- Constant gray level (flat segments)
- At the onset and end of discontinuities (step and ramp discontinuities), and
- Along gray-level ramps.

These types of discontinuities can be used to model

- Noise points,
- Lines, and
- Edges

in an image.

Image details	First-order derivatives	Second-order derivatives
Flat segment	0	0
Ramp of constant slope	nonzero	0
Onset and end of discontinuity	nonzero	nonzero

Table 3.2: Image details and their derivatives.

- Remember:**
- Images are digital quantities.
 - Their values are finite.
 - The maximum possible gray-level change is also finite.
 - The shortest distance over which a gray-level change can occur is between adjacent pixels.

Basic definition of the first-order derivative of a one-dimensional function $f(x)$:

$$\frac{\partial f}{\partial x} = f(x + 1) - f(x) \quad (3.27)$$

Hint: Partial derivative used to keep the notation the same way when an image function of two variables is considered.

Definition of a second-order derivative:

$$\frac{\partial^2 f}{\partial x^2} = f(x + 1) + f(x - 1) - 2f(x) \quad (3.28)$$

Fig. 3.37 visualizes the differentiation of typical image structures.

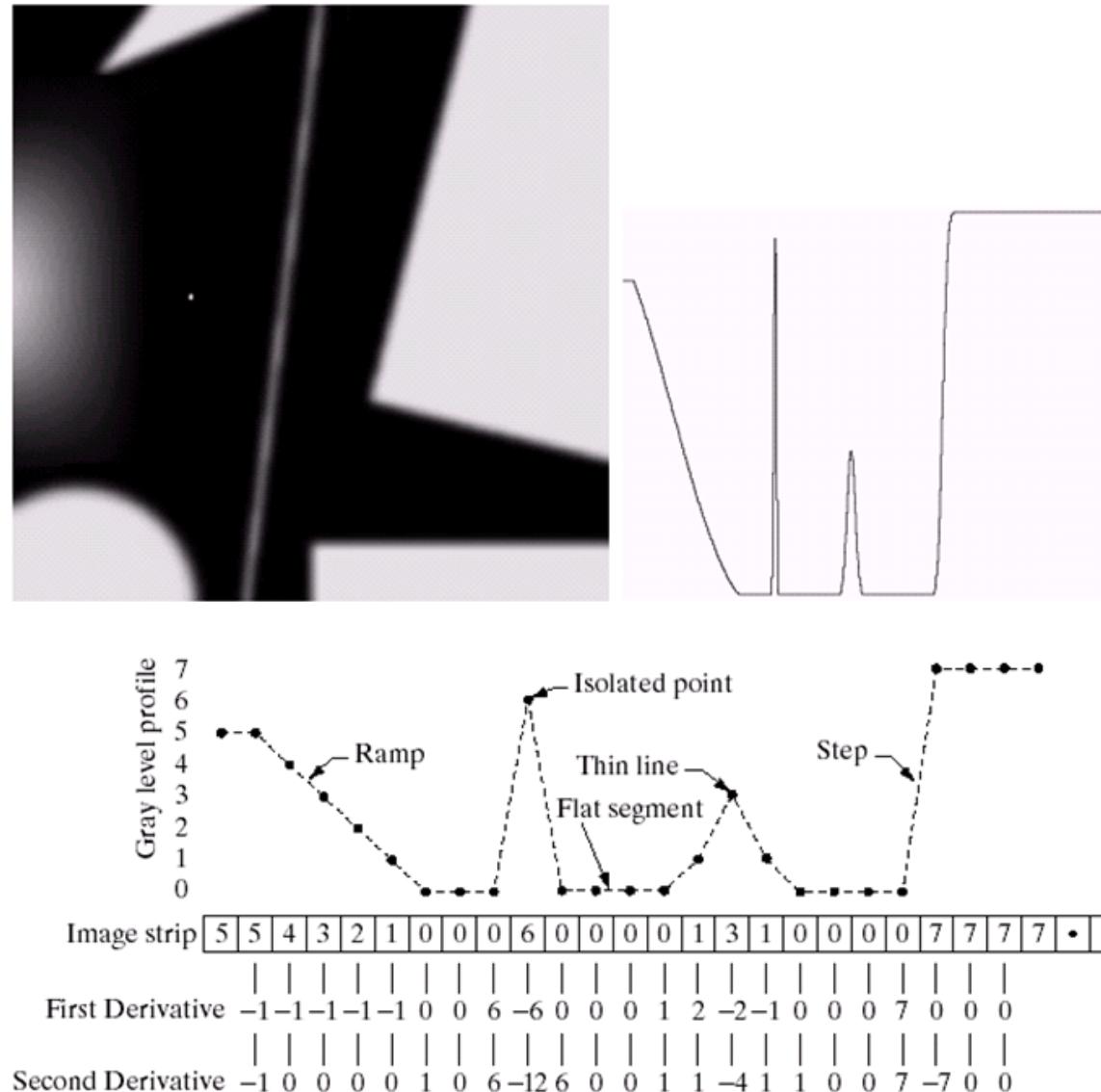


Fig. 3.37: (a) A simple image. (b) 1D horizontal gray-level profile along the center of the image and including an isolated noise point. (c) Simplified profile (the points are joint by darked lines to simplify interpretation.). (From [GW18])

- Interpretation:**
- First-order derivatives generally produce thicker edges in an image.
 - Second-order derivatives
 - have a stronger response to fine detail including noise and
 - produce a double response to a gray-level step.

The second-order derivative is better suited for image enhancement than the first-order derivative and will be investigated in section 3.6.2.

First-order derivatives are mainly used for edge detection, but they can also have important use in image enhancement. They are discussed in section 3.6.3.

3.6.2 Second-Order Derivatives - The Laplacian

Idea: Discrete formulation of the second-order derivative as an **isotropic filter**:

Filter response should be independent of the direction of the discontinuities in the image (**rotation invariance**).

Simplest isotropic derivative operator:

→ **Laplacian operator**

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3.29)$$

Hint: As derivatives of any order are linear operations, the Laplacian is a linear operator.

Eq. (3.29) in discrete form: Partial second-order derivative in the x- and y- direction:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \quad (3.30)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \quad (3.31)$$

Summing the components yields the digital implementation of the 2D Laplacian in eq. (3.29):

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)] \quad (3.32)$$

Eq. (3.32) can be implemented using one of the commonly used masks in fig. 3.38.

0	1	0
1	-4	1
0	1	0

(a)

1	1	1
1	-8	1
1	1	1

(b)

-1	2	-1
2	-4	2
-1	2	-1

(c)

0	-1	0
-1	4	-1
0	-1	0

(d)

-1	-1	-1
-1	8	-1
-1	-1	-1

(e)

1	-2	1
-2	4	-2
1	-2	1

(f)

Fig. 3.38: Commonly used discrete approximations to the Laplacian operator. (a)-(c) Operators using a negative peak (more common). (d)-(f) Operators using a positive peak.

Interpretation: The masks shown in fig. 3.38(a) and (d) give an isotropic result for rotations in increments of 90° .

The masks in fig. 3.38(b) and (e) yield isotropic results for increments of 45° .

Visualization of enhancement results:

The Laplacian is a derivative operator:

- It highlights gray-level discontinuities in an image and
- deemphasizes regions with slowly varying gray-levels.

This tends to produce images with

- grayish edge lines and other continuities,
- all superimposed on a dark, featureless background.

Counter measure:

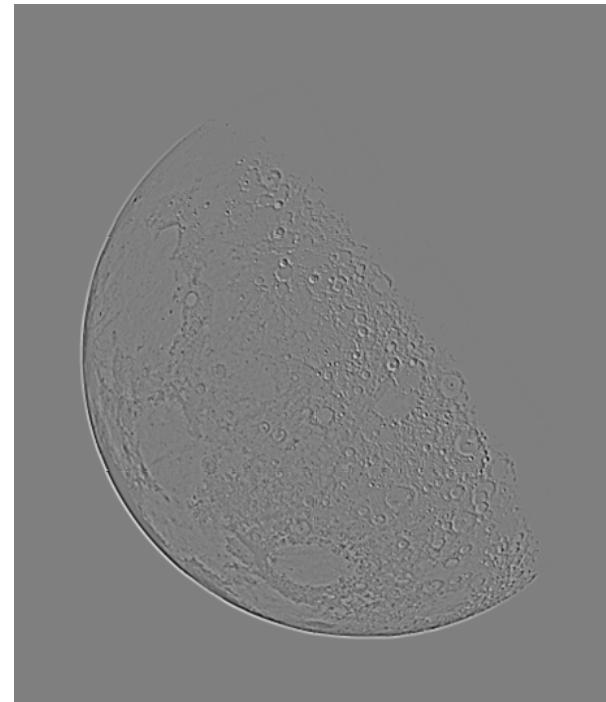
Background features can be “recovered” while still preserving the sharpening effect of the Laplacian operation by simply adding (or subtracting) the original and the Laplacian images.

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is negative.} \\ f(x, y) + \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is positive.} \end{cases} \quad (3.33)$$

Fig. 3.40 illustrates the results of the correction measure.



(a)



(b)



(c)

Fig. 3.39: (a) Image of the North Pole of the moon. (b) Laplacian filtered image. (c) Image enhanced by using eq. (3.33) and subsequent clipping. (Adapted from [GW18])



(a)



(b)



(c)

Fig. 3.40: (a) Lena. (b) Laplacian filtered image. (c) Image enhanced using eq. (3.33).

Interpretation: The resulting image in fig. 3.40(c) is unmistakably clearer and sharper than the original image.

→ Laplacian-based enhancement is a fundamental tool used frequently for sharpening digital images.

Simplifications:

The computation of the image in fig. 3.40(c) can be considerably simplified by substituting eq. (3.32) for $\nabla^2 f(x, y)$ in the first line of eq. (3.33):

$$\begin{aligned} g(x, y) &= f(x, y) - [f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1)] + 4f(x, y) \\ &= 5f(x, y) - [f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1)] \end{aligned} \quad (3.34)$$

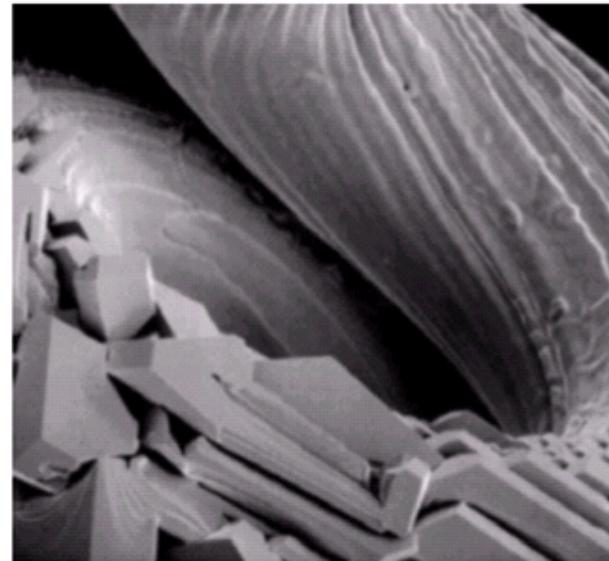
Hint: The results obtainable with the mask with the diagonal terms usually are a little bit sharper than those obtained with the more basic mask (see fig. 3.41).

0	-1	0
-1	5	-1
0	-1	0

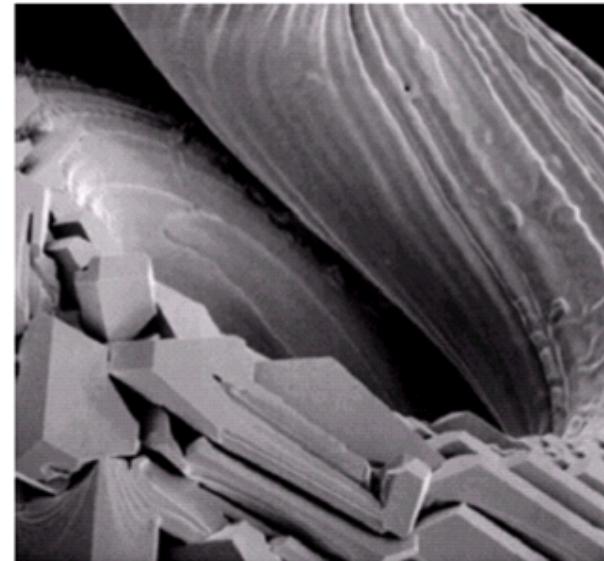
(a)

-1	-1	-1
-1	9	-1
-1	-1	-1

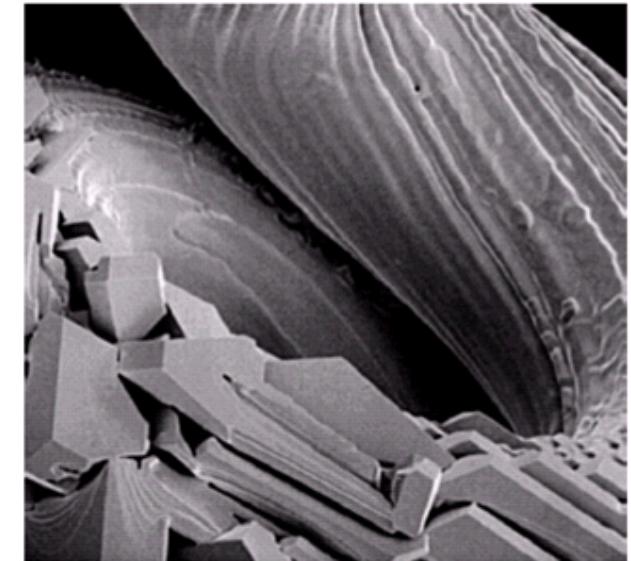
(b)



(c)



(d)



(e)

Fig. 3.41: (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Scanning with the masks in (a) and (b), respectively.

The Laplacian generally is not used in its original form for edge detection for several reasons:

- The magnitude of the Laplacian produces double edges (see figs. 3.42 and 3.43)
→ undesirable effect due to complication of segmentation
- As a second-order derivative, the Laplacian is unacceptably sensitive to noise (see fig. 3.43).
- The Laplacian is unable to detect the direction of an edge.

To counter this

- an image is often Gaussian smoothed before applying the Laplacian and
- the **zero-crossing** property of the Laplacian is used for edge detection.

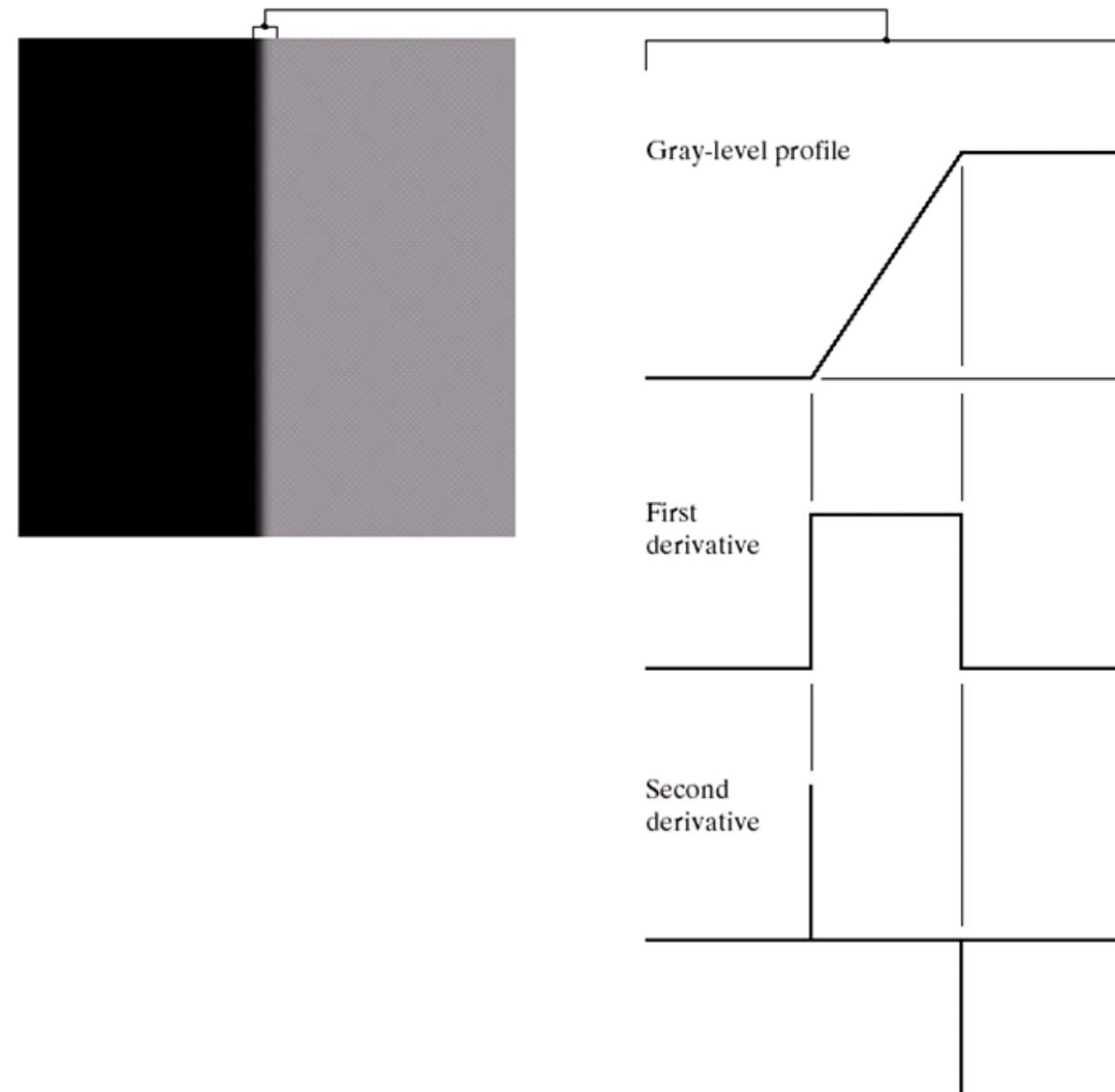


Fig. 3.42: (a) Two regions separated by a vertical edge. (b) Detail near the edge, showing a gray-level profile, and the first and second derivatives of the profile. (From [GW18])

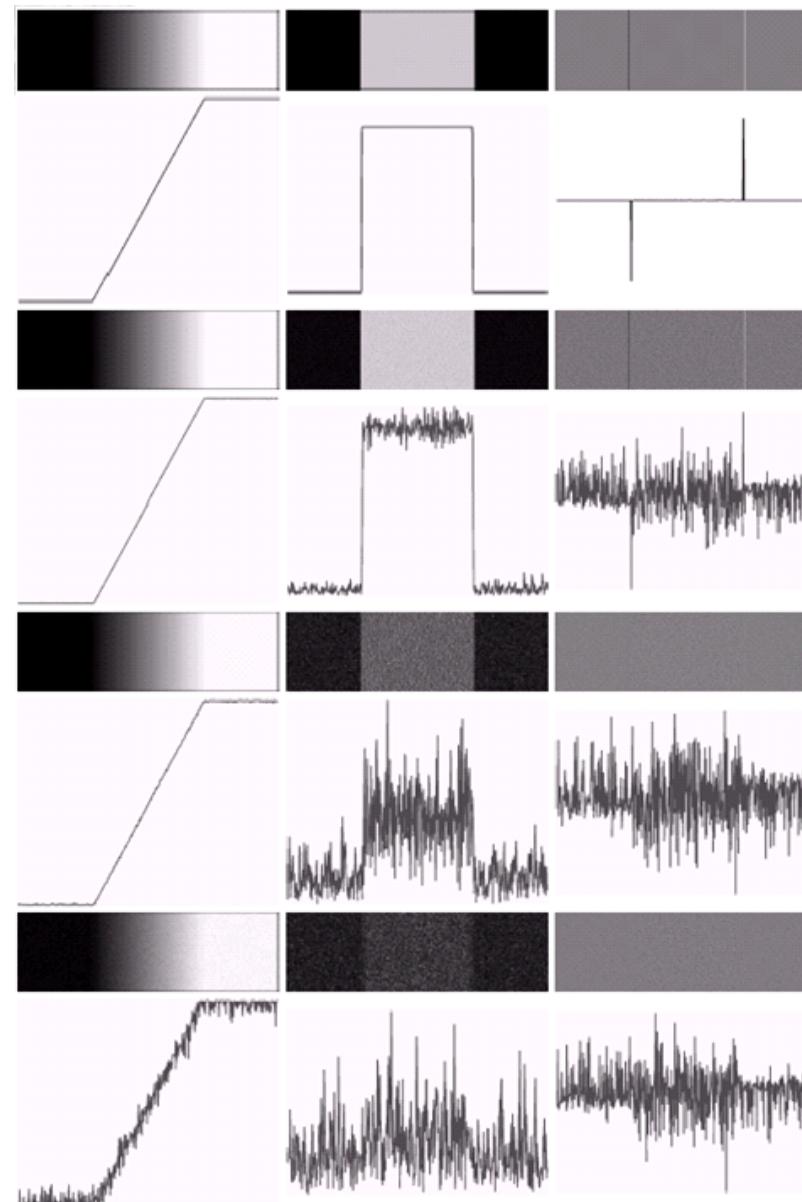


Fig. 3.43: (a) Images and gray-level profiles of a ramp edge corrupted by random Gaussian noise of mean 0 and $\sigma = 0.0, 0.1, 1.0, 10.0$, respectively. (b) First-derivative images and gray-level profiles. (c) Second-derivative images and gray-level profiles. (From [GW18])

Important: Since the convolution operation is associative, the Gaussian operator can be convolved with the Laplacian filter first.

Then this hybrid filter can be convolved with the image.

Advantages:

- Due to the usually small Gaussian and Laplacian mask sizes, this method requires far fewer arithmetic operations.
- The **LoG** mask (**Laplacian of Gaussian**) can be precalculated in advance, so only one convolution needs to be performed at run-time on the image.

Gaussian function:

$$w(r) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2}(r^2 - \mu^2 q)/2\sigma^2} \quad (3.35)$$

where

$$r^2 = x^2 + y^2$$

μ : mean

σ : standard deviation

The Laplacian of w (the second derivative of w with respect to r) is $\nabla^2 w(r)$.

Fig. 3.44 shows a 3D plot, an a image, and a cross section of the LoG function.

Added is a 5x5 mask which approximates $\nabla^2 w(r)$.

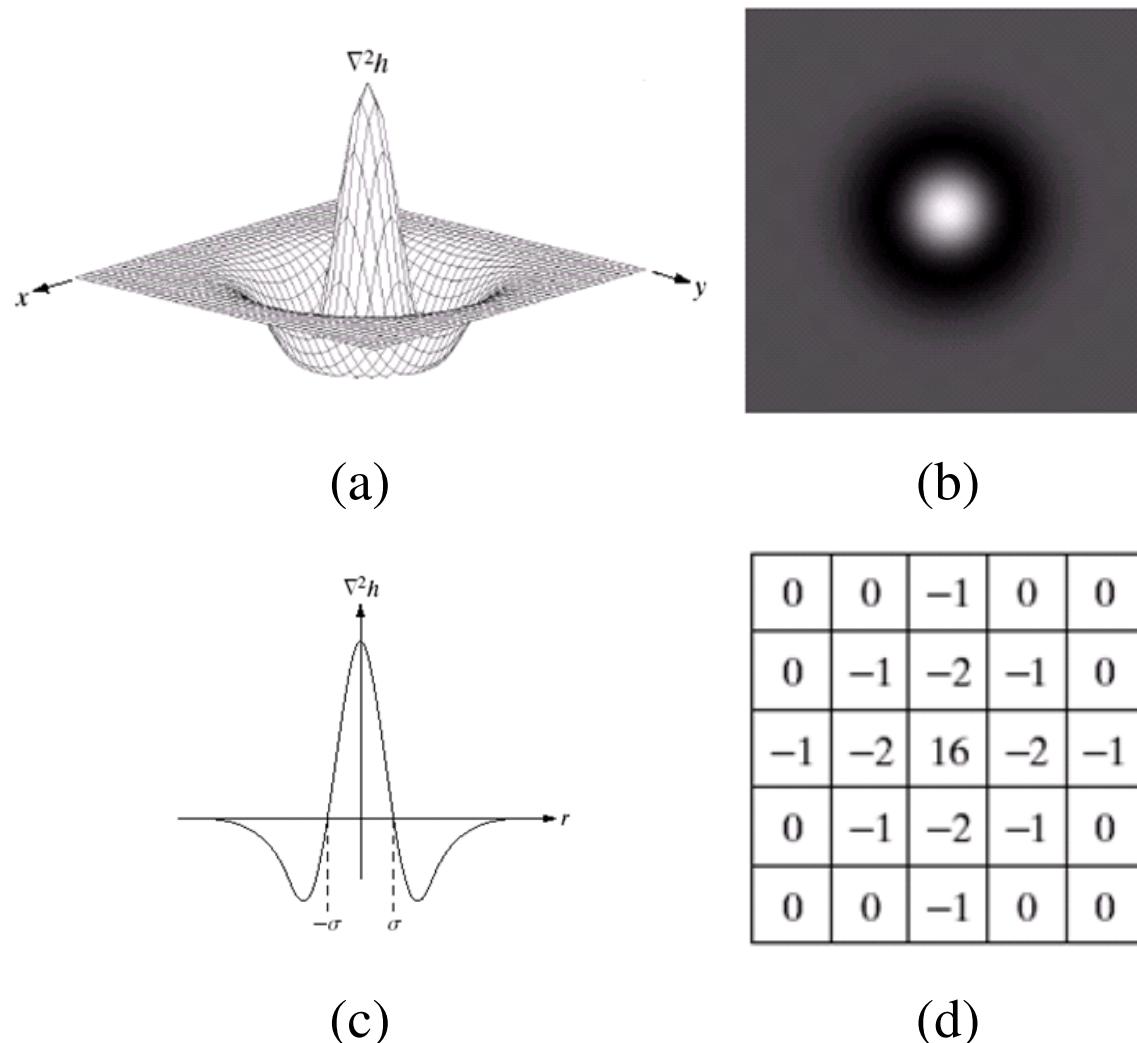


Fig. 3.44: Laplacian of Gaussian (LoG). (a) 3D plot. (b) Image (black is negative, gray is the zero plane, and white is positive). (c) Cross-section showing zero-crossings. (d) 5x5 mask approximation to the shape of (a). (From [GW18])

Fig. 3.45 shows the application of the LoG operator to an image from the medical domain, an angiogram already discussed in fig. 1.10(b).

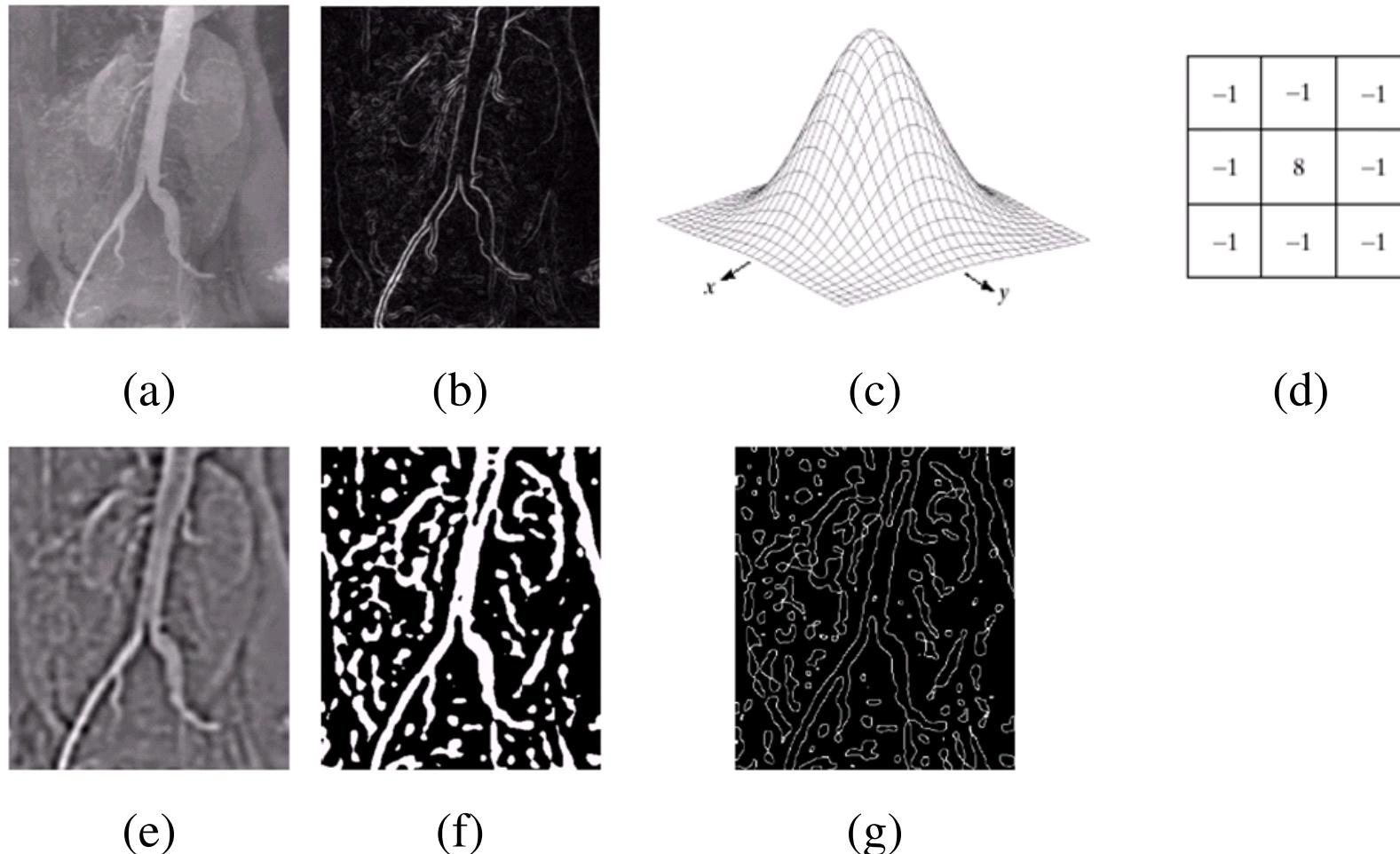


Fig. 3.45: (a) Original image. (b) Sobel gradient (shown for comparison). (c) Spatial Gaussian smoothing function. (d) Laplacian mask. (e) LoG. (f) Thresholded LoG. (g) Zero Crossings. (From [GW18])

3.6.3 First-Order Derivatives - The Gradient

First derivatives in DIP are implemented using the magnitude of the gradient.

For a function $f(x,y)$, the **gradient** of f at coordinates (x,y) is defined as the 2D **column vector**

$$\begin{aligned}\nabla f &= \begin{bmatrix} G_x \\ G_y \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}\end{aligned}\tag{3.36}$$

The magnitude of this vector is given by

$$\begin{aligned}|\nabla f| &= \text{mag}(\nabla f) \\ &= [G_x^2 + G_y^2]^{\frac{1}{2}} \\ &= \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{\frac{1}{2}}\end{aligned}\tag{3.37}$$

- Interpretation:**
- The components of the gradient vector itself are linear operators, but the magnitude obviously is not.
 - The partial derivatives in eq. (3.36) are not rotation invariant, while the magnitude of the gradient vector is.

Hint: Although it is not strictly correct, the magnitude of the gradient vector is often referred to as the **gradient**.

This term will be used in the following discussions, explicitly referring to the vector or its magnitude only in cases where confusion is likely.

The discrete formulation of eq. (3.36) over an entire image is not trivial.

Common practice:

Approximation of the magnitude of the gradient by using absolute values instead of squares and square roots:

$$|\nabla f| \approx |G_x| + |G_y| \quad (3.38)$$

Now: Digital approximations to the preceding equations.

The notation in fig. 3.46(a) will be used to denote image points in a 3×3 region.

f_1	f_2	f_3
f_4	f_5	f_6
f_7	f_8	f_9

(a)

-1	0
0	1

(b)

-1	-2	-1
0	0	0
1	2	1

(d)

0	-1
1	0

(c)

-1	0	1
-2	0	2
-1	0	1

(e)

Fig. 3.46: (a) A 3×3 region of an image. (b) and (c) Masks for Roberts gradient operator. (d) and (e) Masks for Sobel operator.

Roberts gradient operator

As indicated in eq. (3.27) the simplest approximations to a first-order derivative that satisfy the conditions stated are

$$G_x = (f_9 - f_5) \quad (3.39)$$

$$G_y = (f_8 - f_6) \quad (3.40)$$

The gradient yields to

$$|\nabla f| = [(f_9 - f_5)^2 + (f_8 - f_6)^2]^{\frac{1}{2}} \quad (3.41)$$

Approximation using eq. (3.38):

$$|\nabla f| \approx |f_9 - f_5| + |f_8 - f_6| \quad (3.42)$$

Fig. 3.47 illustrates the Roberts gradient operator.

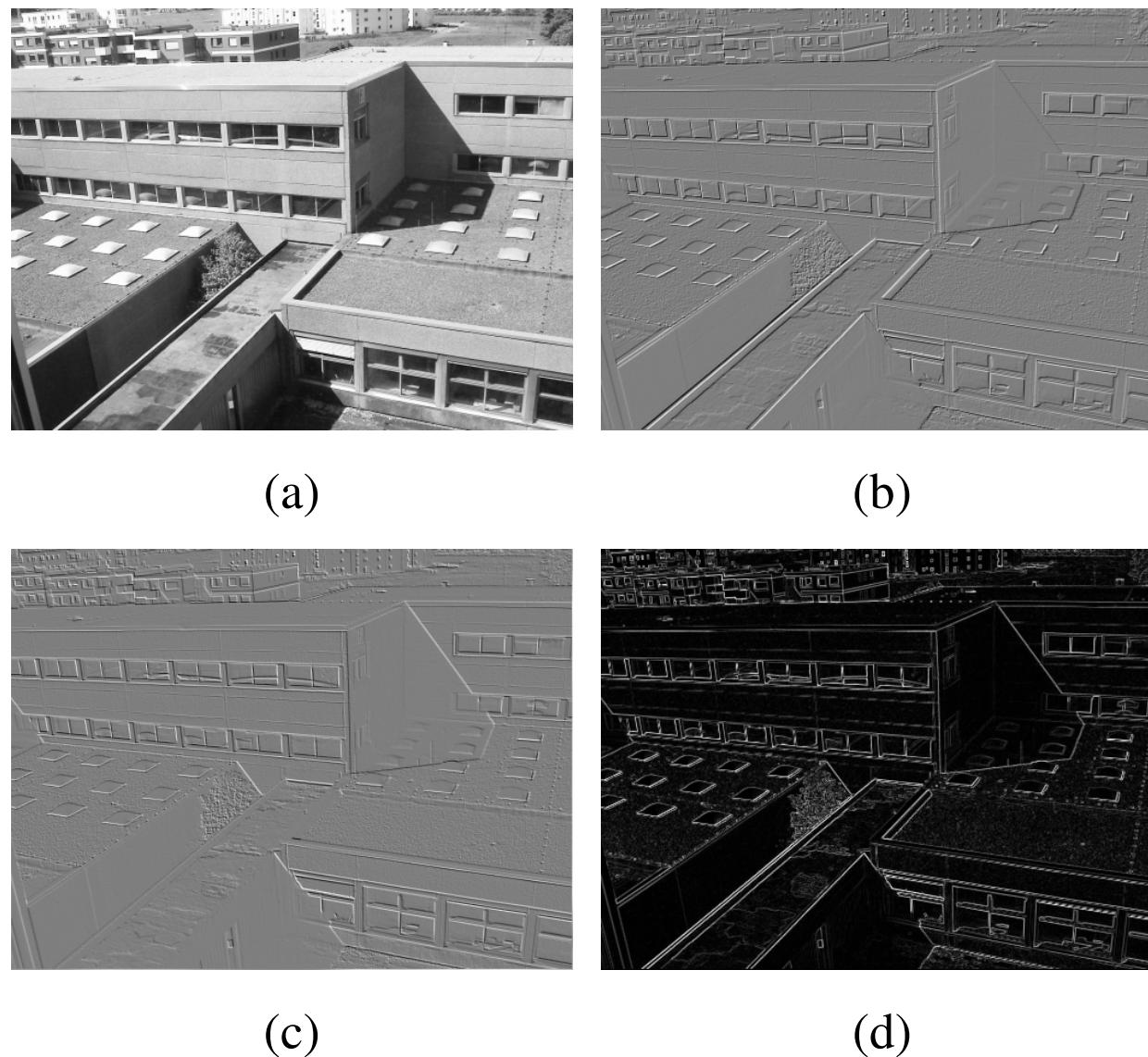


Fig. 3.47: (a) Original image. (b) Roberts of (a) using the mask in fig. 3.46(b). (c) Roberts of (a) using the mask in fig. 3.46(c). (d) Sum of (b) and (c).

Hint: Masks of even size are awkward to implement.
→ use of 3×3 filter masks.

Sobel operator

An approximation using absolute values, still at point f_5 , but using a 3×3 mask.

$$|\nabla f| \approx |(f_7 + 2f_8 + f_9) - (f_1 + 2f_2 + f_3) + (f_3 + 2f_6 + f_9) - (f_1 + 2f_4 + f_7)| \quad (3.43)$$

The masks shown in fig. 3.46(d) and (e) can be used to implement eq. (3.43) via the mechanics given in eq. (3.20).

Hint: The idea behind using a weight value of 2 is to achieve some smoothing by giving more importance to the center point.

Note: All coefficients in the mask in fig. 3.46 sum to 0; indicating that they would give a response of 0 in an area of constant gray level, as expected of a derivative operator.

Fig. 3.48 illustrates the Sobel operators.

Note: In fig. 3.47(b)-(d) and fig. 3.48(b)-(d) the gray-level 127 is added to the calculated differences.

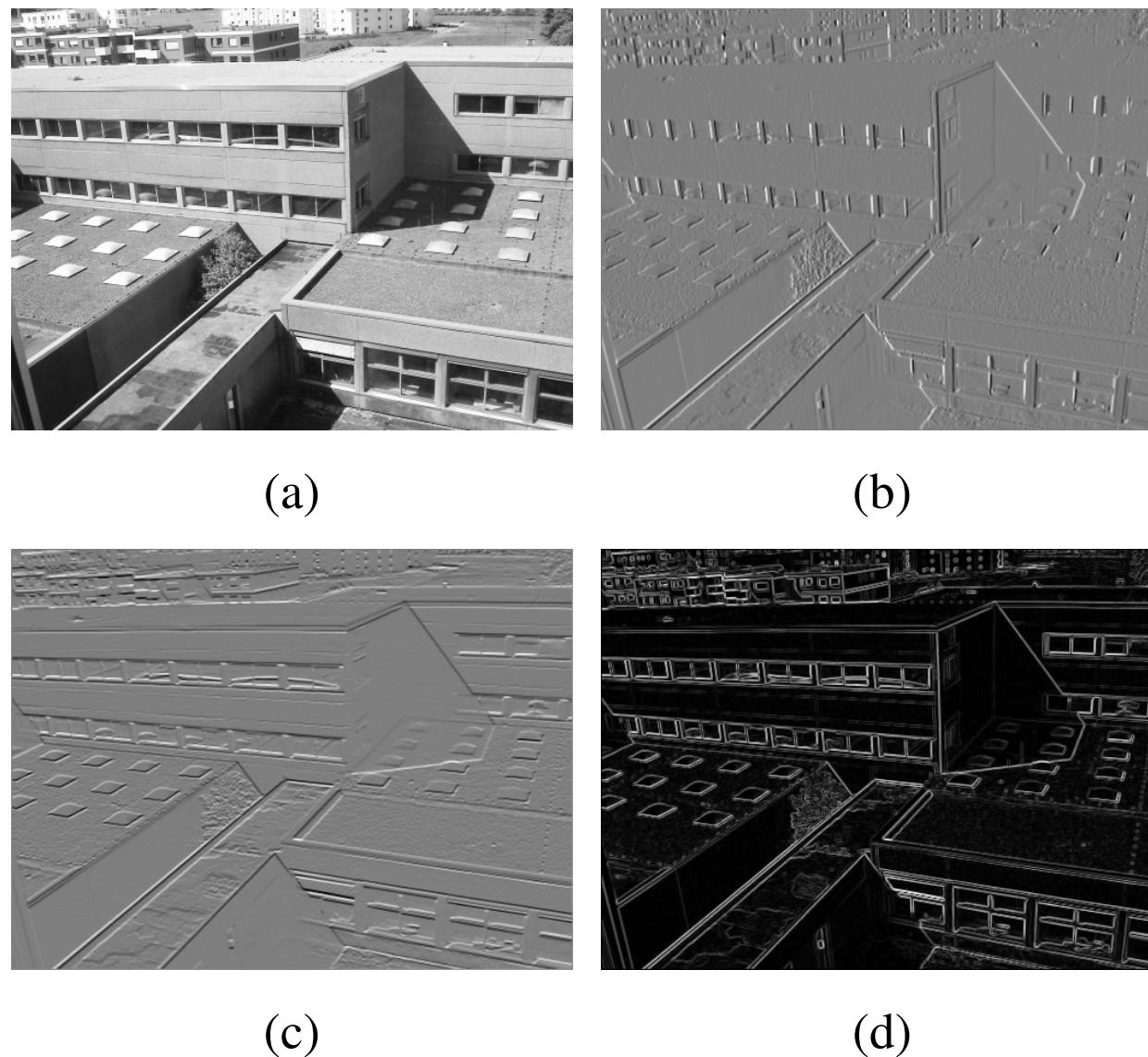


Fig. 3.48: (a) Original image. (b) Sobel of (a) using mask in fig. 3.46(d). (c) Sobel of (a) using mask in fig. 3.46(e). (d) Result using eq. (3.42).



(a)



(b)



(c)



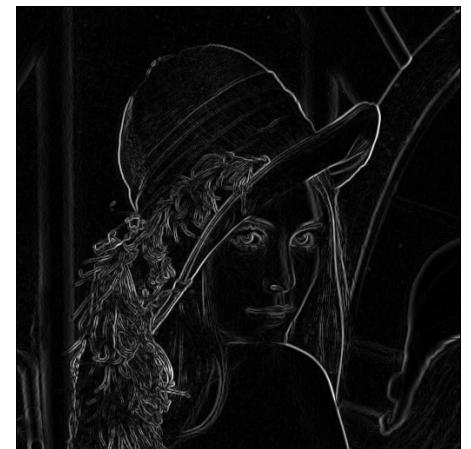
(d)



(e)



(f)



(g)

Fig. 3.49: Comparison of the first-order derivatives. (a) Original image. (b)-(d) Roberts gradient operator. (e)-(g) Sobel operators.

4 Image Enhancement in the Frequency Domain

Contents

4.1 Introduction to the Fourier Transform and the Frequency Domain	4-2
4.1.1 1D Fourier Transform and its Inverse	4-3
4.1.2 2D Fourier Transform and Its Inverse	4-18
4.1.3 Some Useful Theorems of the (2D) Fourier Transform	4-23
4.1.4 Correspondence between Filtering in the Spatial and Frequency Domains .	4-31
4.1.5 Filtering in the Frequency Domain	4-40
4.2 Smoothing Frequency Domain Filter	4-50
4.2.1 Ideal Lowpass Filter	4-51
4.2.2 Butterworth Lowpass Filter	4-59
4.2.3 Gaussian Lowpass Filter	4-64
4.2.4 Additional examples of Lowpass Filtering	4-68
4.3 Sharpening Frequency-Domain Filters	4-70
4.3.1 Ideal Highpass Filters	4-74

4.3.2	Butterworth Highpass Filters	4-76
4.3.3	Gaussian Highpass Filters	4-78
4.4	Implementation of the Fourier Transform	4-80
4.4.1	Separability	4-80
4.4.2	Implementation of the inverse Fourier transform	4-82
4.4.3	Function Padding	4-84
4.4.4	Convolution and Correlation Theorems	4-96
4.4.5	Fast Fourier Transform	4-101

Besides the spatial domain, the frequency domain can be used for image processing.

Before techniques for image enhancement can be introduced, a brief outline of the Fourier transform is given.

4.1 Introduction to the Fourier Transform and the Frequency Domain

While sec. 4.1.1 summarizes the 1D Fourier transform, which is already familiar to the audience, sec. 4.1.2 provides an extension to two dimensions.

4.1.1 1D Fourier Transform and its Inverse

At first, a single variable, continuous function, $f(x)$ is regarded.

Its **Fourier transform**, $F(u)$, is defined as

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-j 2\pi u x} dx \quad (4.1)$$

where $j = \sqrt{-1}$

Conversely, given $F(u)$, $f(x)$ can be obtained by means of the **inverse Fourier transform**:

$$f(x) = \int_{-\infty}^{+\infty} F(u) e^{j 2\pi u x} du \quad (4.2)$$

The eq. (4.1) and (4.2) comprise the **Fourier transform pair**.

Now: Consideration of discrete functions

The Fourier transform of a discrete function $f(x)$, with $x = 0, 1, \dots, M - 1$, can be derived from eq. (4.1):

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j 2\pi ux/M} \quad (4.3)$$

where $u = 0, 1, \dots, M - 1$

Eq. (4.3) describes the **discrete Fourier transform, (DFT)**, which is the foundation for this section ¹.

Similarly, given $F(u)$, the original discrete function, $f(x)$, can be obtained back using the **inverse DFT**:

$$f(x) = \sum_{u=0}^{M-1} F(u) e^{j 2\pi ux/M} \quad (4.4)$$

where $x = 0, 1, \dots, M - 1$

¹ The factor $\frac{1}{M}$ is needed if $F(0)$ should be the standard average of an image.

- Hint:**
- Sometimes the multiplier $\frac{1}{M}$ is placed in front of the inverse Fourier transform.
 - Other times, both equations are multiplied by $\frac{1}{\sqrt{M}}$.
- The location of the multiplier does not matter.

Computation of $F(u)$ in eq. (4.3):

→ Approximately M^2 summations and multiplications.

The reduction of the computational effort will be discussed at the end of section 4.4 in detail.

Important: Unlike the continuous case, the DFT or its inverse always exist, if finite quantities are used.

Using the Euler's formula

$$e^{j\phi} = \cos \phi + j \sin \phi \quad (4.5)$$

eq. (4.3) can be rewritten as

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) \left[\cos \left(\frac{2\pi ux}{M} \right) - j \sin \left(\frac{2\pi ux}{M} \right) \right] \quad (4.6)$$

using $\cos \phi = \cos(-\phi)$

for $u = 0, 1, \dots, M - 1$

- Interpretation:**
- Each term of $F(u)$ is composed of the sum of *all* values of $f(x)$. (see flash for details. [Loc]).
 - The values of $f(x)$ are multiplied by sines and cosines of various frequencies.

The domain (values of u) over which the values of $F(u)$ range is called the **frequency domain**, because u determines the frequency of the components of the transform.

Each of the M terms of $F(u)$ is called **frequency component** of the transform.

Useful analogy: Comparison of Fourier transform to a glass prism (see fig. 2.1)
 → “mathematical prism”

Important: The components of the Fourier transform are complex quantities.

Sometimes useful: Expression of $F(u)$ in polar coordinates:

$$F(u) = |F(u)| e^{-j \phi(u)} \quad (4.7)$$

where

$$|F(u)| = \left[\operatorname{Re}\{F(u)\}^2 + \operatorname{Im}\{F(u)\}^2 \right]^{\frac{1}{2}} \quad (4.8)$$

and

$$\phi(u) = \arctan \left(\frac{\operatorname{Im}\{F(u)\}}{\operatorname{Re}\{F(u)\}} \right) \quad (4.9)$$

where

$\operatorname{Re}\{F(u)\}$ and $\operatorname{Im}\{F(u)\}$: real and imaginary parts of $F(u)$.

$|F(u)|$ is called **magnitude** or **spectrum** of the Fourier transform, and $\phi(u)$ is called **phase angle** or **phase spectrum** of the transform.

The equations (4.3), (4.6), and (4.7) are three equivalent notations for the DFT.

Another important quantity: **power spectrum or spectral density**

$$P(u) = |F(u)|^2 \quad (4.10)$$

$$= Re\{F(u)\}^2 + Im\{F(u)\}^2 \quad (4.11)$$

Example 4.1 Discrete Fourier Transform of a Set of Pixels I

Given: Samples of image row (here: constant gray level):

$$M = 4;$$

$$f_k = 1 =$$

$$f(0) = f(1) = f(2) = f(3) = 1$$

Wanted: DFT{ f_k }

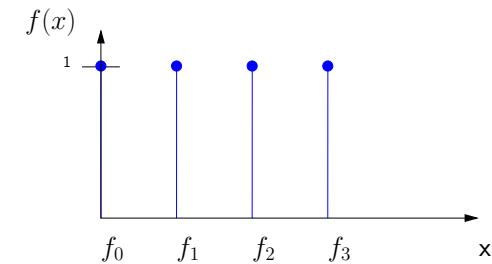


Fig. 4.1: Image samples.

Solution: With eq. (4.3):

$$F_0 = F(u = 0) = \frac{1}{4} 4 = 1$$

$$F_1 = F(u = 1) = 0$$

$$F_2 = F(u = 2) = 0$$

$$F_3 = F(u = 3) = 0$$

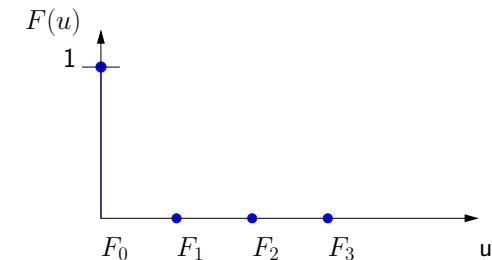


Fig. 4.2: DFT of image samples.

Interpretation: $\{f_k\}$ is an even sequence $\rightarrow \{F_k\}$ has no imaginary part.

*

Example 4.2 Discrete Fourier Transform of a Set of Pixels II

Given: Samples of image row

(here: Cosine wave with $M = 4$):

$$f(0) = 1; f(1) = 0; f(2) = -1;$$

$$f(3) = 0$$

Wanted: DFT $\{f_k\}$

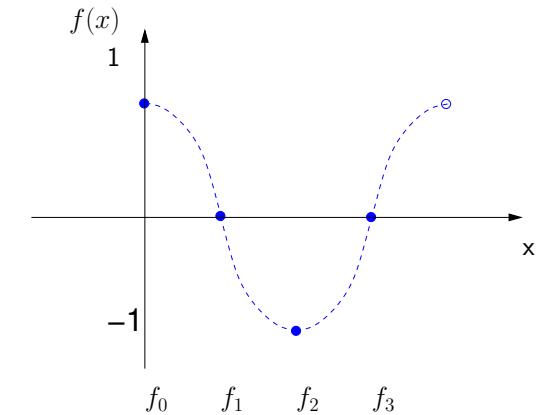


Fig. 4.3: Image samples.

Solution: With eq. (4.3):

$$F_0 = F(u = 0) = 0 \text{ (Mittelwert = 0!)}$$

$$F_1 = F(u = 1) = \frac{1}{4} (1 + (-1)(-1)) = \frac{1}{2}$$

$$F_2 = F(u = 2) = \frac{1}{4} (1 + (-1)1) = 0$$

$$F_3 = F(u = 3) = \frac{1}{4} (1 + (-1)(-1)) = \frac{1}{2}$$

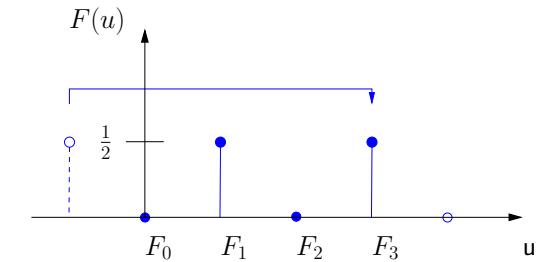


Fig. 4.4: DFT of image samples.

Interpretation: If $\{f_k\}$ contains *real* data it holds that $F_{M-l} = F_l^*$.

Example 4.3 Discrete Fourier Transform of a Set of Pixels III

Given: Samples of image row

(here: Sine wave with $M = 4$):

$$f(0) = 0; f(1) = 1; f(2) = 0;$$

$$f(3) = -1$$

Wanted: DFT $\{f_k\}$

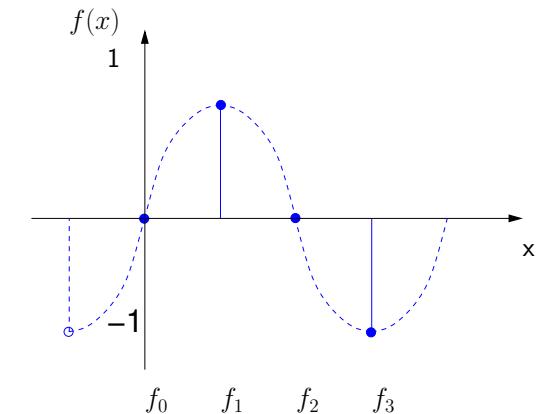


Fig. 4.5: Image samples.

Solution: With eq. (4.3):

$$F_0 = F(u = 0) = 0 \text{ (Mittelwert = 0!)}$$

$$F_1 = F(u = 1) = \frac{1}{4}(-j + (-1)j) = -\frac{j}{2}$$

$$F_2 = F(u = 2) = \frac{1}{4}(1(-1) + (-1)(-1)) = 0$$

$$F_3 = F(u = 3) = \frac{1}{4}(1j + (-1)(-j)) = \frac{j}{2}$$

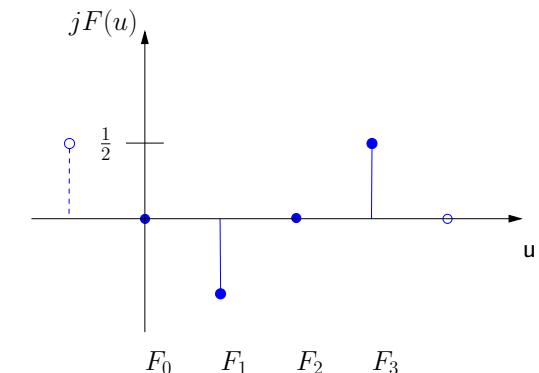


Fig. 4.6: DFT of image samples.

Interpretation: If $\{f_k\}$ contains *real* data it holds that $F_{M-l} = F_l^*$.

Example 4.4 Fourier Spectrum of two Simple 1D Functions

Consideration of a 1D example of the DFT.

Fig. 4.7(a) shows a function and Fig. 4.7(b) its Fourier spectrum.

Hint: Both, $f(x)$ and $F(u)$, are discrete functions. Points in the plots are linked to make them easier to follow visually.

Parameters:

- $M = 1024$
- $A = 1$, and
- $k = 8$ points

Hint: The spectrum is centered at $u = 0$. This can be accomplished by shifting $f(x)$ accordingly or by multiplying $f(x)$ by $(-1)^x$ before taking the transform.

In the next two figures, k is set to 16 points (see fig. 4.7(c) and (d)).

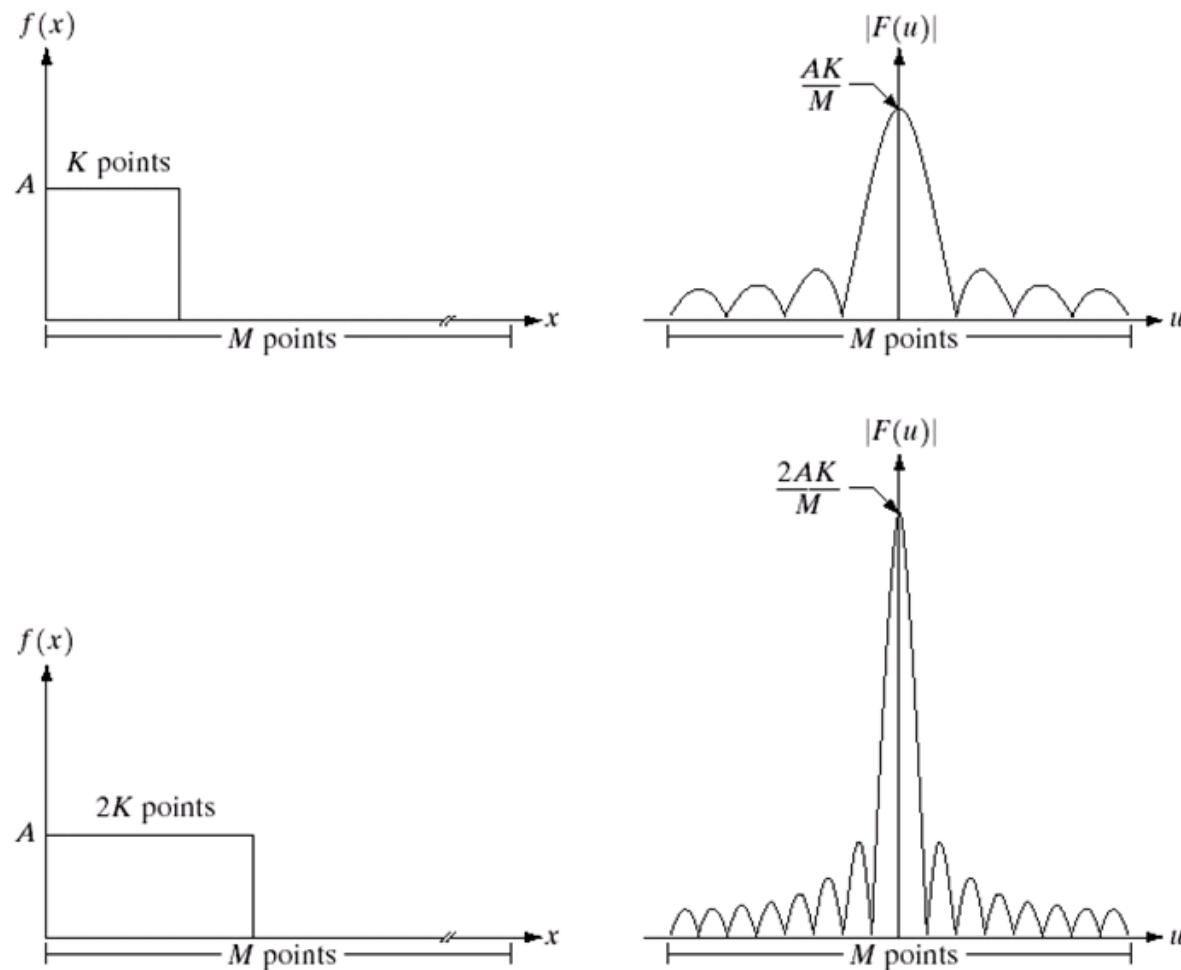


Fig. 4.7: (a) A discrete function of n points. (b) Its Fourier spectrum. (c) A discrete function with twice the number of nonzero points. (d) Its Fourier spectrum. (From [GW18])

- Interpretation:**
- The area under the curve in the spatial domain and the height of the spectrum have doubled.
 - The number of zeros in the spectrum in the same interval and the number of the samples in the spatial domain have doubled.
- “reciprocal nature” of the Fourier transform pair.
-
- *

The discrete function $f(x)$, for $x = 0, 1, \dots, M - 1$, of eq. (4.3) represents M samples from its continuous counterpart.

Important: These samples are not necessarily taken of integer values of x in the interval $[0, M - 1]$, but they are taken equally spaced.

Samples:

- $f(x_0)$ (arbitrarily located point in the sequence)
- $f(x_0 + \Delta x)$ (second point)
- $f(x_0 + k \Delta x)$ (k th sample)
- $f(x_0 + [M - 1] \Delta x)$ (final sample)

Interpretation:

$$f(x) \stackrel{!}{=} f(x_0 + x \Delta x) \quad (4.12)$$

Similarly:

$$\begin{aligned} F(u) &\stackrel{!}{=} F(u_0 + u \Delta u) \\ &\text{where } u_0: \text{true zero frequency} \\ &= F(u \Delta u) \end{aligned} \tag{4.13}$$

Δx and Δu are inversely related:

$$\Delta u = \frac{1}{M \Delta x} \tag{4.14}$$

4.1.2 2D Fourier Transform and Its Inverse

The continuous functions, $F(u)$ and $f(x)$, in the eq. (4.1) and (4.2) are easily extended to two variables:

$$F(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-j 2\pi(u x + v y)} dx dy \quad (4.15)$$

$$f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u, v) e^{+j 2\pi(u x + v y)} du dv \quad (4.16)$$

The extension of the 1D DFT and its inverse to two dimensions is equally straight forward.

The DFT of a discrete function $f(x, y)$ (image) of size $M \times N$ is given by the equation:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j 2\pi (\frac{ux}{M} + \frac{vy}{N})} \quad (4.17)$$

where

$$u = 0, 1, \dots, M - 1 \text{ and}$$

$$v = 0, 1, \dots, N - 1.$$

Similarly, the inverse Fourier transform for two dimensions is given by

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j 2\pi (\frac{ux}{M} + \frac{vy}{N})} \quad (4.18)$$

where

$$x = 0, 1, \dots, M - 1 \text{ and}$$

$$y = 0, 1, \dots, N - 1.$$

The eq. (4.17) and (4.18) comprise the **2D discrete Fourier transform pair**.

- Variables:**
- x, y : **spatial or image variables**
 - u, v : **transform or frequency variables**

Definitions as in sec. 4.1.1:

- **Fourier spectrum:**

$$|F(u, v)| = \left[\operatorname{Re}\{F(u, v)\}^2 + \operatorname{Im}\{F(u, v)\}^2 \right]^{\frac{1}{2}} \quad (4.19)$$

- **Phase angle:**

$$\phi(u, v) = \arctan \frac{\operatorname{Im}\{F(u, v)\}}{\operatorname{Re}\{F(u, v)\}} \quad (4.20)$$

- **Power spectrum:**

$$\begin{aligned} P(u, v) &= |F(u, v)|^2 \\ &= \operatorname{Re}\{F(u, v)\}^2 + \operatorname{Im}\{F(u, v)\}^2 \end{aligned} \quad (4.21)$$

where $\operatorname{Re}\{F(u, v)\}$ and $\operatorname{Im}\{F(u, v)\}$: real and imaginary parts of $F(u, v)$, respectively.

Hint: Common practice to multiply the input image function by $(-1)^{x+y}$ prior to computing the Fourier transform:

$$\mathcal{F}\{f(x, y) \cdot (-1)^{x+y}\} = \mathcal{F}\left(u - \frac{M}{2}, v - \frac{N}{2}\right) \quad (4.22)$$

The above multiplication shifts the origin of $F(u, v)$ to the frequency coordinates $(\frac{M}{2}, \frac{N}{2})$ which is the center of $M \times N$ area occupied by the 2D DFT (“frequency rectangle”) (see subsection 4.1.3).

In order to guarantee that these shifted coordinates are integers, M and N have to be even.

Note: If $f(x, y)$ is real which applies to an image, its Fourier transform is conjugate symmetric:

$$F(u, v) = F^*(-u, -v) \quad (4.23)$$

It follows that the spectrum of the Fourier transform is symmetric:

$$|F(u, v)| = |F(-u, -v)| \quad (4.24)$$

→ simplifies the specification of circularly symmetric filters in the frequency domain.

According to eq. (4.14) follows:

$$\Delta u = \frac{1}{M \Delta x} \quad (4.25)$$

$$\Delta v = \frac{1}{N \Delta y} \quad (4.26)$$

4.1.3 Some Useful Theorems of the (2D) Fourier Transform

The following features of the Fourier transform are known from the system theory course.

They will be extended to 2D signals.

Translation

$$f(x, y) e^{j 2\pi \left(\frac{u_0 x}{M} + \frac{v_0 y}{N} \right)} \circledast F(u - u_0, v - v_0) \quad (4.27)$$

and

$$f(x - x_0, y - y_0) \circledast F(u, v) e^{-j 2\pi \left(\frac{u x_0}{M} + \frac{v y_0}{N} \right)} \quad (4.28)$$

The flash ([Loc]) shows the shifting property of the Fourier transform.

Assumption: $u_0 = \frac{M}{2}$ and $v_0 = \frac{N}{2}$

The exponential term in eq. (4.27) becomes:

$$\begin{aligned} e^{j2\pi\left(\frac{u_0x}{M} + \frac{v_0y}{N}\right)} &= e^{j\pi(x+y)} \\ &= (-1)^{x+y} \end{aligned} \quad (4.29)$$

The Fourier transform pair in the eqs. (4.27) and (4.28) become:

$$f(x, y)(-1)^{x+y} \circlearrowright F\left(u - \frac{M}{2}, v - \frac{N}{2}\right) \quad (4.30)$$

$$f\left(x - \frac{M}{2}, y - \frac{N}{2}\right) \circlearrowright F(u, v)(-1)^{u+v} \quad (4.31)$$

Note: The eq. (4.30) is the same as eq. (4.22) which has been used to center the transform.

Note: In a computer implementation the variables u and v will run from $u = 1$ to M and $v = 1$ to N instead of lying in the range $[0, M - 1]$ resp. $[0, N - 1]$.

Distributivity and Scaling

Distributivity

From the definition of the Fourier transform follows that

$$\mathcal{F}\{f_1(x, y) + f_2(x, y)\} = \mathcal{F}\{f_1(x, y)\} + \mathcal{F}\{f_2(x, y)\} \quad (4.32)$$

$$\mathcal{F}\{f_1(x, y) \cdot f_2(x, y)\} \neq \mathcal{F}\{f_1(x, y)\} \cdot \mathcal{F}\{f_2(x, y)\} \quad (4.33)$$

Interpretation: The Fourier transform is distributive over addition, but not over multiplication.

Scaling

Multiplication with scalars:

$$a f(x, y) \circlearrowright a F(u, v) \quad (4.34)$$

$$f(ax, by) \circlearrowright \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right) \quad (4.35)$$

The flash ([Loc]) shows the scaling property of the Fourier transform.

Rotation

With

$$x = r \cos\Theta, \quad y = r \sin\Theta, \quad u = \omega \cos\Phi, \quad v = \omega \sin\Phi$$

follows

$$f(x, y) \rightarrow f(r, \Theta)$$

$$F(u, v) \rightarrow f(\omega, \Phi)$$

Direct substitution yields:

$$f(r, \Theta + \Theta_0) \circlearrowleft F(\omega, \Phi + \Theta_0) \tag{4.36}$$

The effect of the rotation theorem has already been shown in fig. 4.11.

Periodicity and Conjugate Symmetry

Periodicity properties of the 2D Fourier transform:

$$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N) \quad (4.37)$$

And its inverse:

$$f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N) \quad (4.38)$$

The importance of periodicity is visualized in fig. 4.8.

Recapitulation for convenience:

$$\text{Eq. (4.23): } F(u, v) = F^*(-u, -v) \quad (4.39)$$

$$\text{Eq. (4.24): } |F(u, v)| = |F(-u, -v)| \quad (4.40)$$

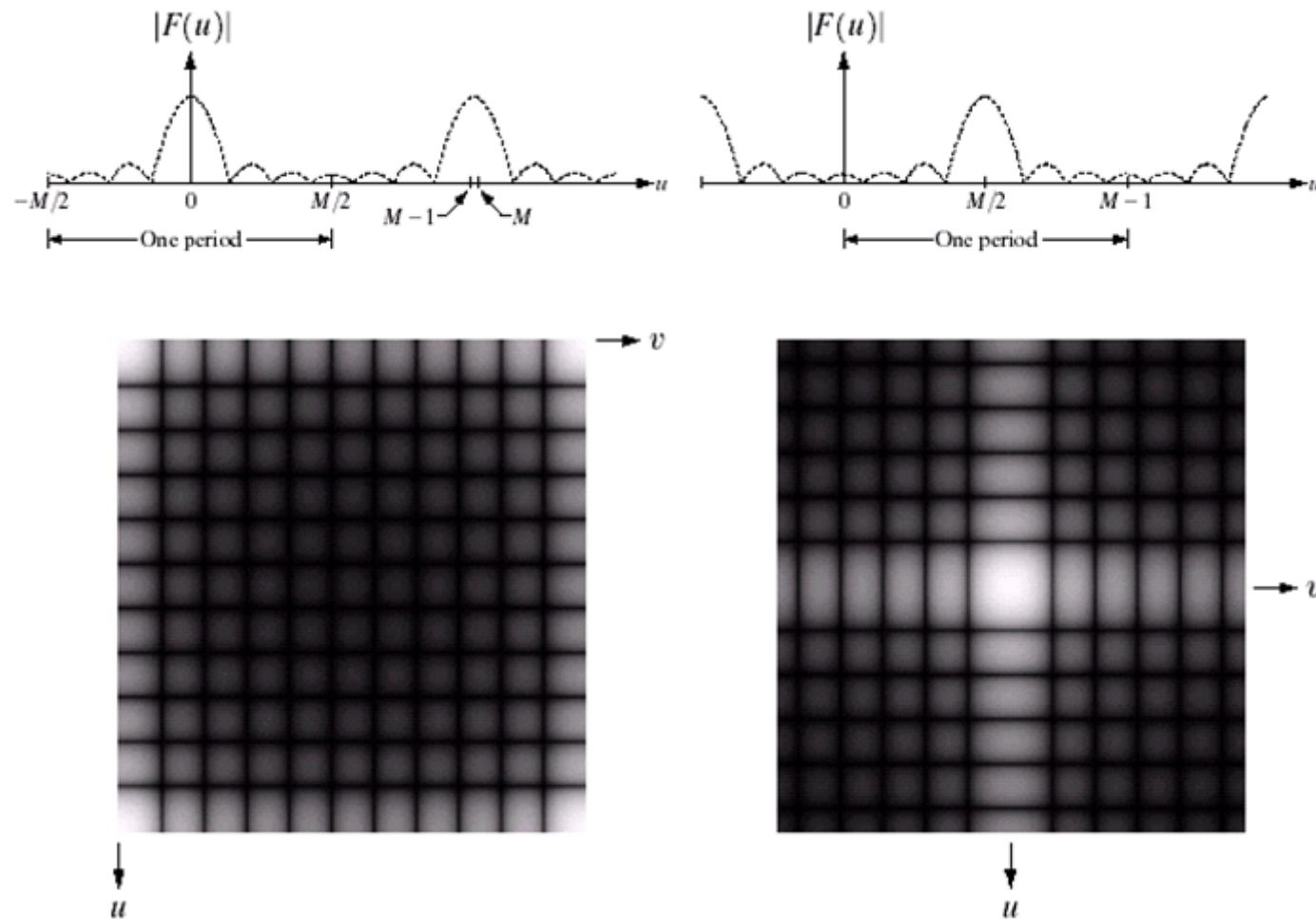


Fig. 4.8: (a) Fourier spectrum showing back-to-back half periods in the interval $[0, M - 1]$. (b) Shifted spectrum showing a full period in the same interval. (c) Fourier spectrum of an image, showing the same back-to-back properties as (a), but in two dimensions. (d) Centered Fourier spectrum. (From [GW18])

Example 4.5 Fourier Spectrum of a White Rectangle

Fig. 4.9 shows a white rectangle of size 20×40 pixels superimposed on a black background of size 512×512 pixels.

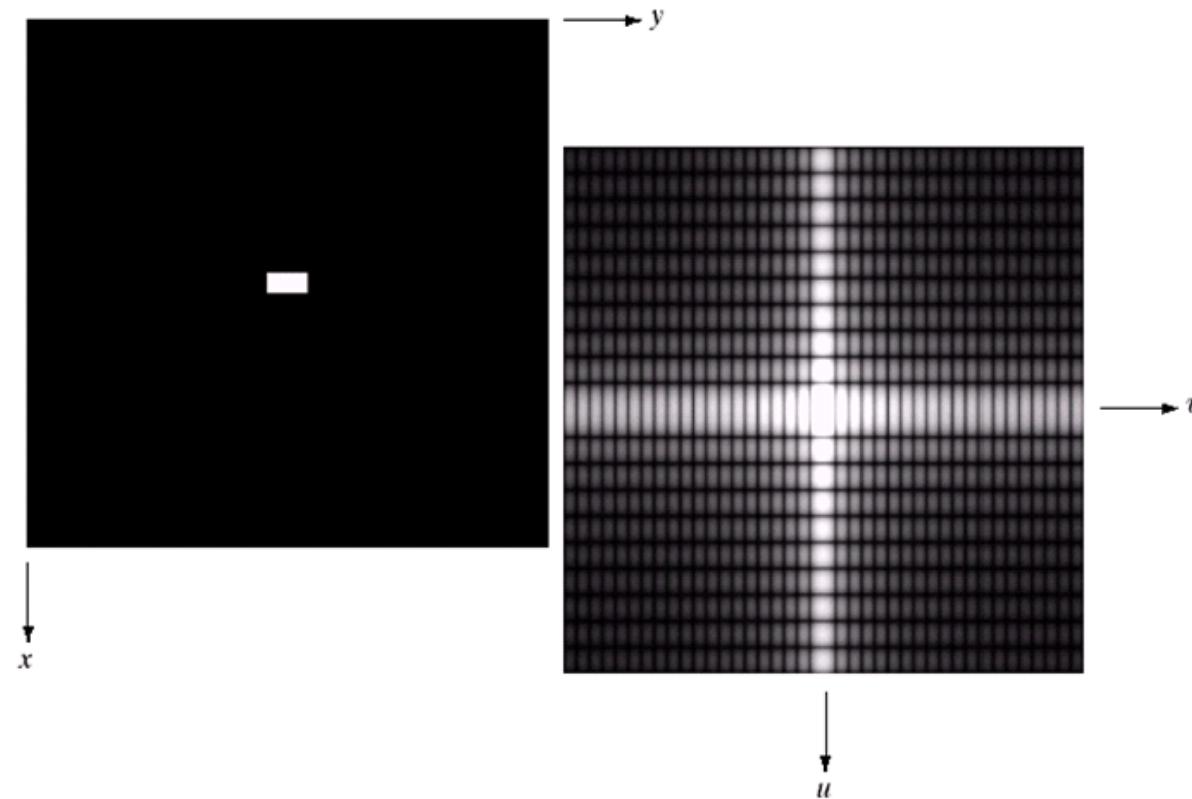


Fig. 4.9: (a) Image of a 20×40 white rectangle on a black background of size 512×512 pixels. (b) Centered Fourier spectrum shown after application of the log transformation. (From [GW18])

4.1.4 Correspondence between Filtering in the Spatial and Frequency Domains

Now: Introduction of link between spatial filters and their frequency domain counterparts

Most fundamental relationship between the spatial and frequency domains

→ **convolution theorem**

The basic concepts and mechanics of convolution in the spatial domain are described in section 3.4.

Formally, the discrete convolution of two functions $f(x, y)$ and $h(x, y)$ of size $M \times N$ is denoted by

$$g(x, y) = f(x, y) * h(x, y) \quad (4.41)$$

and defined by

$$g(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n) \quad (4.42)$$

- Interpretation I:**
- Eq. (4.42) is similar to eq. (3.20)
 - With following exceptions:
 - Leading constant,
 - Minus signs, and
 - Limits of the summation
 - Minus signs
→ function h is mirrored at the origin.

- Interpretation II:** Eq. (4.42) is an implementation for
1. Flipping one function about the origin,
 2. Shifting that function with respect to the other by changing the values of (x, y) , and
 3. Computing the sum of products over all values of m and n , for each displacement (x, y) .

The convolution theorem states two Fourier transform pairs:

$$1) \quad f(x, y) * h(x, y) \circlearrowleft \bullet F(u, v) \cdot H(u, v) \quad (4.43)$$

$$2) \quad f(x, y) \cdot h(x, y) \circlearrowleft \bullet F(u, v) * H(u, v) \quad (4.44)$$

Further concept for completing the tie between the spatial and frequency domains

→ **impulse function** of strength A , located at coordinates (x_0, y_0) and denoted by $A \delta(x - x_0, y - y_0)$.

Definition of the impulse function:

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} s(x, y) A \delta(x - x_0, y - y_0) = A s(x_0, y_0) \quad (4.45)$$

- Interpretation:**
- The summation of a function multiplied by an impulse is simply the value of the function at the location of the impulse.
 - $A \delta(x - x_0, y - y_0)$ is also an image of size $M \times N$, which is composed of all zeros, except at coordinates x_0, y_0 , where the value of the image is A .

Assumption: Either f or h in eq. (4.42) is an impulse function, located at the origin.

Using eq. (4.45) reveals that the convolution of a function with an impulse “copies” the value of that function at the location of the impulse:

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} s(x, y) \delta(x, y) = s(0, 0) \quad (4.46)$$

→ **Shifting property** of the impulse function.

If $f(x, y) = \delta(x, y)$, the Fourier transform of a unit impulse at the origin is using eq. (4.17):

$$\begin{aligned} F(u, v) &= \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \delta(x, y) e^{-j 2\pi \left(\frac{ux}{M} + \frac{vy}{N}\right)} \\ &= \frac{1}{MN} \end{aligned} \quad (4.47)$$

- Interpretation:**
- The Fourier transform of an impulse at the origin of the spatial domain is a *real* constant (i. e. the phase angle is zero).
 - If the impulse were located elsewhere, the transform would have complex components (i. e. same magnitude, but a nonzero phase angle in the transform).

Let $f(x, y) = \delta(x, y)$ and carry out the convolution defined in eq. (4.42):

$$\begin{aligned} g(x, y) &= f(x, y) * h(x, y) \\ &= \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \delta(m, n) h(x - m, y - n) \end{aligned}$$

With eq. (4.46) follows:

$$= \frac{1}{MN} h(x, y) \quad (4.48)$$

Combining the results in eq. (4.47) and (4.48) with eq. (4.43) follows:

$$\delta(x, y) * h(x, y) \circlearrowright F\{\delta(x, y)\} H(u, v) \quad (4.49)$$

$$h(x, y) \circlearrowright H(u, v) \quad (4.50)$$

Interpretation: Using only the properties of the impulse function and the convolution theorem, it can be established that filters in the spatial and frequency domain constitute a Fourier transform pair.

Note: All functions in this section are of the same size, $M \times N$.

- Interpretation:**
- Specifying a filter in the frequency domain and then taking the inverse transform to compute an equivalent spatial domain filter is not useful from a computational point of view.
 - If both filters are of the same size, it is generally more efficient to do the filtering in the frequency domain (see last section in this chapter).
 - When small filters are used, filtering in the spatial domain is more efficient.
 - But: Filtering is often more intuitive in the frequency domain.

Particular important: **Gaussian functions**

Here: For simplification: only 1D function is regarded.

Their shapes are easily specified and the DFT and DFT^{-1} of a Gaussian function are **real** Gaussian functions.

Gaussian filter function in the frequency domain:

$$H(u) = A e^{-\frac{u^2}{2\sigma^2}} \quad (4.51)$$

where σ : standard deviation

The corresponding filter in the spatial domain is:

$$h(x) = \sqrt{2\pi}\sigma A e^{-2\pi^2 \sigma^2 x^2} \quad (4.52)$$

Interpretation: • $H(u)$ and $h(x)$ are both Gaussian *and* real functions.

- They form a Fourier transform pair.
- The function behave reciprocally with respect to another.
- Similar behavior which was depicted in fig. 4.7 and 4.9.

Fig. 4.10(a) shows a Gaussian filter in the frequency domain (lowpass filter).

The corresponding filter in the spatial domain to the filter in 4.10(a) is shown in fig. 4.10(c).

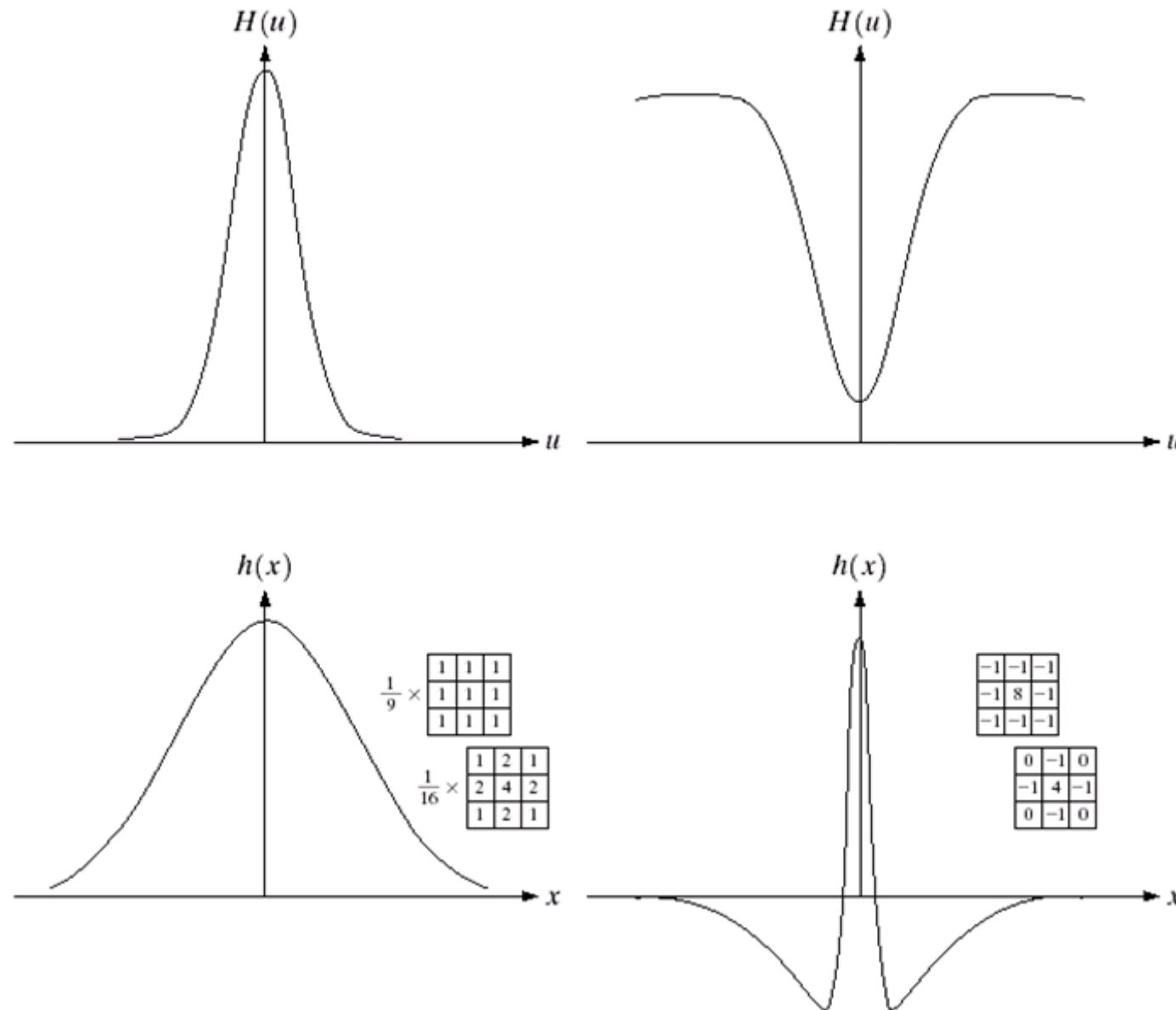


Fig. 4.10: (a) Gaussian frequency domain lowpass filter. (b) Gaussian frequency domain highpass filter. (c) Corresponding lowpass spatial filter. (d) Corresponding highpass spatial filter. (From [GW18])

More complex filters can be constructed with the combination of basic Gaussian functions:

Using eq. (4.51) for a highpass filter as a difference of Gaussian:

$$H(u) = A e^{-\frac{u^2}{2\sigma_1^2}} - B e^{-\frac{u^2}{2\sigma_2^2}} \quad (4.53)$$

with

$$A \geq B \text{ and}$$

$$\sigma_1 > \sigma_2$$

Corresponding filter in the spatial domain:

$$h(x) = \sqrt{2\pi} \left[\sigma_1 A e^{-2\pi^2 \sigma_1^2 x^2} - \sigma_2 B e^{-2\pi^2 \sigma_2^2 x^2} \right] \quad (4.54)$$

The plots of the highpass filter in the two spaces are shown in fig. 4.10(b) and (d).

4.1.5 Filtering in the Frequency Domain

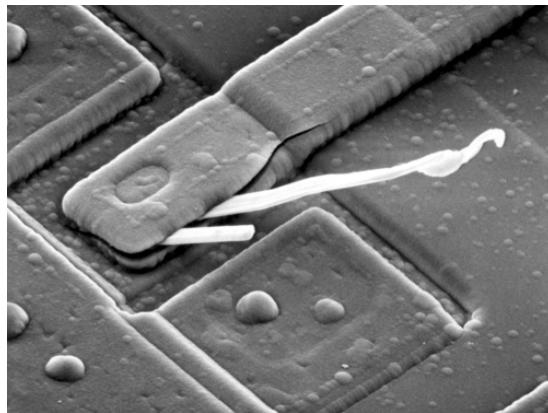
Interpretation of the Fourier Spectrum:

- Each term of $F(u, v)$ contains *all* values of $f(x, y)$, modified by values of exponential terms.
- The slowest varying frequency component ($u = v = 0$) corresponds to the average gray level of an image.
- The higher frequencies correspond to fast gray level changes in the image.

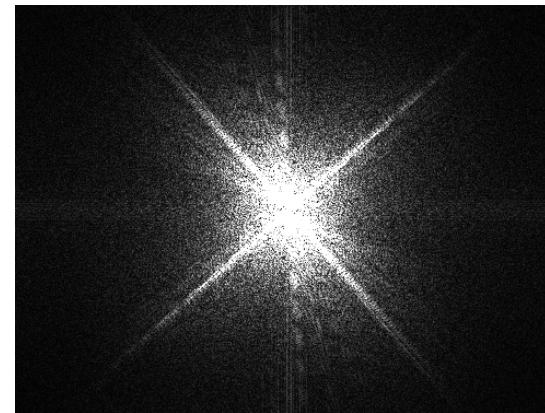
Example 4.6 Interpretation of the Fourier Spectrum

Fig. 4.11(a) shows a scanning electron microscope image (magnified ~ 2500 times) with dominant diagonal structures, fig. 4.11(c) the corresponding image rotated by 90° .

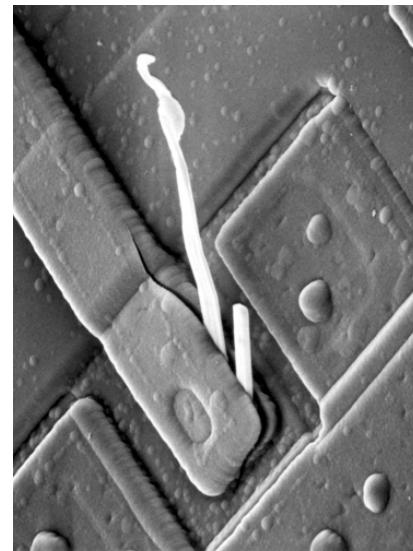
- Interpretation:**
- Prominent components along 45° directions that correspond to the edges of the main image structures.
 - Vertical component that is off-axis slightly to the left, is caused by the edges of the oxide protrusions (see fig. 4.11(b)).



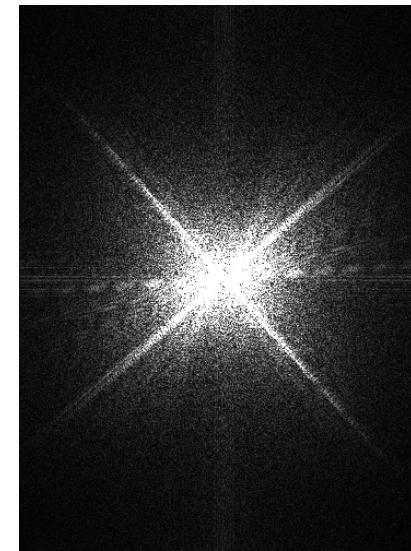
(a)



(b)



(c)



(d)

Fig. 4.11: (a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (c) Image (a) rotated around 90° . (d) Fourier spectrum of (c). (Adapted from [GW18])

Basics of filtering in the frequency domain

In sec. 3.4 spatial filtering was introduced.

In the frequency domain, filtering is quite straight forward:

1. Multiply the input image by $(-1)^{x+y}$ to center the transform.
2. Compute $F(u, v)$, the DFT, of the image from 1.
3. Multiply $F(u, v)$ by a **filter function** $H(u, v)$.
4. Compute the inverse DFT of the result in 3.
5. Obtain the real part of the result in 4.
6. Multiply the result in 5. by $(-1)^{x+y}$.

Hint: $H(u, v)$ is called a **filter transfer function** or short a **filter** because it suppresses certain frequencies in the input image while leaving others unchanged.

In step 3, the Fourier transform of the output image is given by

$$G(u, v) = H(u, v) F(u, v) \quad (4.55)$$

Important: Multiplication of H and F is done on an element-by-element basis:

- In general: Components of H are complex quantities
- Here: Real quantities

Filtered image:

$$g(x, y) = \text{DFT}^{-1}\{G(u, v)\} \quad (4.56)$$

Fig. 4.12 illustrates the basic filtering steps.

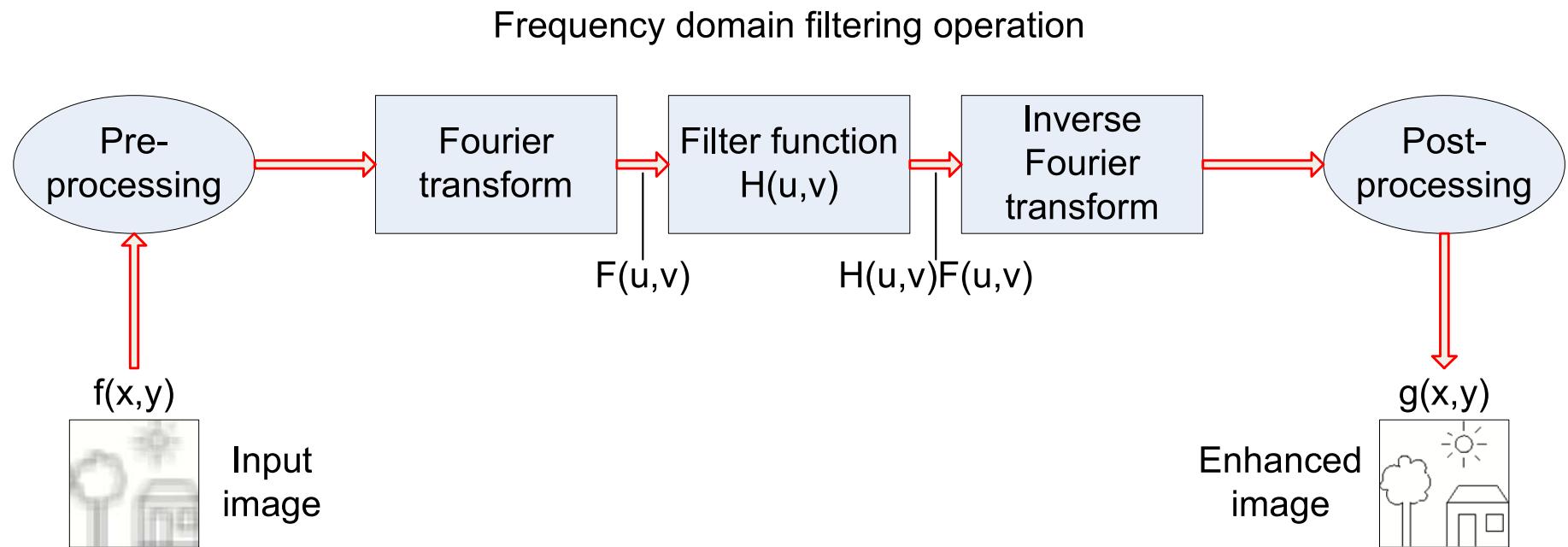


Fig. 4.12: Basic steps for filtering frequency domain.

Some basic filters and their properties

Notch filter: Forces the average value of an image to zero.

From eq. (4.12) follows for the average of $f(x, y)$:

$$F(0, 0) = \frac{1}{M N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \quad (4.57)$$

Notch filtering can be done by multiplying all values of $F(u, v)$ by the filter function:

$$H(u, v) = \begin{cases} 0 & \text{if } (u, v) = (\frac{M}{2}, \frac{N}{2}) \\ 1 & \text{otherwise} \end{cases} \quad (4.58)$$

The result of processing the image in fig. 4.11(a) with this filter is shown in fig. 4.13.

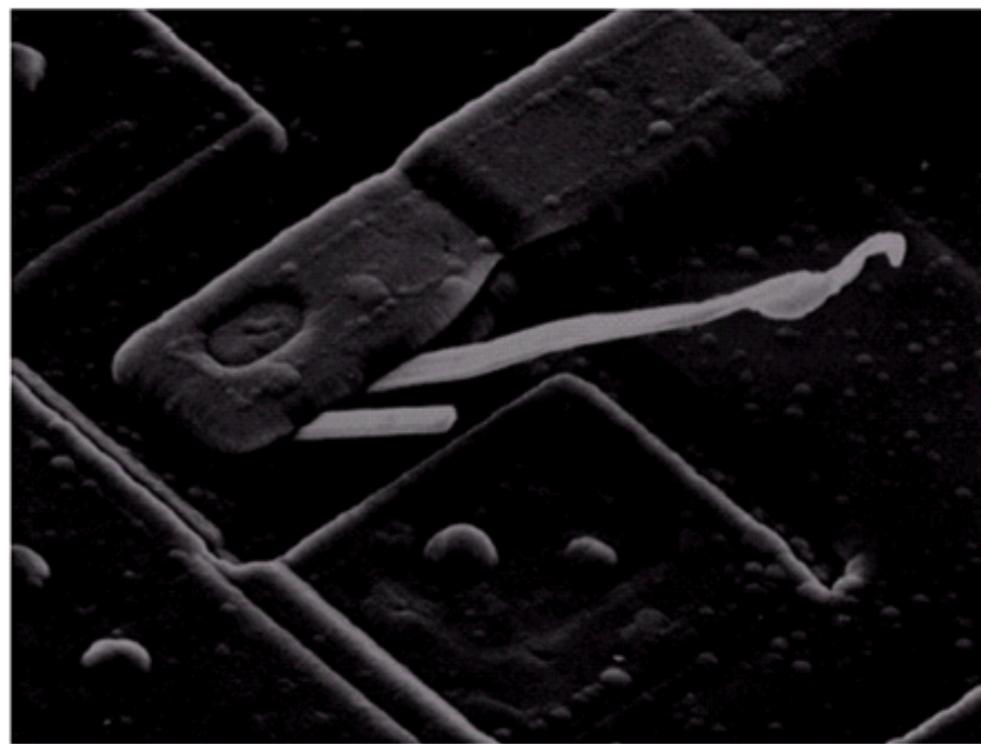


Fig. 4.13: Result of filtering the image in fig. 4.11(a) with a notch filter (From [GW18])

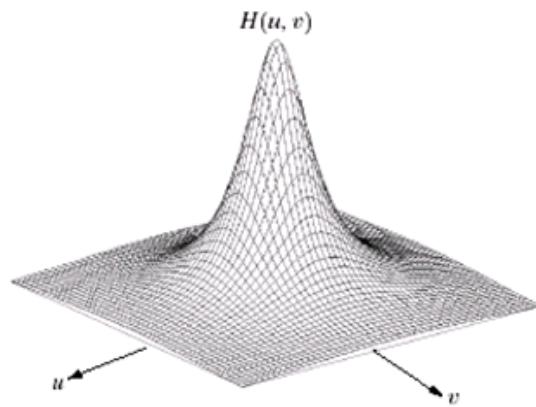
Lowpass filter: Attenuates high spatial frequencies in an image while low frequencies can pass

For an example see fig. 4.14(a) and (b).

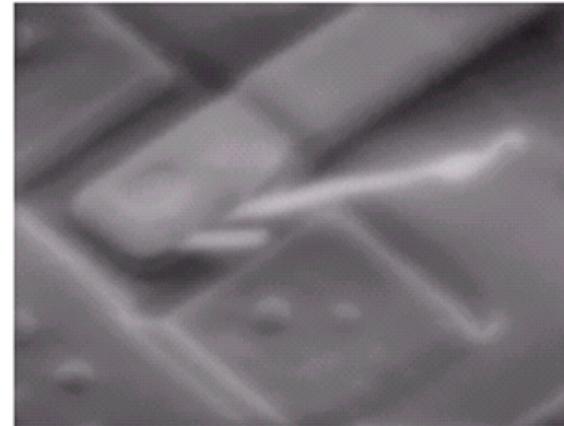
Highpass filter: Filter with the opposite characteristic.

For an example see fig. 4.14(c) and (d).

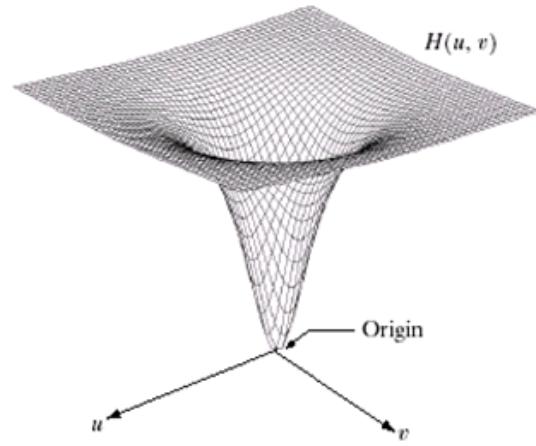
Both filters shown in fig. 4.14(a) and 4.14(c), respectively are circularly symmetric.



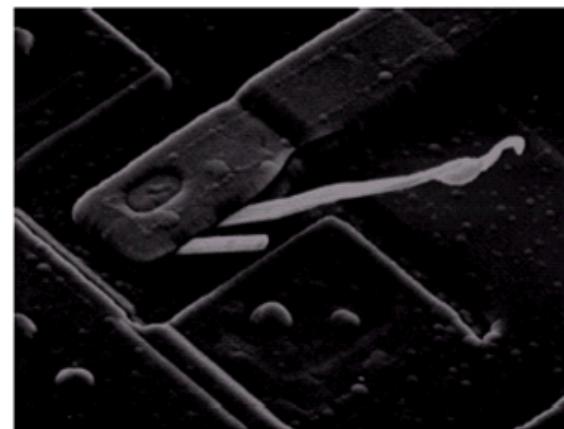
(a)



(b)



(c)



(d)

Fig. 4.14: (a) A 2D lowpass filter function. (b) Result of lowpass filtering the image in fig. 4.11(a). (c) A 2D highpass filter function. (d) Result of highpass filtering the image in fig. 4.11(a). (From [GW18])

An improvement of the result of highpass filtering is possible by adding a constant.

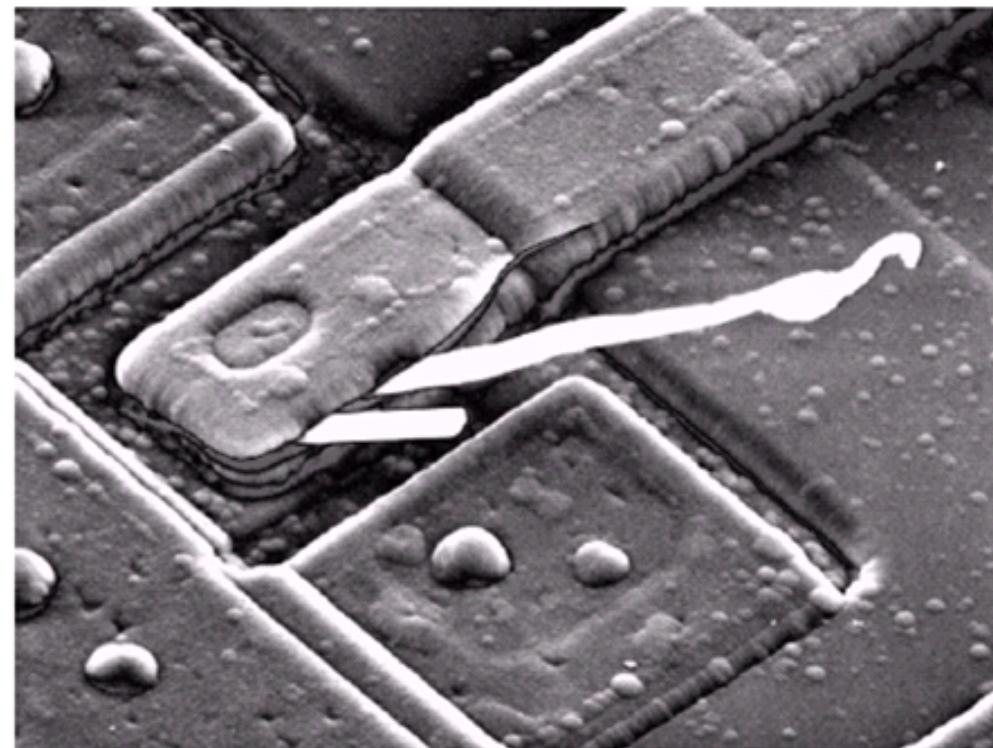


Fig. 4.15: Results of highpass filtering the image in fig. 4.11(a) with the filter in fig. 4.14(c), modified by adding a constant one half the filter height to the filter function. Compare to fig. 4.11(a). (From [GW18])

4.2 Smoothing Frequency Domain Filter

Basic “model” for filtering in the frequency domain is given in eq. (4.55):

$$G(u, v) = H(u, v)F(u, v) \quad (4.59)$$

Now: Consideration of three types of lowpass filters:

- Ideal lowpass filter,
- Butterworth filter, and
- Gaussian filter.

These filters differ how they handle the transition from low to high frequencies.

4.2.1 Ideal Lowpass Filter

Simplest lowpass filter: All high-frequency components are “cut off” at a distance greater than a specified distance D_0 from the origin of the (centered) transform.

Transfer function of an **ideal lowpass filter (ILPF)**:

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases} \quad (4.60)$$

where

$D(u, v)$: distance from point (u, v) to the center, and

D_0 : non-negative quantity

Distance from any point (u, v) to the center (origin) of the Fourier transform:

$$D(u, v) = \left[\left(u - \frac{M}{2} \right)^2 + \left(v - \frac{N}{2} \right)^2 \right]^{\frac{1}{2}} \quad (4.61)$$

Fig. 4.16 shows different perspectives of $H(u, v)$.

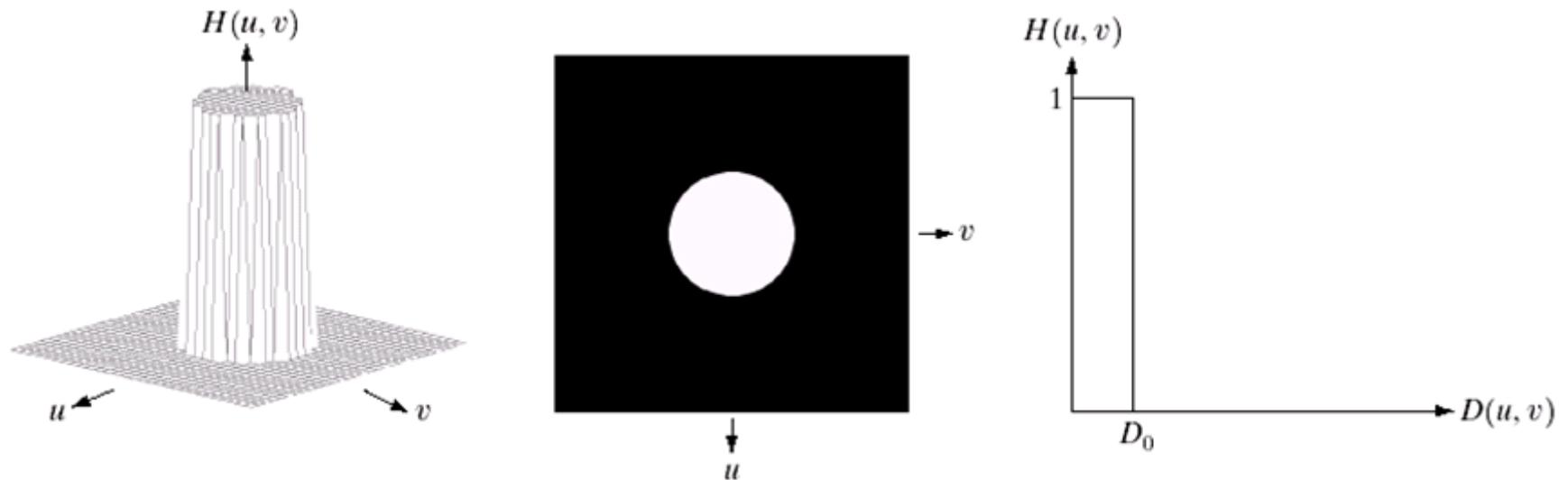


Fig. 4.16: (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section. (From [GW18])

The transition between $H(u, v) = 1$ and $H(u, v) = 0$ is called the **cutoff frequency**.

Note: ILPFs can be implemented in a computer, but cannot be realized with electronic components.

The filters in this section are compared by studying their behavior as a function of the same cutoff frequency.

One approach: Registration of amounts of total image power P_T at different frequency loci.

Total image power:

$$P_T = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} P(u, v) \quad (4.62)$$

where $P(u, v)$ is given in eq. (4.21)

If the transform has been centered, a circle of radius r with origin at the center of the frequency rectangle encloses α percent of the power, where

$$\alpha = 100 \left[\sum_u \sum_v \frac{P(u, v)}{P_T} \right] \quad (4.63)$$

The summation in eq. (4.63) is done over the values of (u, v) that lie inside the circle or on its boundary.

In fig. 4.17 the percentage of the image power is demonstrated for circles of different radii, while fig. 4.18 demonstrates the results of applying ideal lowpass filters with cutoff frequencies at the radii shown in fig. 4.17(b).

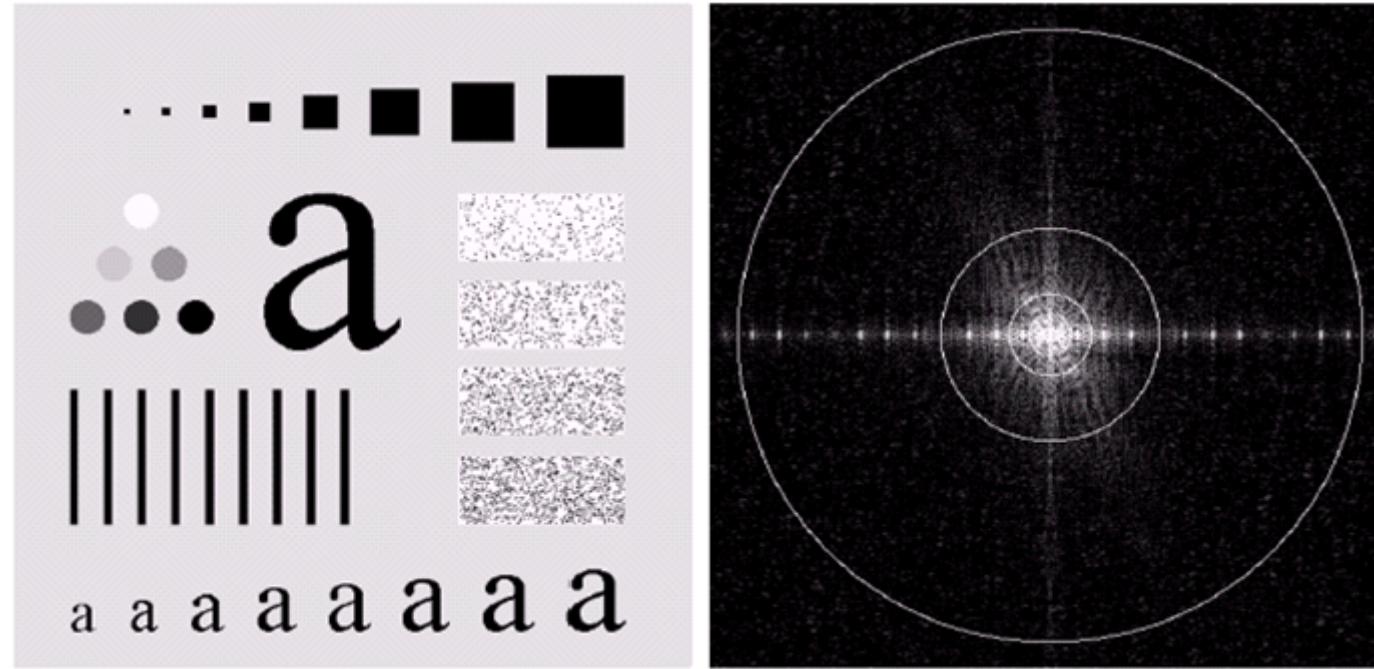


Fig. 4.17: (a) An image of size 500×500 pixels. (b) Fourier spectrum of (a). The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively. (From [GW18])

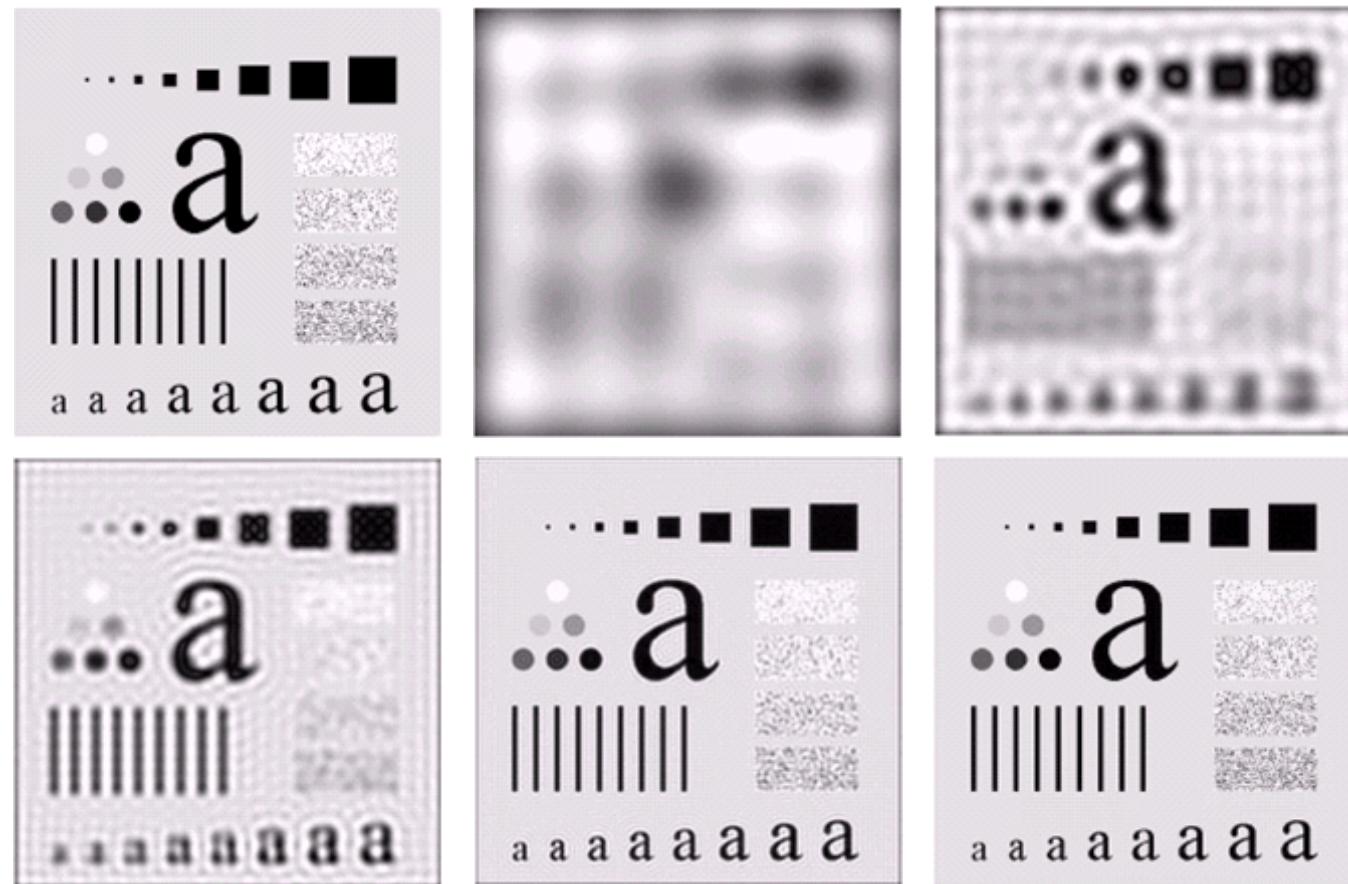


Fig. 4.18: (a) Original image. (b)-(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in fig. 4.17(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively. (From [GW18])

Note: The images in fig. 4.18(c) through (e) are characterized by “ringing”.

Interpretation: Ideal lowpass filtering is not very practical.

Fig. 4.19 helps to understand the “ringing” and the blurring as result of the convolution process in the spatial domain.

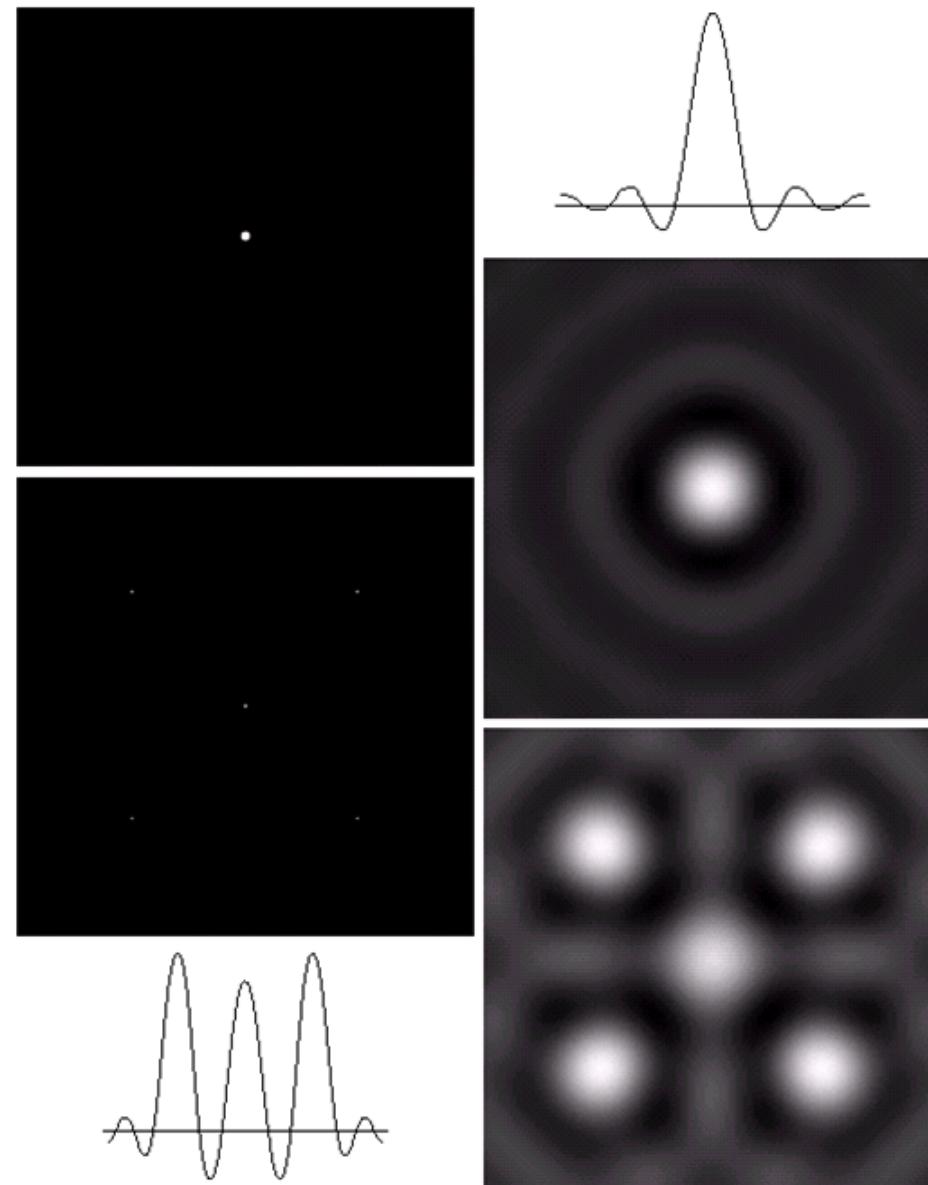


Fig. 4.19: (a) Frequency domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain. (From [GW18])

4.2.2 Butterworth Lowpass Filter

Transfer function of a **Butterworth lowpass filter (BLPF)** of order n and with cutoff frequency at a distance D_0 from the origin:

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0} \right]^{2n}} \quad (4.64)$$

where $D(u, v)$ is given by eq. (4.61)

Fig. 4.20 shows different views of $H(u, v)$.

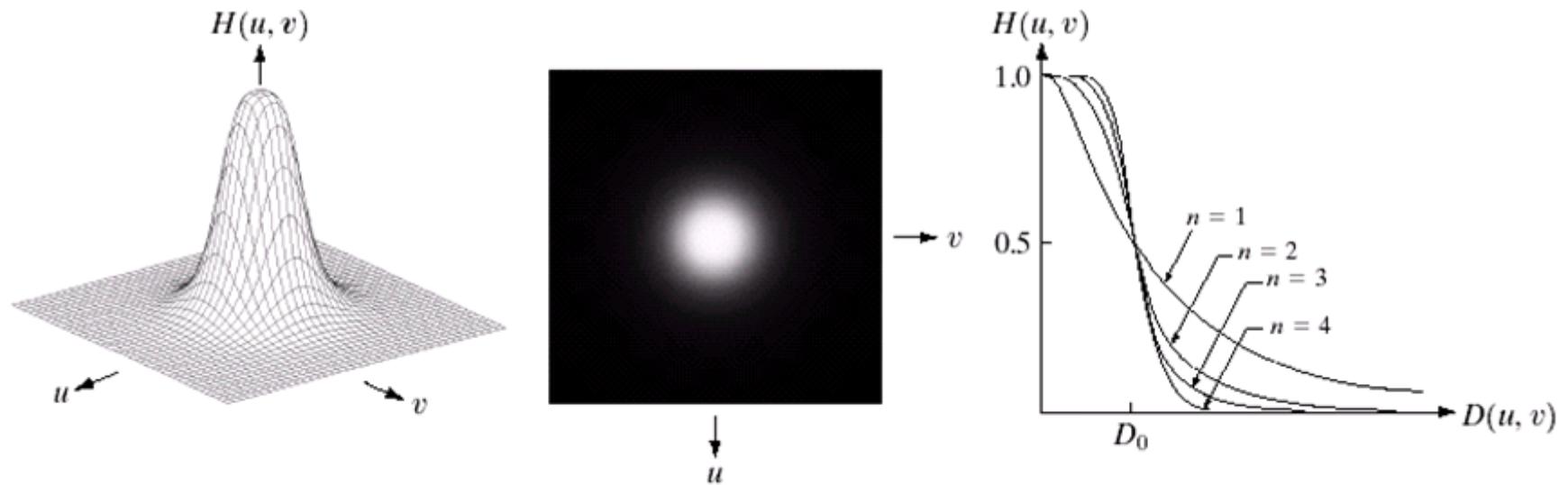


Fig. 4.20: (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4. (From [GW18])

Note: Unlike the ILPF, the BLPF does not have a sharp transition between passed and filtered frequencies.

The cutoff frequency is defined at points for which $H(u, v)$ is down to a certain fraction of its maximum value (here: $H(u, v) = 0.5$ when $D(u, v) = D_0$).

Fig. 4.21 shows the result of applying the BLPF to the test image, with $n = 2$ and D_0 equal to the five radii shown in fig. 4.17(b).

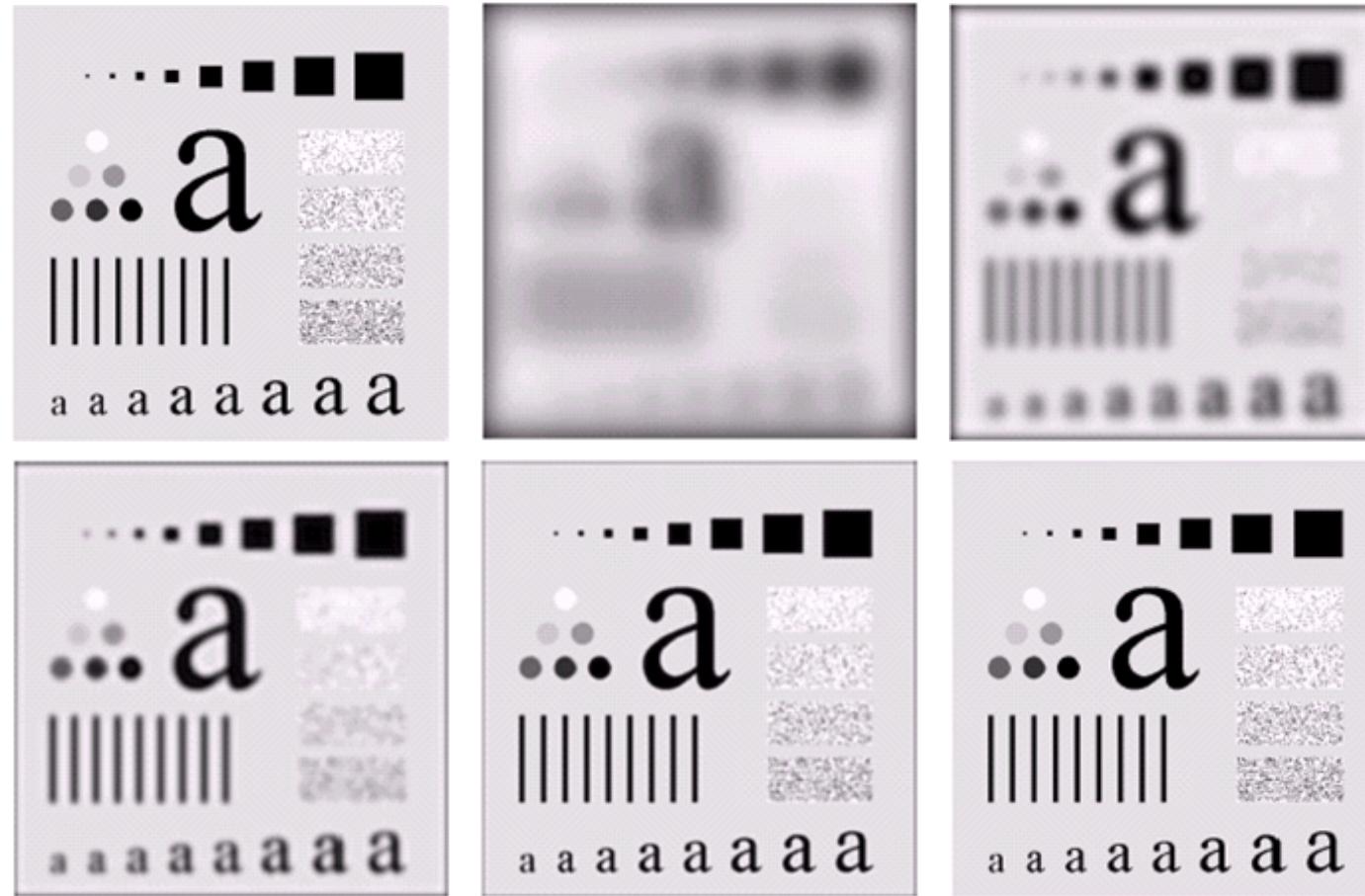


Fig. 4.21: (a) Original image. (b)-(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in fig. 4.17(b). (From [GW18])

- Interpretation:**
- Smooth transition in blurring as a function of increasing cutoff frequency
 - No ringing visible

Note: Ringing can become significant in filters of higher order.

Interpretation: In general, BLPFs of order $n = 2$ are a good compromise between effective lowpass filtering and acceptable ringing characteristics.

4.2.3 Gaussian Lowpass Filter

Transfer function of a **Gaussian Lowpass Filter (GLPF)**:

$$H(u, v) = e^{-\frac{D^2(u,v)}{2\sigma^2}} \quad (4.65)$$

where $D(u, v)$ is given by eq. (4.61)

Hint: No constant is used in eq. (4.65) in front of the filter function in order to be consistent with all the other filters in section 4.2 (value of 1 at the origin!).

Letting $\sigma = D_0$, the transfer function gets a more familiar form:

$$H(u, v) = e^{-\frac{D^2(u,v)}{2D_0^2}} \quad (4.66)$$

where D_0 is the cutoff frequency.

If $D(u, v) = D_0$, the filter is down to 0.667 of its maximum value.

Fig. 4.22 shows different views of $H(u,v)$.

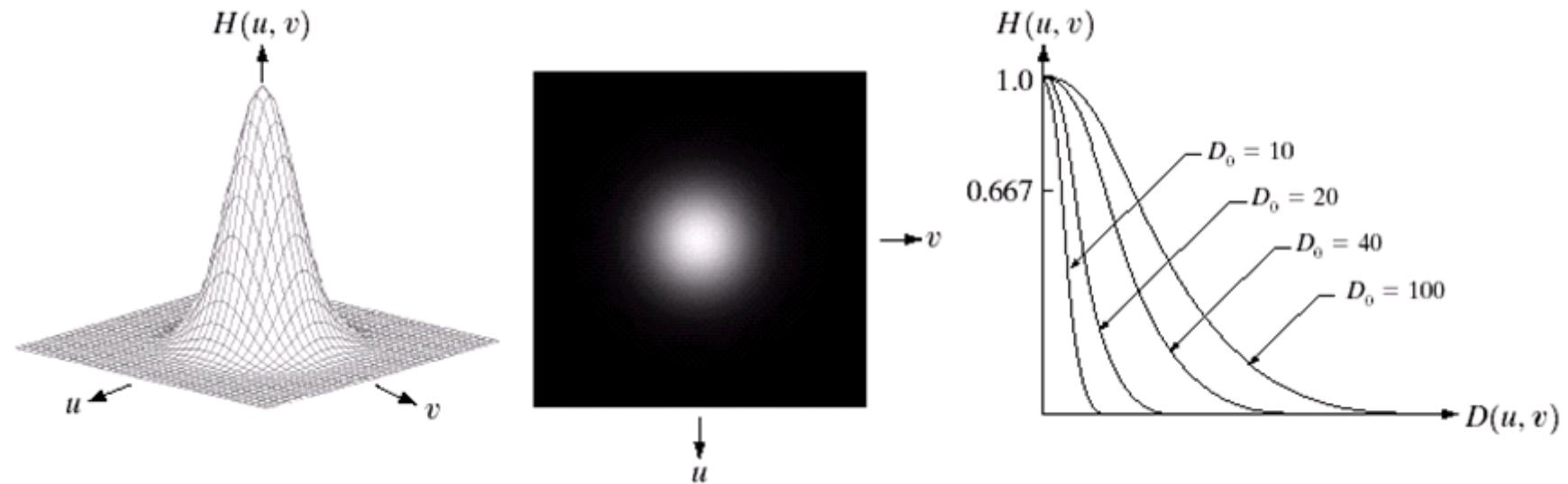


Fig. 4.22: (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0 . (From [GW18])

Fig. 4.23 shows the results of applying the GLPF of eq. (4.66) to the test image, with D_0 equal to the five radii shown in fig. 4.17(b).

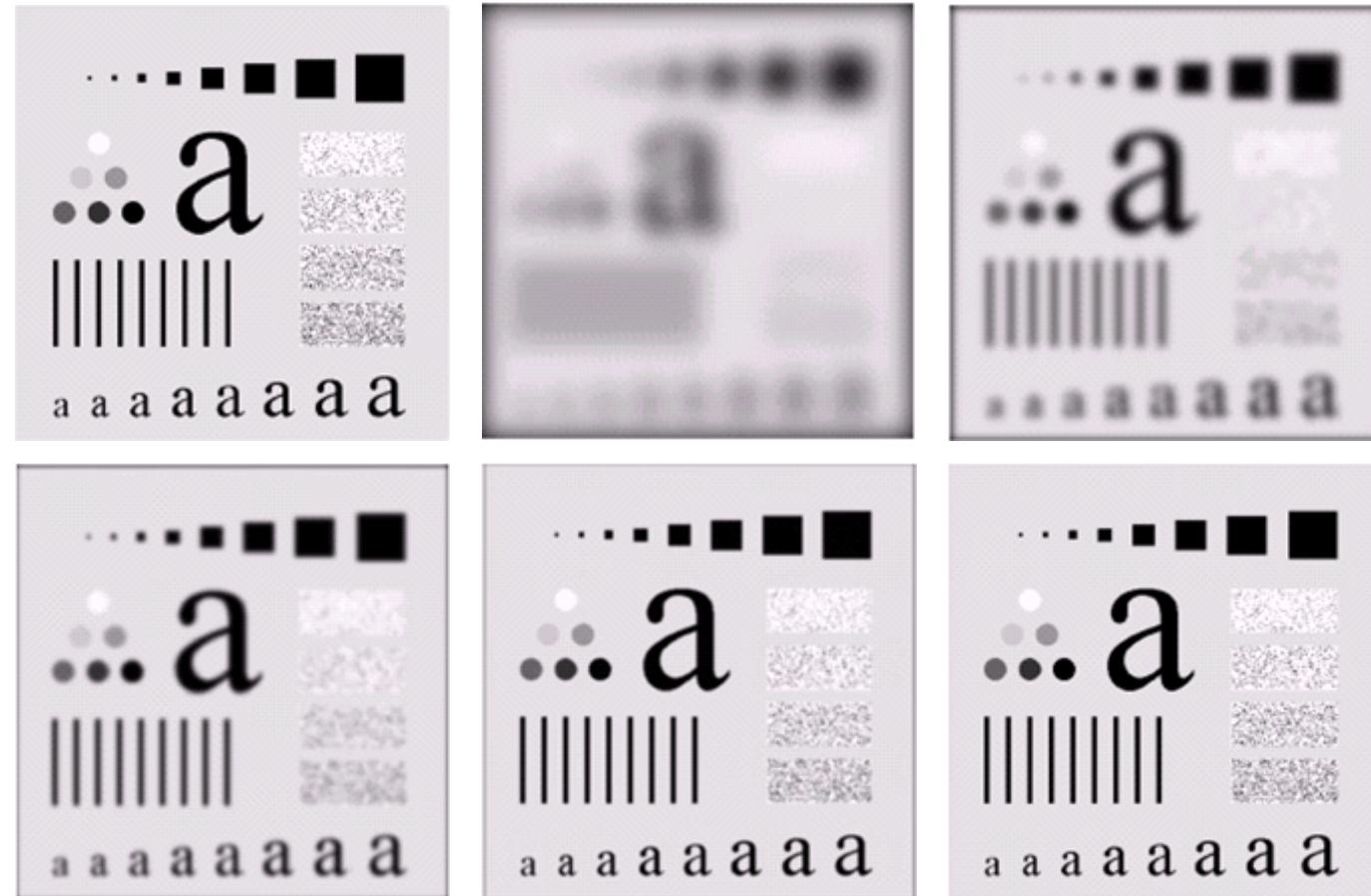


Fig. 4.23: (a) Original image. (b)-(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radius values of 5, 15, 30, 80, and 230, as shown in fig. 4.17(b). (From [GW18])

- Interpretation:**
- Here again: smooth transition in blurring as a function on increasing cutoff frequency.
 - Less smoothing achieved in comparison to a BLPF of order $n = 2$ (see e. g. fig. 4.21(c) vs. fig. 4.23(c)).
 - No ringing possible
→ important in applications where any type of artifact is not acceptable (e. g. in medical imaging)

4.2.4 Additional examples of Lowpass Filtering

In the fig. 4.24 and 4.25, more practical applications of lowpass filtering are shown.

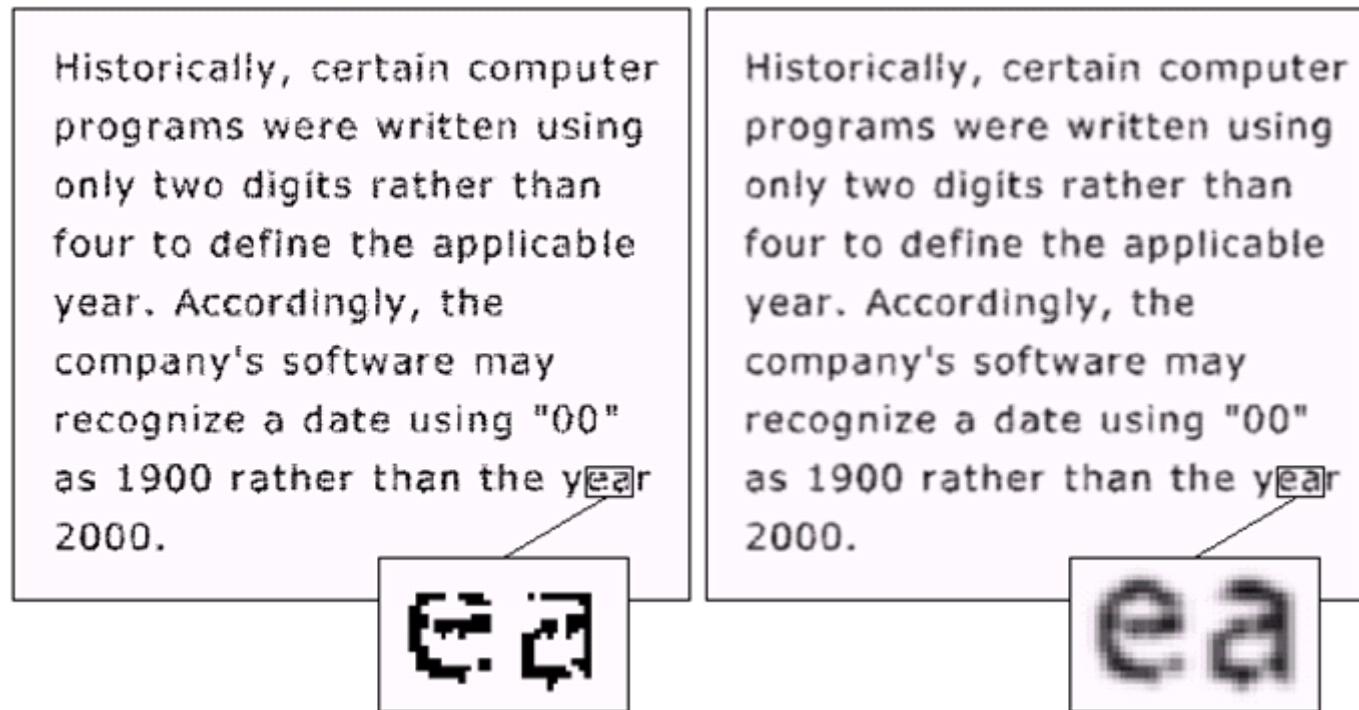


Fig. 4.24: (a) Sample text of poor resolution (note broken characters in magnified view). (b) Results of filtering with a GLPF (broken character segments were joined). (From [GW18])



Fig. 4.25: (a) Original image (1078×732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c). (From [GW18])

4.3 Sharpening Frequency-Domain Filters

Image sharpening can be achieved in the frequency domain by highpass filtering.

According to sec. 4.2: Consideration of

- Zero-phase-shift filters that are radially symmetric
- Three basic filter types:
 - Ideal highpass filter
 - Butterworth highpass filters, and
 - Gaussian highpass filters.

The transfer functions of the filters are calculated by using the relation:

$$H_{\text{HP}}(u, v) = 1 - H_{\text{LP}}(u, v) \quad (4.67)$$

where $H_{\text{LP}}(u, v)$: transfer function of corresponding lowpass filter

Fig. 4.26 shows different views of the three highpass filters.

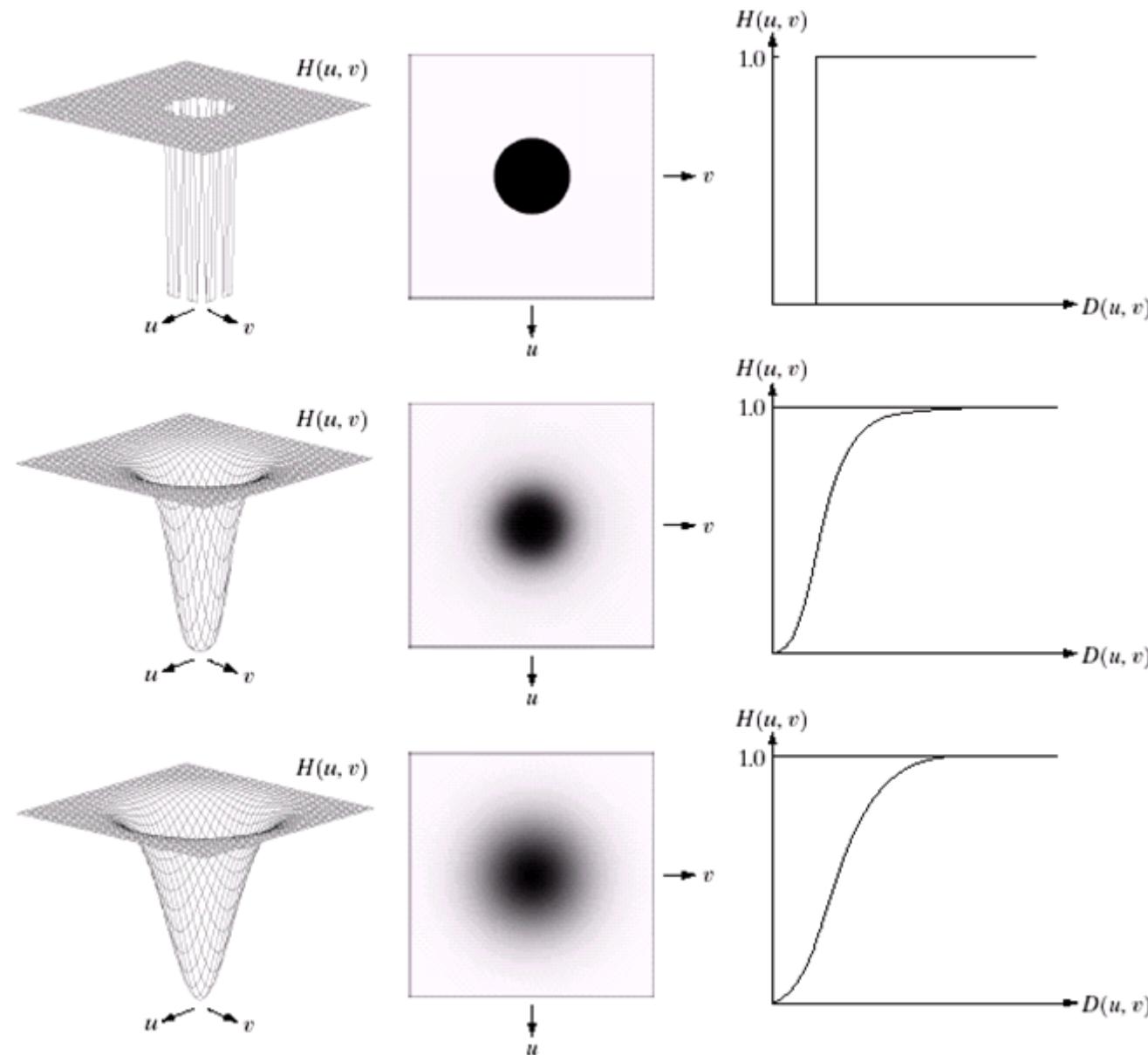


Fig. 4.26: Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters. (From [GW18])

The spatial representation of a frequency domain filter is obtained by:

1. Multiplying $H(u, v)$ by $(-1)^{u+v}$ for centering,
2. Computing the inverse DFT, and
3. Multiplying the real part of the inverse DFT by $(-1)^{x+y}$.

Fig. 4.27 illustrates what these filters look like in the spatial domain.

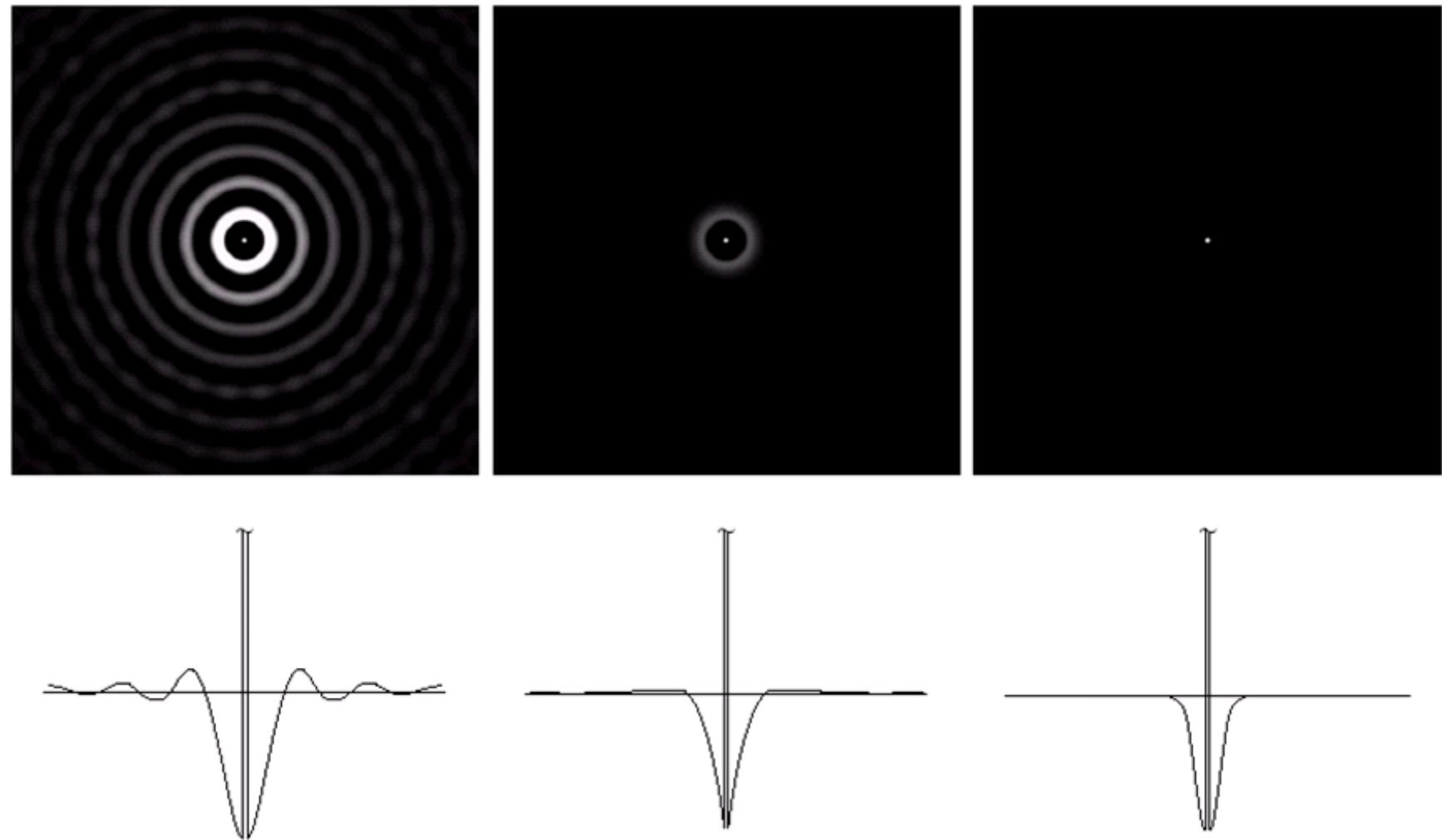


Fig. 4.27: Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles. (From [GW18])

4.3.1 Ideal Highpass Filters

Transfer function of ideal highpass filters (IHPF):

$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases} \quad (4.68)$$

where

D_0 : cutoff distance measured from the origin of the frequency rectangle

$D(u, v)$ is given by eq. (4.61)

Fig. 4.28 shows the results of IHPFs using the original image in fig. 4.17(a) with D_0 set to 15, 30, and 80 pixels, respectively.

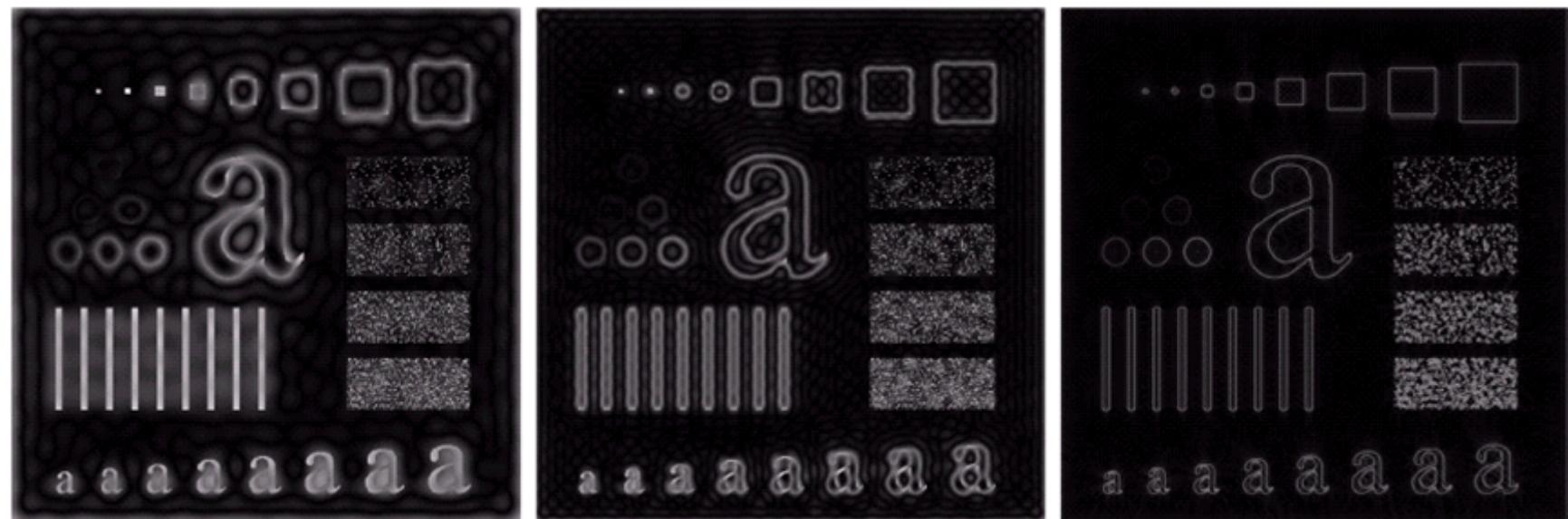


Fig. 4.28: Results of ideal highpass filtering the image in fig. 4.17(a) with $D_0=15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b). (From [GW18])

Note: IHPFs produce the same "ringing" effects than ILPFs.

4.3.2 Butterworth Highpass Filters

Transfer function of the Butterworth highpass filter (BHPF) of order n and with cutoff frequency at a distance D_0 from the origin:

$$H(u, v) = \frac{1}{1 + \left[\frac{D_0}{D(u, v)} \right]^{2n}} \quad (4.69)$$

where $D(u, v)$ is given by eq. (4.61)

Fig. 4.29 shows the results of a BHPF of order 2, using the same parameter as in fig. 4.28.

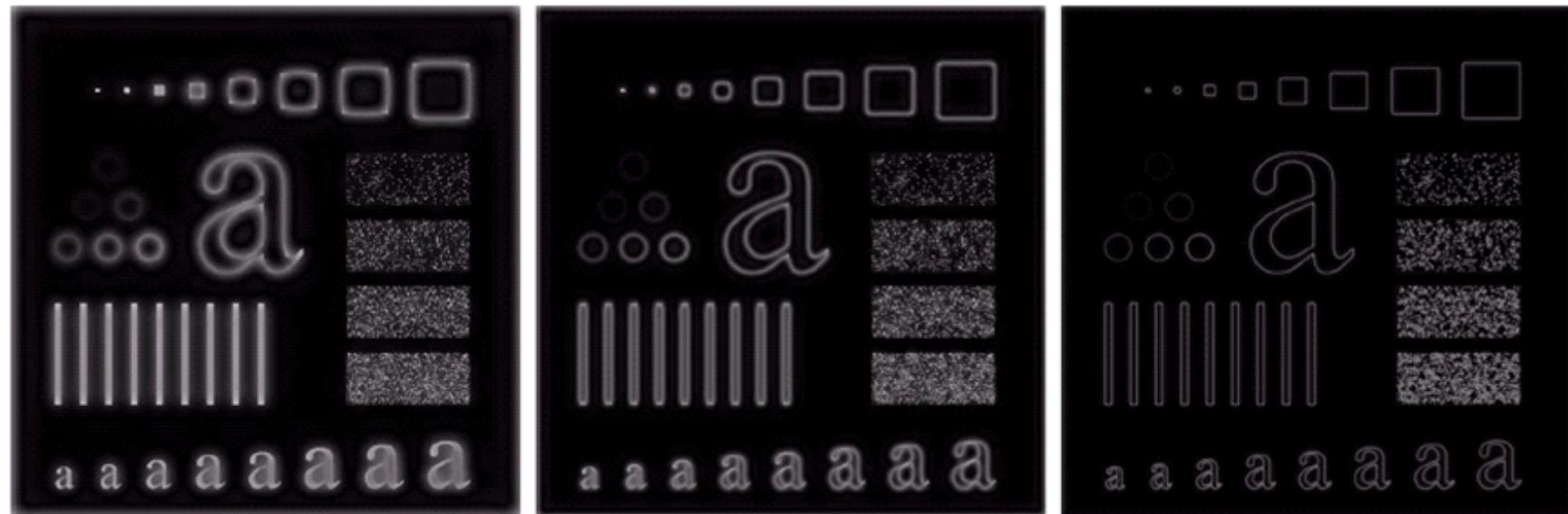


Fig. 4.29: Results of highpass filtering the image in fig. 4.17(a) using a BHPF of order 2 with $D_0=15$, 30, and 80, respectively. These results are much smoother than those obtained with an HLPF. (From [GW18])

Interpretation: Smoother behavior than IHPFs

4.3.3 Gaussian Highpass Filters

Transfer function of the Gaussian highpass filter (GHPF) with cutoff frequency D_0 :

$$H(u, v) = 1 - e^{-\frac{D^2(u,v)}{2D_0^2}} \quad (4.70)$$

Fig. 4.30 shows the results of the GHPF using the same parameters as in fig. 4.28.



Fig. 4.30: Results of highpass filtering the image in fig. 4.17(a) using a GHPF of order 2 with $D_0=15$, 30, and 80, respectively. (From [GW18])

Interpretation:

- Results are smoother than with the previous two filters
- The filtering of small objects (e. g. the squares) and the thin bars is clearer

4.4 Implementation of the Fourier Transform

4.4.1 Separability

The discrete 2D Fourier transform in eq. (4.17) can be written in the separable form:

$$\begin{aligned}
 F(u, v) &= \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j 2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)} \\
 &= \frac{1}{M} \sum_{x=0}^{M-1} e^{-j 2\pi ux/M} \frac{1}{N} \sum_{y=0}^{N-1} f(x, y) e^{-j 2\pi vy/N} \\
 &= \frac{1}{M} \sum_{x=0}^{M-1} F(x, v) e^{-j 2\pi ux/M}
 \end{aligned} \tag{4.71}$$

where

$$F(x, v) = \frac{1}{N} \sum_{y=0}^{N-1} f(x, y) e^{-j 2\pi vy/N}$$

Important: The 2D Fourier transform can be implemented by first computing a 1D transform along each row of the input image, and then compute a 1D transform along each column of this intermediate result (see 4.31).

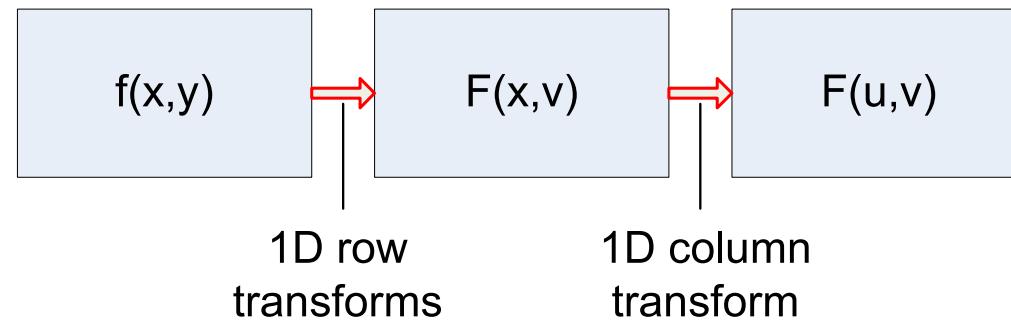


Fig. 4.31: Computation of the 2-D Fourier transform as a series of 1-D transforms.

The equivalent holds for the inverse Fourier transform.

4.4.2 Implementation of the inverse Fourier transform

Idea: Using a forward transform algorithm.

A brief look at the 1D Fourier transform again:

$$\text{Eq. (4.3): } F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j 2\pi ux/M} \quad \text{for } 0 \leq u \leq M - 1 \quad (4.72)$$

$$\text{Eq. (4.4): } f(x) = \sum_{u=0}^{M-1} F(u) e^{j 2\pi ux/M} \quad \text{for } 0 \leq x \leq M - 1 \quad (4.73)$$

Taking the conjugate complex of eq. (4.73) and dividing both sides by M yields:

$$\frac{1}{M} f^*(x) = \frac{1}{M} \sum_{u=0}^{M-1} F^*(u) e^{-j 2\pi ux/M} \quad (4.74)$$

Interpretation: Comparing eq. (4.72) with eq. (4.74):
 → right side of eq. (4.74) describes a forward transform.

Putting $F^*(u)$ into the the algorithm for the forward Fourier transform gives the quantity $f^*(x)/M$.

Taking the conjugate complex and multiplying by M yields $f(x)$.

The extension of the approach to 2D yields:

$$\frac{1}{MN} f^*(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) e^{-j 2\pi (\frac{ux}{M} + \frac{vy}{N})} \quad (4.75)$$

Note: If $f(x)$ or $f(x, y)$ are real functions (e. g. an image) then the conjugate complex of eq. (4.75) is unnecessary: take real part and ignore parasitic imaginary parts.

4.4.3 Function Padding

Reminder: Multiplication in the spatial domain is equivalent to convolution in the frequency domain and vice versa.

Important: Up to now (sec. 4.4) periodicity of Fourier transform has been neglected.
→ Now it can no longer be ignored.

In sec. 4.1.3 the periodicity of a function was discussed.

Now: The significance of periodicity for the convolution process is regarded.

Assumption: 1D convolution

$$f(x) * h(x) = \frac{1}{M} \sum_{m=0}^{M-1} f(m)h(x - m) \quad (4.76)$$

Fig. 4.32 explains the convolution of two aperiodic functions. Fig. 4.33 depicts the same function while taking into account the implied periodicity of the DFT.

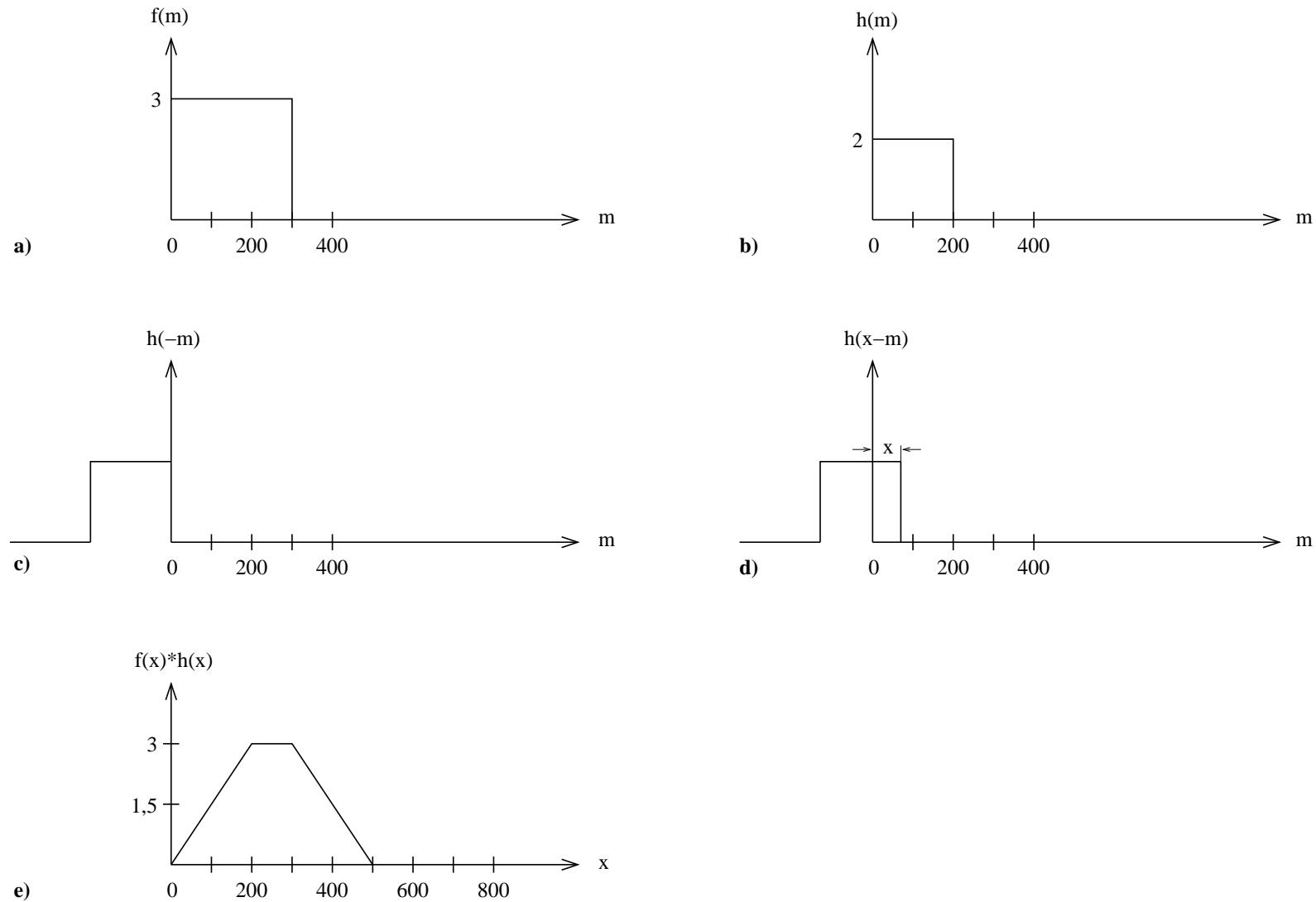


Fig. 4.32: Convolution of two discrete functions. (Adapted from [GW18])

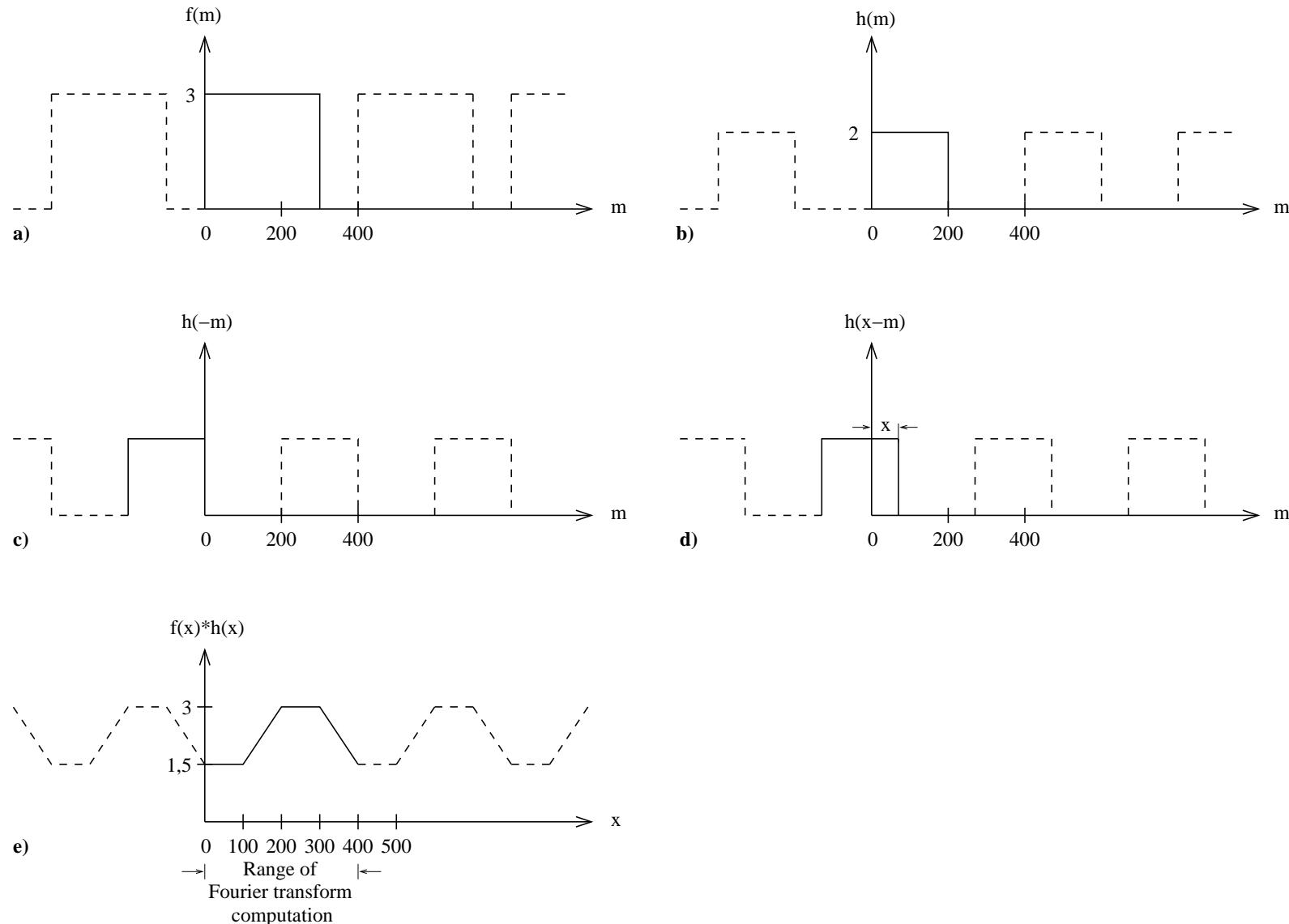


Fig. 4.33: Convolution of the same functions taking into account the implied periodicity of the DFT. Note in (e) how data from adjacent periods corrupt the result of convolution. (Adapted from [GW18])

Interpretation: As $h(x - m)$ slides to the right (see fig. 4.33(d)), the section that was inside $f(m)$ starts to move out the right, but it is replaced by an identical section from the left side of $h(x - m)$.

The convolution gets a constant value in the segment $[0, 100]$ in fig. 4.33(e).

The segment $[100, 400]$ is correct.

In the segment $[400, 500]$, periodicity starts again, thus causing part of the tail of the convolution to be lost.

Solution: Append zeros to both functions so that they have identical periods.

Assumption: f and h consists of A and B points, respectively.

Zeros are added so that both functions have then P points.

The **padded functions** (extended functions):

$$f_e(x) = \begin{cases} f(x) & 0 \leq x \leq A - 1 \\ 0 & A \leq x \leq P \end{cases} \quad (4.77)$$

$$h_e(x) = \begin{cases} h(x) & 0 \leq x \leq B - 1 \\ 0 & B \leq x \leq P \end{cases} \quad (4.78)$$

Note: It can be shown that the individual periods will overlap and cause **wraparound errors** until $P \geq A + B - 1$ is chosen.

The result of extending both functions is depicted in fig. 4.34

Here: The functions $f(x)$ and $h(x)$ in figs. 4.32 and 4.33 are extended to $P = A + B - 1 = 799$.

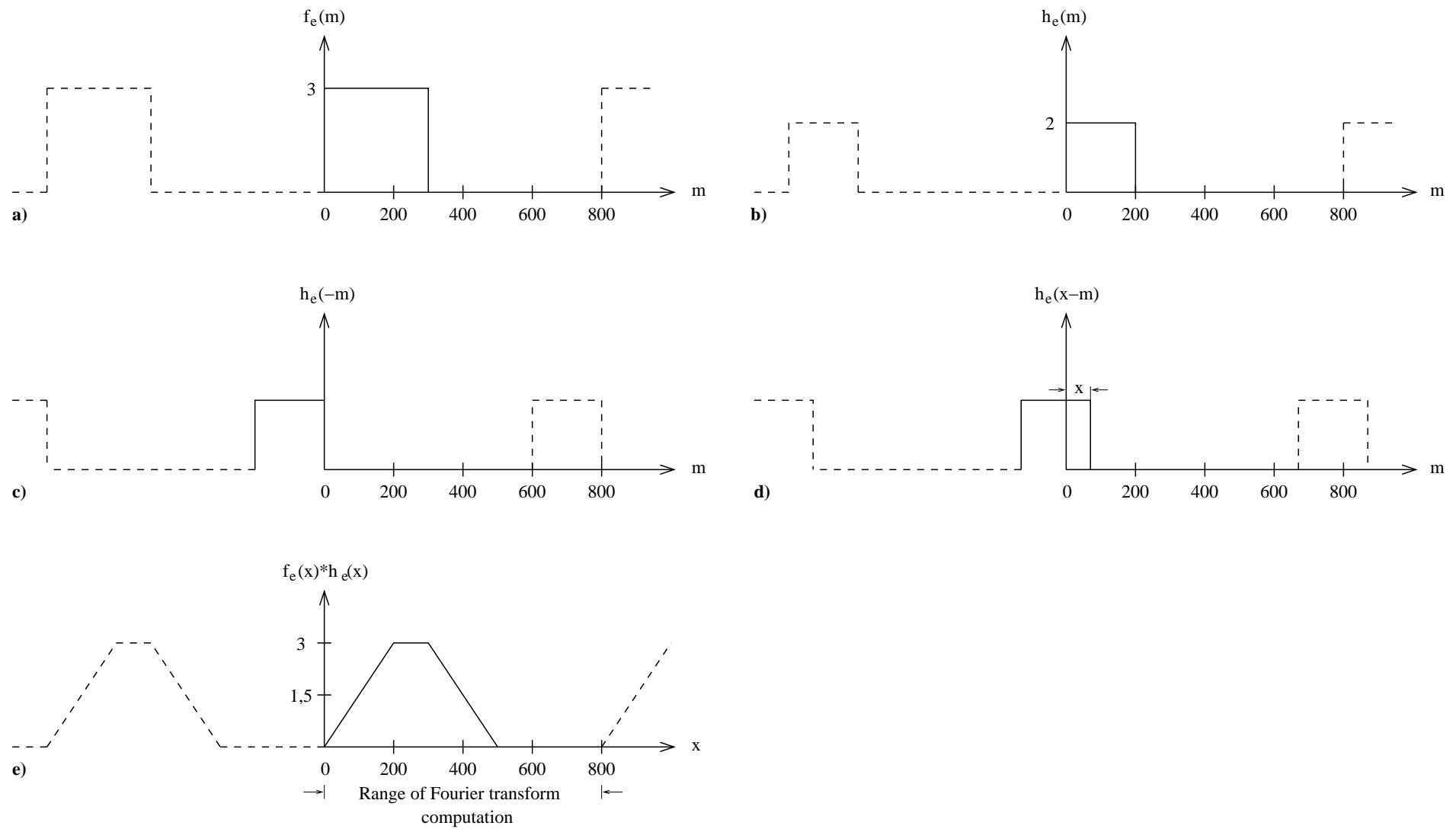


Fig. 4.34: Result of performing convolution with extended functions. Compare figs. 4.34(e) and 4.32(e).
(Adapted from [GW18])

Extensions of these concepts to 2D functions:

Assumption: Two images $f(x, y)$ and $h(x, y)$ of sizes $A \times B$ and $C \times D$, respectively.

Wraparound errors are avoided by choosing

$$P \geq A + C - 1 \quad (4.79)$$

and

$$Q \geq B + D - 1 \quad (4.80)$$

The extended images are formed by extending the original images:

$$f_e(x, y) = \begin{cases} f(x, y) & 0 \leq x \leq A - 1 \text{ and } 0 \leq y \leq B - 1 \\ 0 & A \leq x \leq P \quad \text{or} \quad B \leq y \leq Q \end{cases} \quad (4.81)$$

and

$$h_e(x, y) = \begin{cases} h(x, y) & 0 \leq x \leq C - 1 \text{ and } 0 \leq y \leq D - 1 \\ 0 & C \leq x \leq P \quad \text{or} \quad D \leq y \leq Q \end{cases} \quad (4.82)$$

Important: Unless proper padding is implemented, all filters introduced in the sections 4.2 and 4.3 will deliver erroneous results.

Fig. 4.35 illustrates the effects.

Assumption: $f(x)$ and $h(x)$ are square and both of the same size.

Note: The resulting image in fig. 4.35(c) is twice the size of the original in both directions.

Note: Besides the “padding”, all other aspects of filtering remain as described in section 4.1.

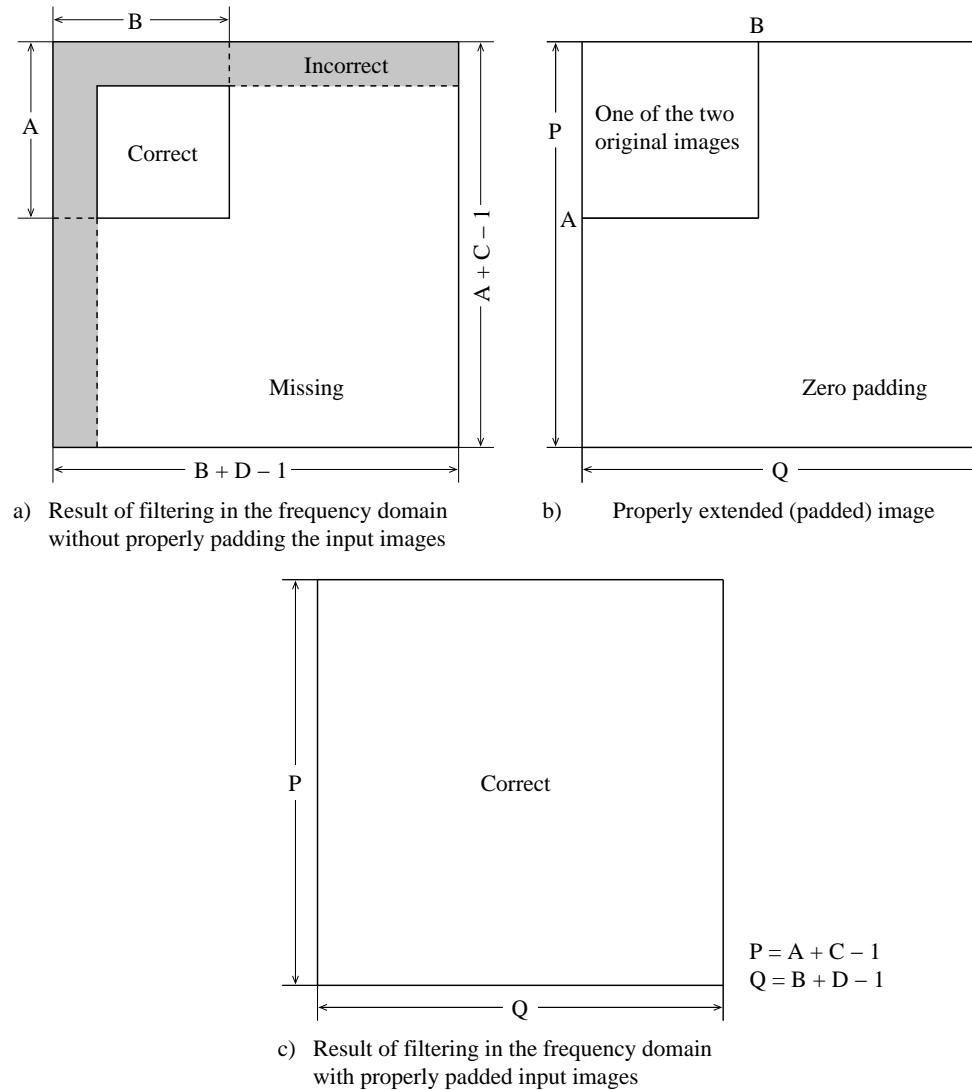


Fig. 4.35: Illustration of the need for function padding. (a) Result of performing 2D convolution without padding. (b) Proper function padding. (c) Correct convolution result. (From [GW18])

Fig 4.36 shows the padded spatial representation of the ideal low pass filter used to generate fig. 4.18(c).

Note: The ILPF was selected for illustration because it has the most “visible” structure in the spatial domain (“ringing”).

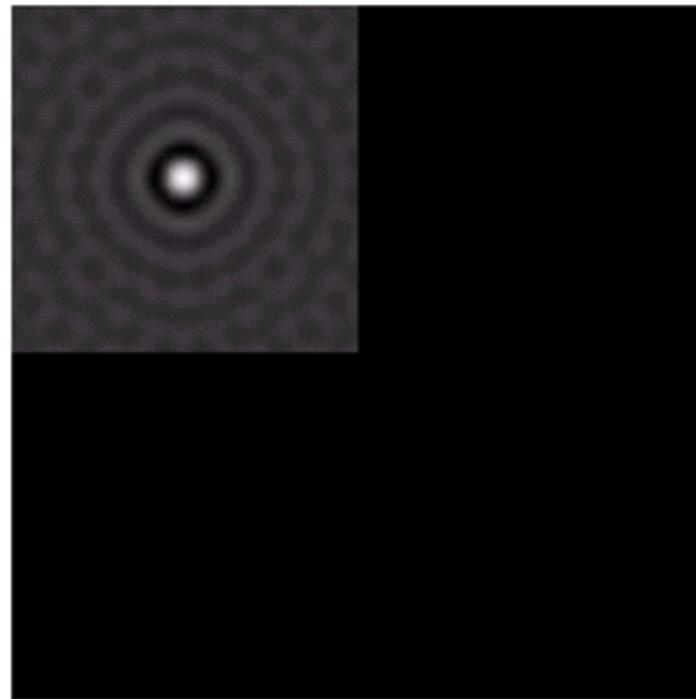


Fig. 4.36: Padded lowpass filter in the spatial domain (only the real part is shown). (From [GW18])

Fig 4.37 shows the result of filtering with padded functions using the depicted approach.

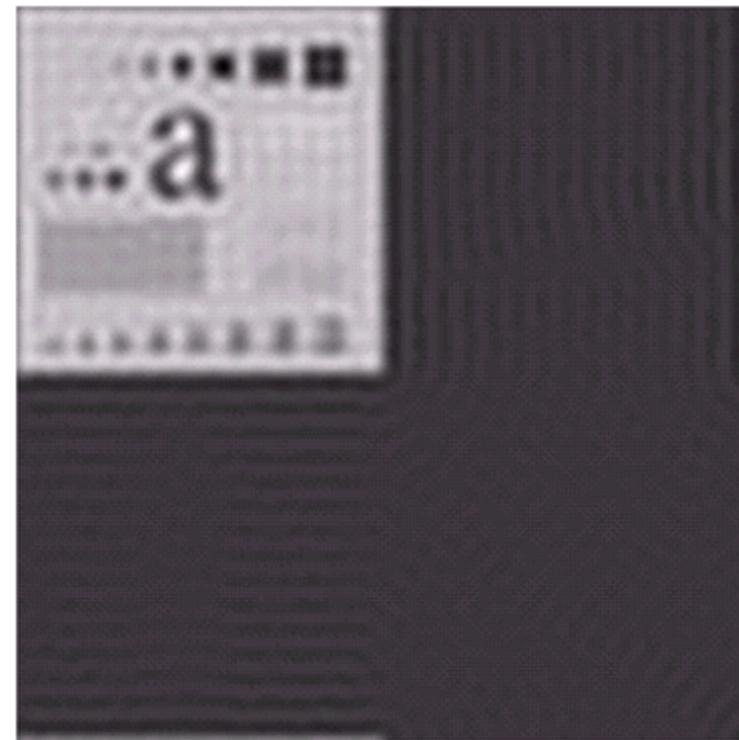


Fig. 4.37: Result of filtering with padding. The image is usually cropped to its original size since there is little valuable information past the image boundaries. (From [GW18])

Interpretation: Three-quarters of the resulting image contain no information:
Cropping back to the original image size delivers the desired filter result.

4.4.4 Convolution and Correlation Theorems

The topic “convolution” was analyzed in detail in section 4.4.2.

The discrete **convolution** is denoted by

$$f(x, y) * h(x, y) = \frac{1}{M N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n) \quad (4.83)$$

The **correlation** of the functions $f(x, y)$ and $h(x, y)$ is defined as

$$f(x, y) \circ h(x, y) = \frac{1}{M N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n) h(x + m, y + n) \quad (4.84)$$

where $f^*(x, y)$ is the conjugate complex form from $f(x, y)$

Hint: The process introduced in eq. (4.84) is often described as **cross correlation** to clarify that the images involved are different.

Note: Images are normally real functions, in which case

$$f^* = f$$

The comparison of eq. (4.83) with (4.84) yields that both equations are equal with two exceptions:

- Conjugate complex image functions and
- Second term in the summation has positive instead of negative signs.

Given the similarity between the two transformations there is a correlation theorem, analogous to the convolution theorem in eqs. (4.3) and (4.4):

$$f(x, y) \circ h(x, y) \quad \circ \bullet \quad F^*(u, v) \cdot H(u, v) \quad (4.85)$$

and

$$f^*(x, y) \cdot h(x, y) \quad \circ \bullet \quad F(u, v) \circ H(u, v) \quad (4.86)$$

Interpretation: Convolution: Tie between filtering in the spatial and frequency domain.
Correlation: Matching between images.

If an image is correlated with itself, the process is called **auto correlation**.

Here, the auto correlation theorem follows from eq. (4.85)

$$f(x, y) \circ f(x, y) \quad \circ \bullet \quad |F(u, v)|^2 \quad (4.87)$$

Note: The Fourier transform of the spatial auto correlation is the **power spectrum** defined in eq. (4.21).

Similarity, it holds

$$|f(x, y)|^2 \quad \circ \bullet \quad F(u, v) \circ F(u, v) \quad (4.88)$$

Fig. 4.38 visualizes the cross correlation algorithm.

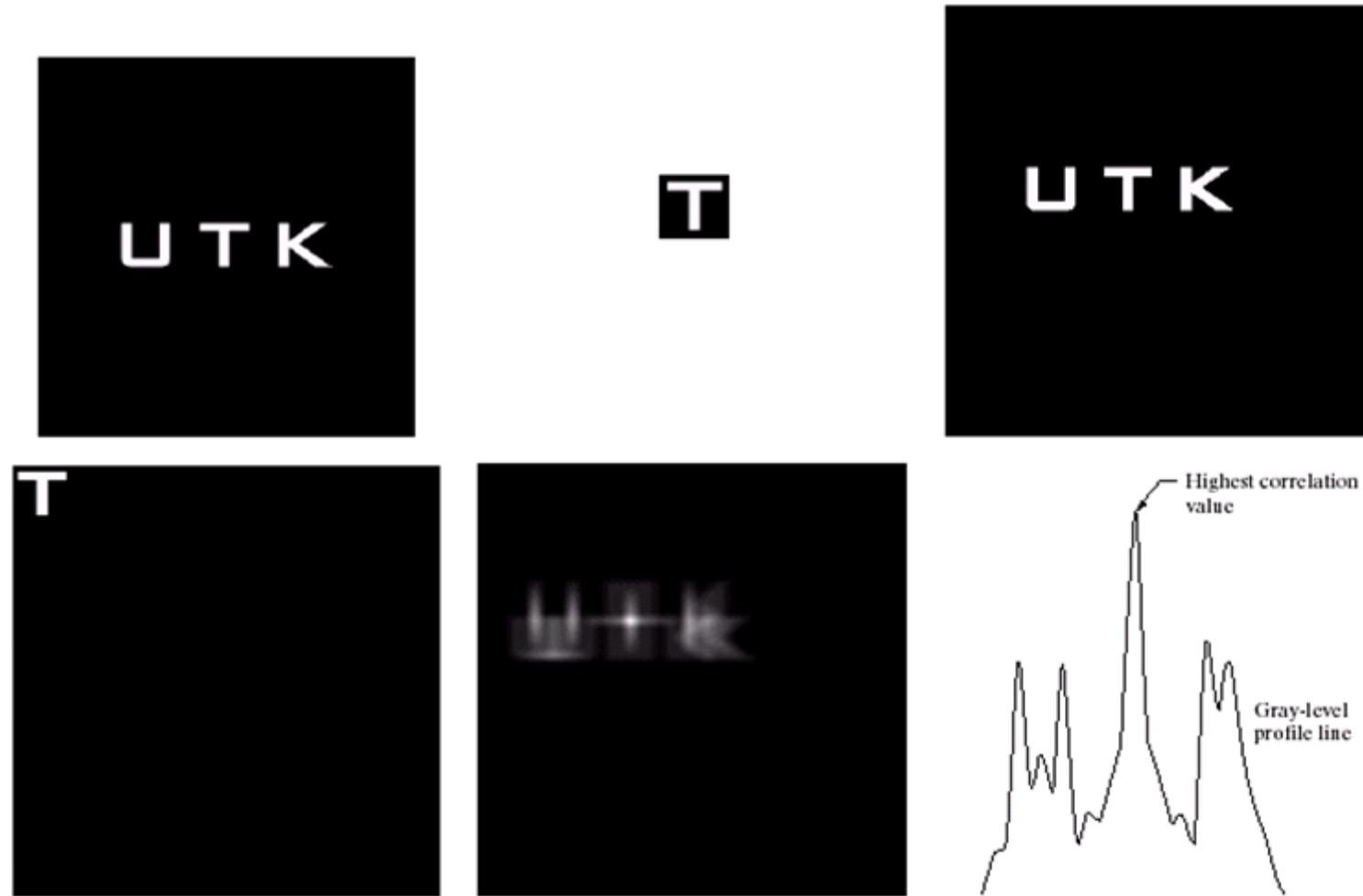


Fig. 4.38: (a) Image. (b) Template. (c)-(d) Padded images. (e) Correlation function displayed as an image. (f) Horizontal profile line through the highest value in (e), showing the point at which the best match took place. (From [GW18])

The template in fig. 4.38(b) is smaller than the image in fig. 4.38(a).

→ Therefore it is padded to the size of the latter (298×298).

Interpretation: The highest value in fig. 4.38(e) occurs at the point where the template is exactly on the top of the "T" in the image.

Note: The variables in the correlation function in the spatial domain are *displacements*.

The left corner of fig. 4.38(e) corresponds to zero displacement.

4.4.5 Fast Fourier Transform

Reconsideration of computing costs:

- 1D Fourier transform in eq. (4.3) requires M^2 multiplications and additions.
If $M = 1024$, the brute force method will required approximately 10^6 operations.

The complexity is $\mathcal{O}(M^2)$.

- A FFT (described below) will need $M \ln M$ operations If $M = 1024$, approximately 10^4 operations.

New complexity: $\mathcal{O}(M \ln M)$.

Interpretation: Computational advantage of 100 to 1.

It gets bigger if M increases.

Now: Introduction of algorithms for a fast Fourier transform.

Assumption:

- 1D Fourier transform.
- A row vector $f(x)$ is interpreted as a vector.

Idea: Divide-And-Conquer Strategy

First: Partitioning the transform of $f(x)$ into a transform of two vectors of length $\frac{M}{2}$ yields half the number of operations:

$$M^2 \rightarrow 2 \left(\frac{M}{2} \right)^2 = \frac{M^2}{2}$$

A number of $\ln M$ successive decompositions are possible, until a row vector of length 1 remains:

From eq. (4.3) with $M = 1$ follows: $F_o = \frac{1}{1} f_0 e^{-j 2\pi 0/1} = f_0$
 → Due to trivial DFT no DFT necessary!

The complexity of the decomposition is $\mathcal{O}(\ln M)$.

Second: After the recursive decomposition the partial results have to be combined.

Here a complexity of $\mathcal{O}(M)$ is possible (see below).

Therefore: The entire complexity can be reduced to $\mathcal{O}(M \ln M)$.

For the notational convenience: eq. (4.3) is written as

$$F(u) = \frac{1}{M} \sum_{x=1}^{M-1} f(x) w_M^{-ux} \quad (4.89)$$

where

$$w_M = e^{\frac{j2\pi}{M}} \quad (4.90)$$

Assumption:

$$M = 2^n = 2K \quad (4.91)$$

with n : positive integer

The vectors $F(u)$ is decomposed in two parts, consisting of the even and odd elements.

The decomposition in even and odd elements is shown in fig. 4.39.

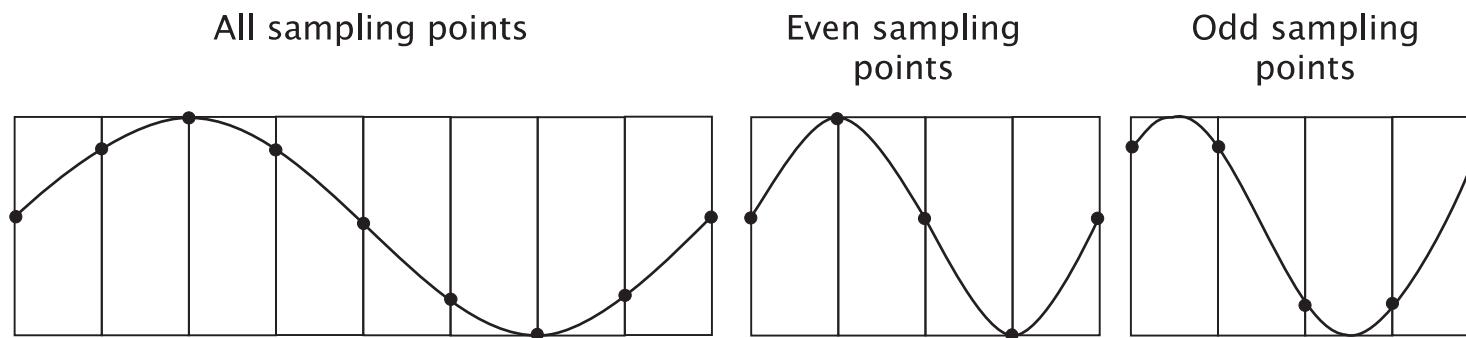


Fig. 4.39: Decomposition of $f(x)$ in even and odd elements. (From [Jäh12])

$$\begin{aligned}
 F(u) &= \frac{1}{2K} \sum_{x=0}^{2K-1} f(x) w_{2K}^{-ux} \\
 &= \frac{1}{2} \left[\frac{1}{K} \sum_{x=0}^{K-1} f(2x) w_{2K}^{-u2x} + \frac{1}{K} \sum_{x=0}^{K-1} f(2x+1) w_{2K}^{-u(2x+1)} \right] \\
 &= \frac{1}{2} \left[\frac{1}{K} \sum_{x=0}^{K-1} f(2x) w_K^{-ux} + w_{2K}^{-u} \frac{1}{K} \sum_{x=0}^{K-1} f(2x+1) w_K^{-ux} \right] \\
 \text{with } F_{\text{even}}(u) &= \frac{1}{K} \sum_{x=0}^{K-1} f(2x) w_K^{-ux} \tag{4.92}
 \end{aligned}$$

$$\text{and } F_{\text{odd}}(u) = \frac{1}{K} \sum_{x=0}^{K-1} f(2x+1) w_K^{-ux} \tag{4.93}$$

$$= \frac{1}{2} [F_{\text{even}}(u) + w_{2K}^{-u} F_{\text{odd}}(u)] \tag{4.94}$$

Interpretation: In eq. (4.94): x runs only from 0 to $K - 1$.

- Interpretation:**
- The aggregation of the partial results consists of a complex multiplication and a complex adding.
→ constant complexity $\mathcal{O}(M)$
 - The second sum in eq. (4.92) contains a phase factor which only depends on the frequency variable u .

It results from the translation theorem of the Fourier transform (see sec. 4.1.3): the odd elements are shifted one position to the left.

Example 4.7 Phase Shift of Odd Samples

The odd samples in fig. 4.39 are shifted by $\frac{\pi}{4}$.

This shift is compensated by the factor w_{2K}^{-u} in eq. (4.92):

$$\begin{aligned} w_{2K}^{-u} &= e^{-\frac{j2\pi u}{8}} \\ &\quad \text{if } u = 1 : \\ &= e^{-\frac{j\pi}{4}} \end{aligned}$$

Using sums in eq. (4.92) resp. eq. (4.94) and the expressions for $F(u)$ yields only $K = \frac{M}{2}$ elements.

→ The double number of elements is required!

Now: Consideration: How to obtain M transform results?

Due to periodicity of Fourier transform:

$$\begin{aligned} F(u) &= \frac{1}{2} [F_{\text{even}}(u) + w_{2K}^{-u} F_{\text{odd}}(u)] \\ F(u + K) &= \frac{1}{2} [F_{\text{even}}(u) - w_{2K}^{-u} F_{\text{odd}}(u)] \end{aligned} \quad (4.95)$$

Factor in second term in eq. (4.95): Originally: Exponent is $(u + K)$:

$$\begin{aligned} w_M^{-(u+K)} &= w_M^{-u} w_M^{-K} \\ &= w_M^{-u} w_M^{-M/2} \\ &= w_M^{-u} e^{-2\pi j M/2M} \\ &= w_M^{-u} e^{-j\pi} \\ &= -w_M^{-u} \end{aligned}$$

Example 4.8 Fast Fourier Transform of a Set of Pixels \mathbf{I}^2

Given: Samples of image row
 (here: Sawtooth with $M = 2$):
 $f(0) = 0; f(1) = 1$

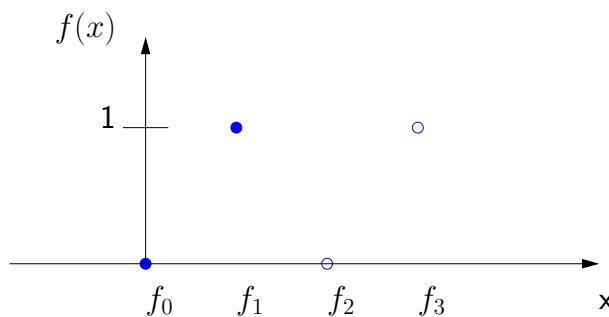


Fig. 4.40: Image samples.

FFT: $K = 1$;
 With eq. (4.92): $F_{even}(0) = 0$
 With eq. (4.93): $F_{odd}(0) = 1$

$$w_2^0 = 1 (!)$$

From eq. (4.92):

$$\begin{aligned} F_0 &= \frac{1}{2} (F_{even}(0) + w_2^0 \cdot F_{odd}(0)) \\ &= \frac{1}{2} \\ F_1 &= \frac{1}{2} (F_{even}(0) - w_2^0 \cdot F_{odd}(0)) \\ &= -\frac{1}{2} \end{aligned}$$

DFT: $w_2 = e^{j\pi} = -1$
 $F(0) = \frac{1}{2}(0 + 1) = \frac{1}{2}$
 $F(1) = \frac{1}{2}(0 + 1 \cdot w_2^{-1}) = -\frac{1}{2}$

² This and the following example are taken from [But11].

Interpretation: Up to now: No big advantage in calculating the FFT instead of the DFT.

*

Example 4.9 Fast Fourier Transform of a Set of Pixels II

Given: Samples of image row

(here: Sawtooth with $M = 4$):

$$f(0) = 0; f(1) = 1; f(2) = 2;$$

$$f(3) = 3$$

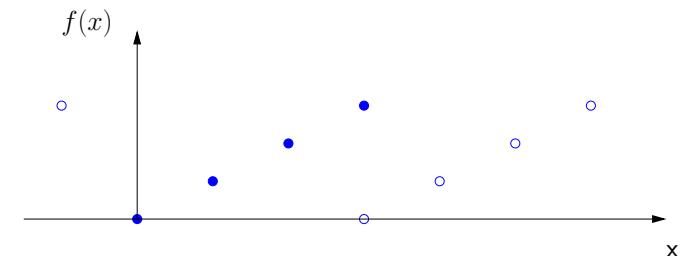


Fig. 4.41: Image samples.

Normal DFT:

$$w_4 = e^{2\pi j/4} = e^{j\pi/2} = j$$

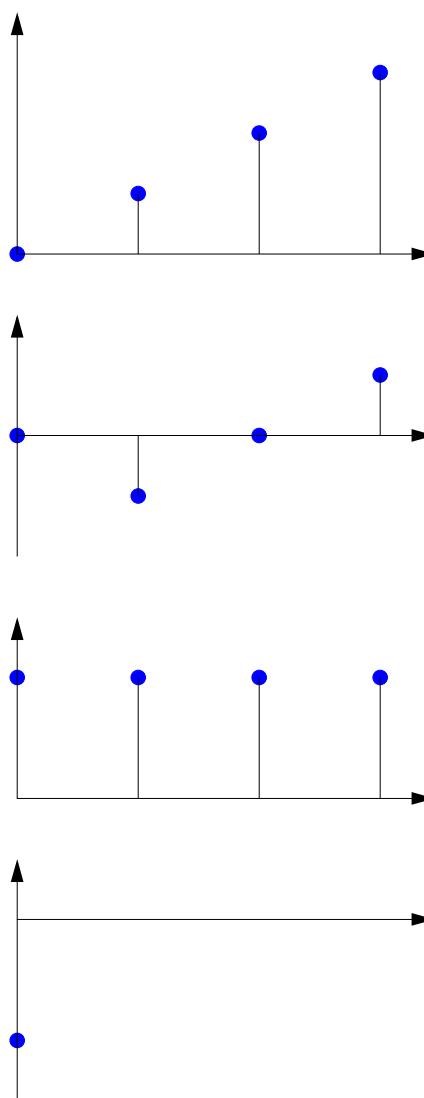
$$F_0 = \frac{1}{4}(0 + 1 + 2 + 3) = \frac{3}{2}$$

$$F_1 = \frac{1}{4}(w_4^{-1} + 2w_4^{-2} + 3w_4^{-3}) = \frac{1}{4}\left(\frac{1}{j} + \frac{2}{-1} + \frac{3}{-j}\right) = -\frac{1}{2} + \frac{j}{2}$$

$$F_2 = \frac{1}{4}(w_4^{-2} + 2w_4^{-4} + 3w_4^{-6}) = \frac{1}{4}(-1 + 2 - 3) = -\frac{1}{2}$$

$$F_3 = \frac{1}{4}(w_4^{-3} + 2w_4^{-6} + 3w_4^{-9}) = \frac{1}{4}\left(-\frac{1}{j} - 2 + \frac{3}{j}\right) = -\frac{1}{2} - \frac{j}{2} \quad (4.96)$$

FFT:



Decomposition of the original signal in three parts:

1. Odd sequence: $F \rightarrow$ imaginary

$$F_1 = \frac{1}{2}$$

$$F_3 = -\frac{1}{2}$$

$$F_0 = F_2 = 0$$

$$F_3 = F(u = 3) = 0$$

2. Constant sequence: $F \rightarrow$ real

$$F_0 = 2$$

$$F_1 = F_2 = F_3 = 0$$

3. δ function: Subtract $\frac{1}{2}$ from each F_u

Fig. 4.42: Decomposition of input sequence.

FFT yields two decompositions:

1. decomposition:

$$M = 4: \quad f_1 = \{0, 2\} \text{ even}$$

$$K = 2: \quad f_2 = \{1, 3\} \text{ odd}$$

2. decomposition: $K' = 1$:

$$f_{1,1} = 0 \quad \text{even} \equiv F_{1,1,0}$$

$$f_{1,2} = 1 \quad \text{odd} \equiv F_{1,2,0}$$

$$f_{2,1} = 2 \quad \text{even} \equiv F_{2,1,0}$$

$$f_{2,2} = 3 \quad \text{odd} \equiv F_{1,2,0}$$

With eq. (4.95) and $u = 0$; $K' = 1$:

$$\begin{aligned} F_{1,x} &= \left\{ \frac{1}{2} F_{1,1,0} + \frac{1}{2} F_{1,2,0} w_2^0, \frac{1}{2} F_{1,1,0} - \frac{1}{2} F_{1,2,0} w_2^0 \right\} = \{1, -1\} \\ F_{2,x} &= \left\{ \frac{1}{2} F_{2,1,0} + \frac{1}{2} F_{2,2,0} w_2^0, \frac{1}{2} F_{2,1,0} - \frac{1}{2} F_{2,2,0} w_2^0 \right\} = \{2, -1\} \end{aligned} \quad (4.97)$$

Applying eq. (4.95) again yields:

$$\begin{aligned} F_0 &= \frac{1}{2} (F_{1,0} + F_{2,0}) = \frac{3}{2} \\ F_1 &= \frac{1}{2} (F_{1,1} + F_{2,1} w_4^{-1}) = \frac{1}{2} (-1 + (-1) \cdot \frac{1}{j}) = -\frac{1}{2} + \frac{j}{2} \\ F_2 &= \frac{1}{2} (F_{1,0} - F_{2,0}) = -\frac{1}{2} \\ F_3 &= \frac{1}{2} (F_{1,1} - F_{2,1} w_4^{-1}) = \frac{1}{2} (-1 - (-1) \cdot \frac{1}{j}) = -\frac{1}{2} - \frac{j}{2} \end{aligned} \tag{4.98}$$

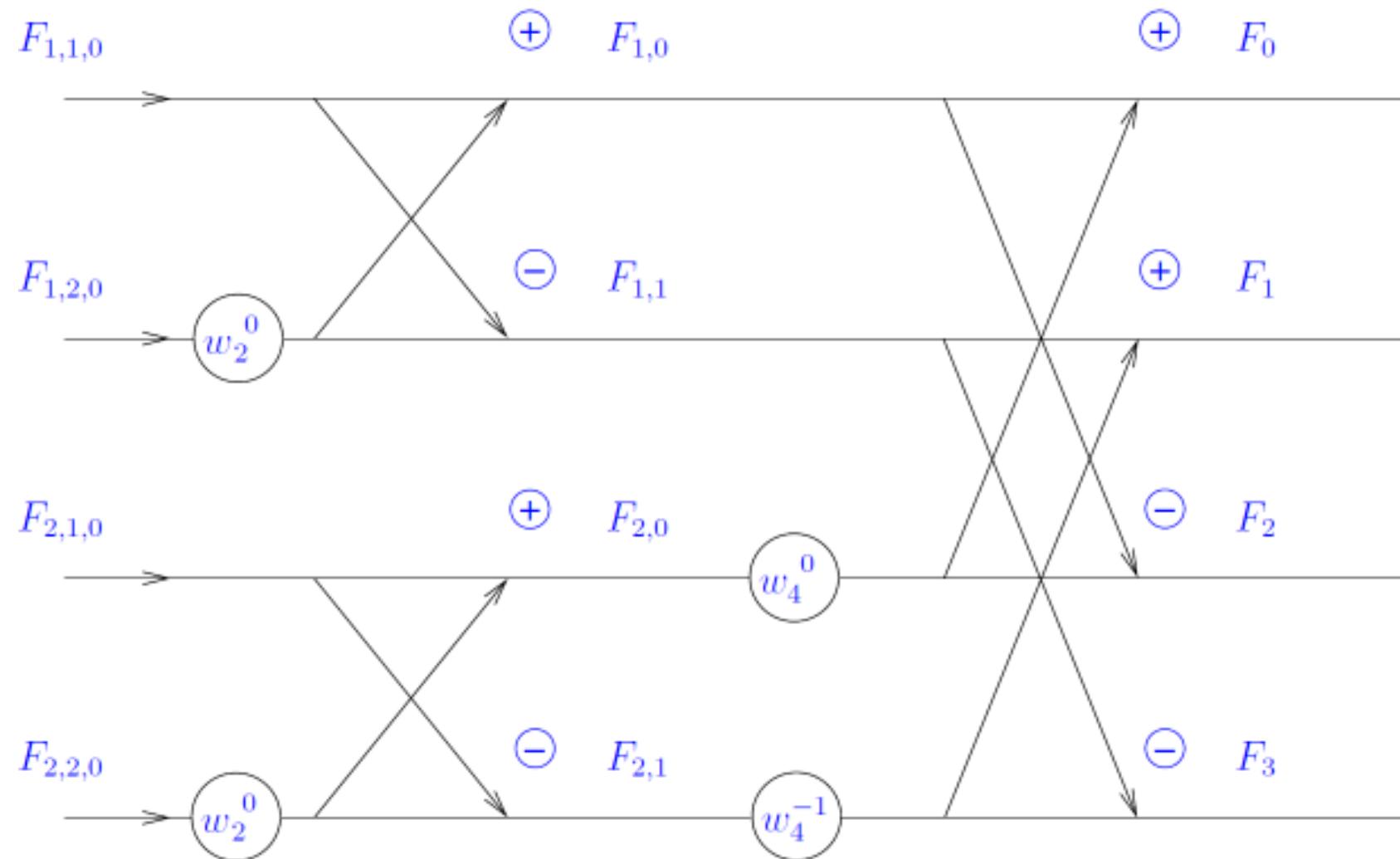


Fig. 4.43: Dataflow graph for FFT with $M = 4$.

The resulting **Fast Fourier Transform (FFT)** algorithm consists of successive bisection of the vector length.

Consideration of the computational implications:

- Assumption:**
- $m(n)$: Number of complex multiplications
 - $a(n)$: Number of complex additions
 - 2^n : Number of samples (as before) where n is a positive integer

$n=1$: Two-point transform requires the evaluation of $F(0)$
 $\rightarrow F_{\text{even}}(0)$ and $F_{\text{odd}}(0)$

$$2^1 = 2K \rightarrow K = 1:$$

- \rightarrow The computation of $F_{\text{even}}(0)$ and $F_{\text{odd}}(0)$ consists of one-point transformations.
- \Rightarrow Fourier transform of a single point is the sample itself: no multiplication or addition required.

One multiplication by w_2^0 and one addition yield $F(0)$ from eq. (4.94).

Now: $F(1)$ follows from the second equation in eq. (4.95) with one more addition³.

As $F_{\text{odd}}(0)w_2^0$ has already been computed, the total number of operations required for a two-point transform consists of:

- $m(1) = 1$ multiplications and
- $a(1) = 2$ additions.

³Subtraction is considered to be the same as addition.

n=2: Four-point transform can be divided into two parts:

- First half: Two two-point transforms as given in eqn. (4.92) and (4.93) for $K = 2$

One two-point transform requires:

- $m(1)$ multiplications and
- $a(1)$ additions

Two two-point transform consists therefore of:

- $2m(1)$ multiplications and
- $2a(1)$ additions

- Two more multiplications and additions are necessary to obtain $F(0)$ and $F(1)$ from eq. (4.94).
- $F_{\text{odd}}(u)w_{2K}^u$ already had been computed for $u = \{0, 1\}$, two more additions give $F(2)$ and $F(3)$.

Total number of operations:

- $m(2) = 2m(1) + 2$, and
- $a(2) = 2a(1) + 4$.

n=3: Two four-point transforms are considered in the evaluation of $F_{\text{even}}(u)$ and $F_{\text{odd}}(u)$.

They require $2m(2)$ multiplications and $2a(2)$ additions.

Four more multiplications and 8 more additions yield the complete transform.

Total number of operations:

- $m(3) = 2m(2) + 4$, and
- $a(3) = 2a(2) + 8$.

Generalization:

n=n: Total number of operations:

$$m(n) = 2m(n-1) + 2^{n-1}, \text{ and} \quad (4.99)$$

$$= \frac{1}{2}M \ln M$$

$$a(3) = 2a(n-1) + 2^n. \quad (4.100)$$

$$= M \ln M$$

where $m(0) = a(0) = 0$ because the transform of a single point requires no operations.

Computational advantage of the FFT over a direct implementation of the 1D DFT:

$$\begin{aligned}\mathcal{C}(M) &= \frac{M^2}{M \ln M} \\ &= \frac{M}{\ln M}\end{aligned}\tag{4.101}$$

If $M = 2^n$

$$\mathcal{C}(M) = \frac{2^n}{n}\tag{4.102}$$

Fig. 4.44 shows a plot of the function:

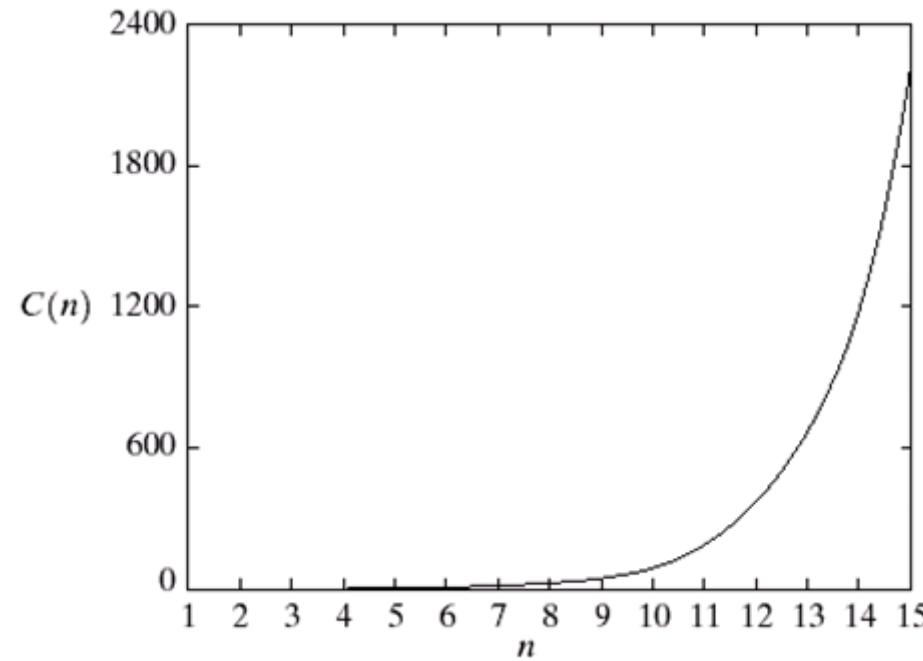


Fig. 4.44: Computational advantage of the 1-D DFT. (From [GW18])

Interpretation: The advantage of the FFT versus the 1D DFT increases rapidly as a function of n .

5 Color Image Processing

Contents

5.1	Color Fundamentals	5-2
5.2	Color Spaces	5-23
5.2.1	RGB color space	5-24
5.2.2	YUV Color Space	5-36
5.2.3	Digital Television and Photography	5-42
5.2.4	CMY and CMYK Color Models	5-45
5.2.5	HSI Color Model	5-51
5.3	Pseudo Color Image Processing	5-67
5.3.1	Intensity Slicing	5-67
5.3.2	Gray Level to Color Transformation	5-74
5.4	Full-Color Image Processing	5-83
5.4.1	Color Transformations	5-86
5.4.2	Color Complements	5-94
5.4.3	Color Slicing	5-97
5.4.4	Tone and Color Corrections	5-102

5.4.5	Histogram Processing	5-111
5.5	Smoothing and Sharpening	5-113
5.5.1	Color Image Smoothing	5-113
5.5.2	Color Image Sharpening	5-119
5.6	Color Segmentation	5-122
5.6.1	Segmentation in HSI Color Space	5-122
5.6.2	Segmentation in RGB Vector Space	5-126
5.6.3	Color Edge Detection	5-132
5.7	Color Image Compression	5-141

Use of color motivated by two factors:

- Color is a powerful descriptor which can be used to simplify image interpretation.
- Humans are very sensitive to colors:
 - They can discern thousands of color shades and intensities, compared to
 - Only two dozen shades of gray.

Two categories of color processing:

- Full-color and
- Pseudo-color.

Hint: • Some of the gray-scale methods covered in previous sections are directly applicable to color images.
• Others require reformulation to be consistent with the properties of the color space introduced in this chapter.

5.1 Color Fundamentals

Note: Color perception by humans is a psychophysical phenomenon:
Besides physically defined units subjective parameters are used.

All statements refer to a “**standardized observer**”.

Experiments by Isaac Newton (1642-1727):

- A beam of white light (sunlight) that passes a glass prism is split into a **continuous spectrum of colors** (see fig. 5.1).
- The single color cannot be divided further.
- All spectral parts combined result in white light again.

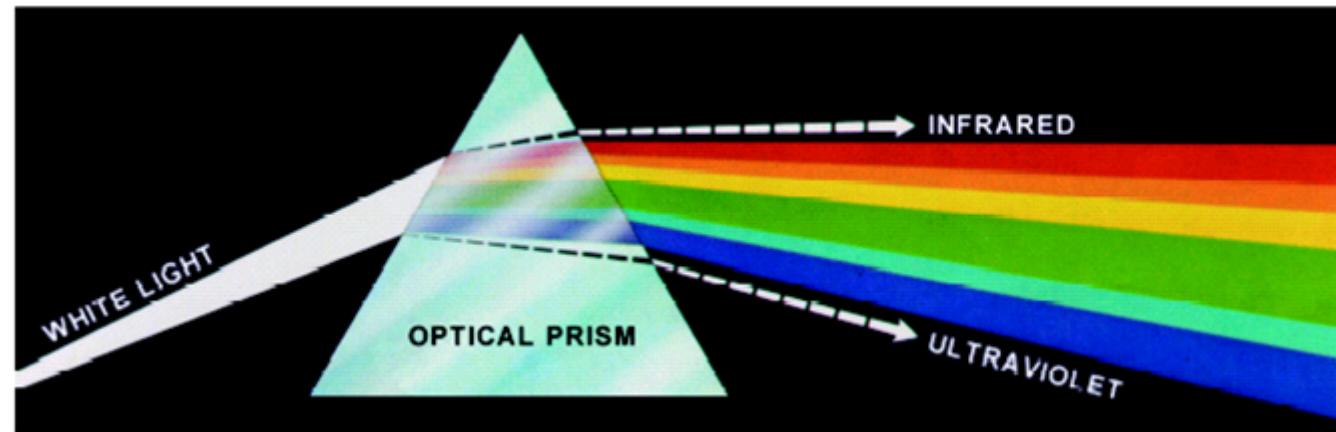


Fig. 5.1: Color spectrum seen by passing white light through a prism. (From [GW18])

Note: All frequencies emitted by radiant objects are part of the spectrum of colors, but color and frequency do not describe the same phenomenon.

- 1) Colors exist that do not occur in the visible spectrum (e. g. brown, violet).

These surfaces are generated by the reflection of light at non-radiant objects.

- 2) The mixture of spectral colors yield the same color perception than a single spectral color.

Newton mixed ‘orange’ from red and yellow spectral colors which could not be distinguished from the orange spectrum color.

Chromatic lights spans the electromagnetic spectrum from approximately 380 nm to 780 nm (see fig. 5.2).

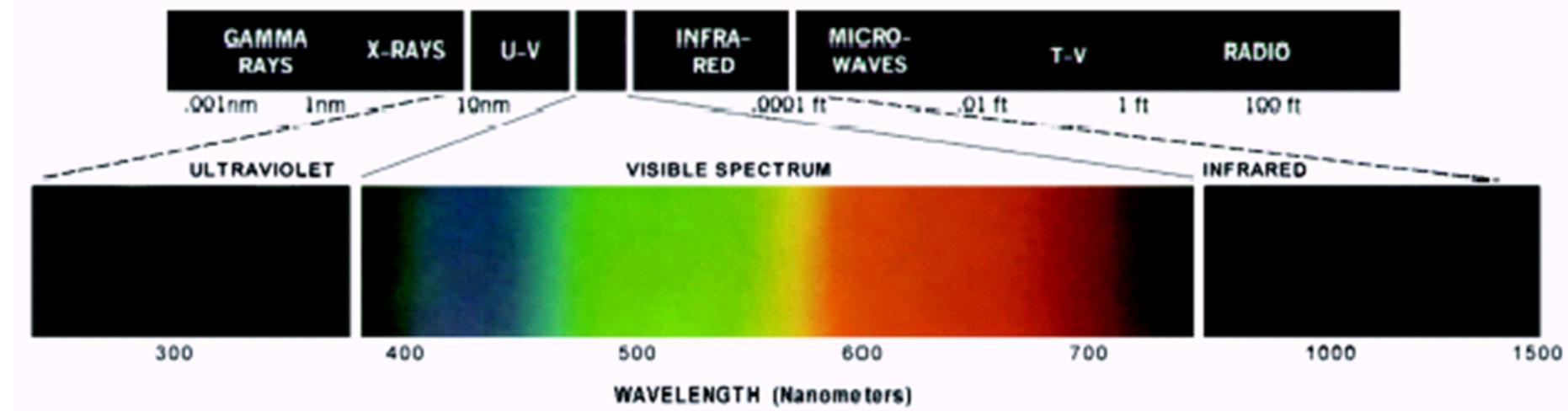


Fig. 5.2: Wavelength comprising the visible range of the electromagnetic spectrum. (From [GW18])

Characterization of color perception:

Radiance:

- Total amount of energy that flows from the light source.
- Measured in watts (W).

Luminance:

- Amount of energy an observer perceives from a light source.
- Measured in lumen (lm).

Brightness:

- Subjective descriptor.
- Almost impossible to measure.

Color metrics: Measurement of color perception.

The human color perception is realized by the approximately 6 to 7 million cones, the color sensors, in the eye.

Three principal sensing categories:

- Roughly corresponding to red (R), green (G), and blue (B).
- Distribution of spectral sensitivities:
 - 65% of all cones: sensitive to red light,
 - 33% of all cones: sensitive to green light, and only
 - 2% sensitive to blue light.

Trichromatic model: Every color sensation can be generated from three **primary colors** (e. g. R, G, and B).

Fig. 5.3 shows average experimental curves, detailing the absorption of light by the red, green, and blue cones in the eye.

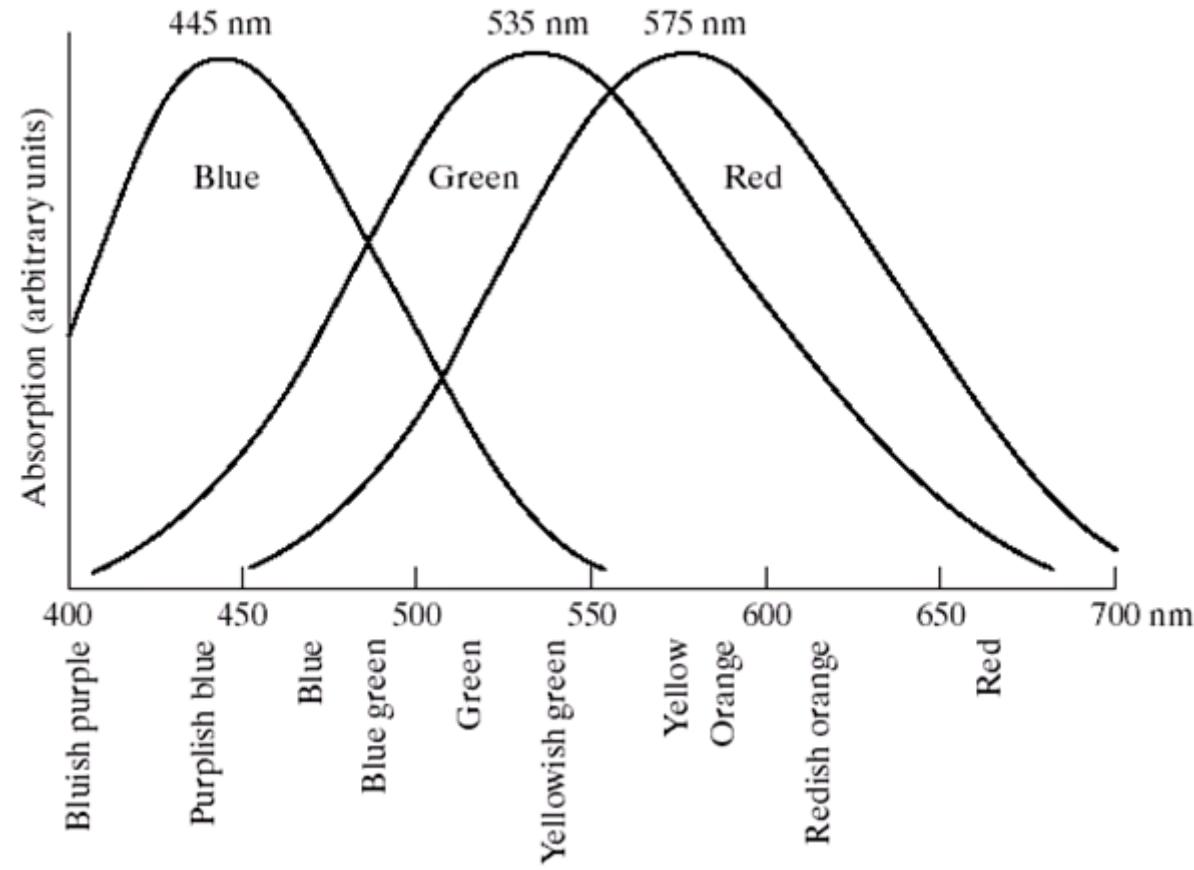


Fig. 5.3: Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.
(From [GW18])

Designation of the following specific wave length values to the three primary colors¹.

- Blue: 435,8 nm,
- Green: 546,1 nm, and
- Red: 700,0 nm.

Note: The standard was set *before* the detailed experimental curves shown in fig. 5.3 become available in 1965.

Note: Do not misinterpret the term *primary*:
The three standard primaries cannot be used to generate *all* visible colors.

¹Standardization by CIE (Commission International de l'Eclairage or International Commission on Illumination), 1931

Mixture of colors:

- **Additive:**
 - The primary color R , G , and B can be added to produce the **secondary colors** “cyan (C)”, “magenta (M)”, and “yellow (Y)”.
 - Mixing the three primary, or a secondary with its opposite primary color, in the right intensities produces white light².
 - The mixture of lights is shown in fig. 5.4.
- **Subtractive:**
 - Mixing the the secondaries from above produces the primaries or “black” (see fig. 5.5.)
 - The secondaries can be created by directing a beam of white light through color filters in the secondary colors.

² There are two nice web applications describing color mixing: [http://www.cs.brown.edu/exploratories/freeSoftware/repository/edu/brown/cs/exploratories/applets/colorMixing/additive_color_mixing_guide.html] and [http://www.cs.brown.edu/exploratories/freeSoftware/repository/edu/brown/cs/exploratories/applets/colorMixing/subtractive_color_mixing_guide.html]

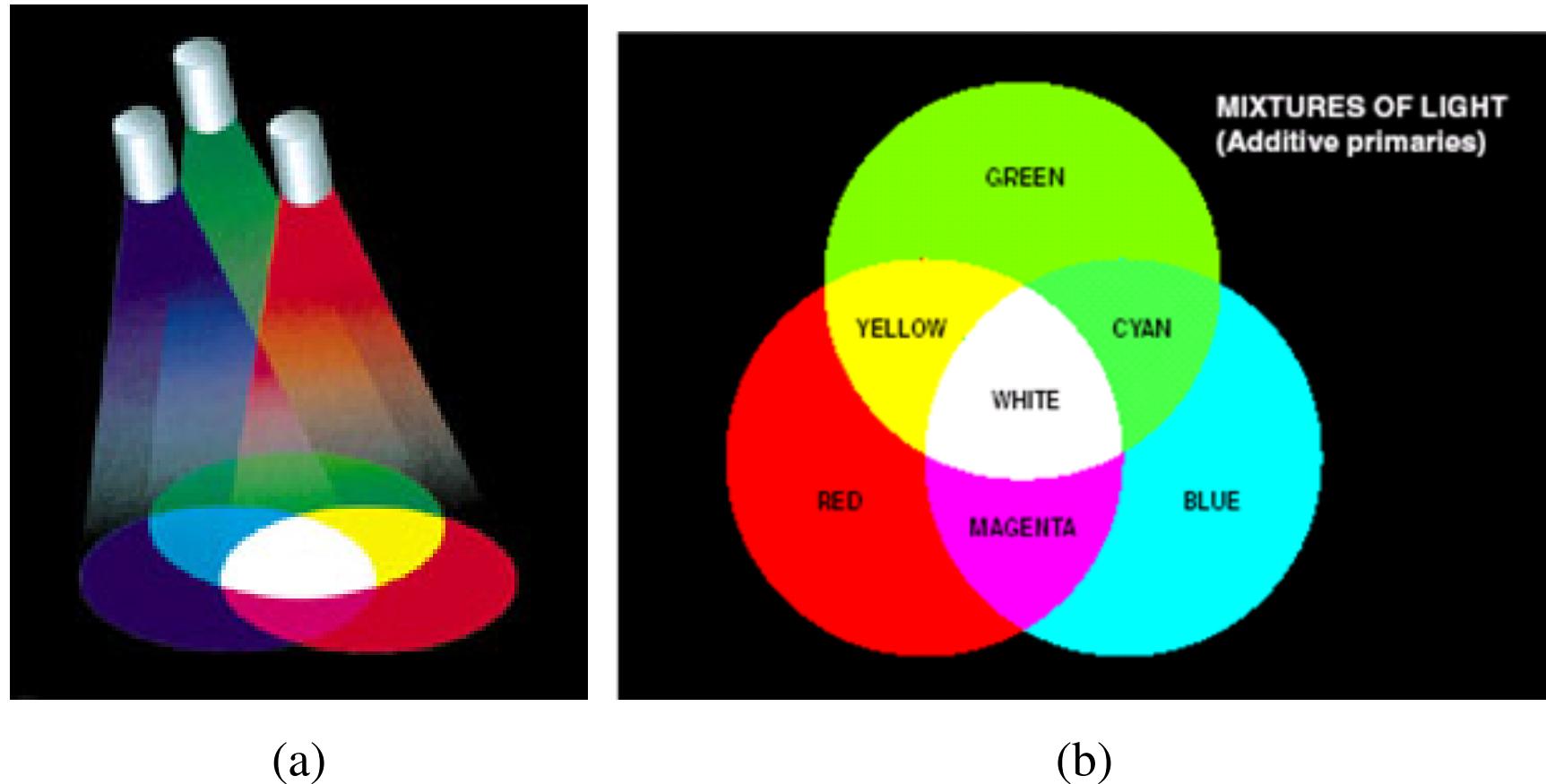
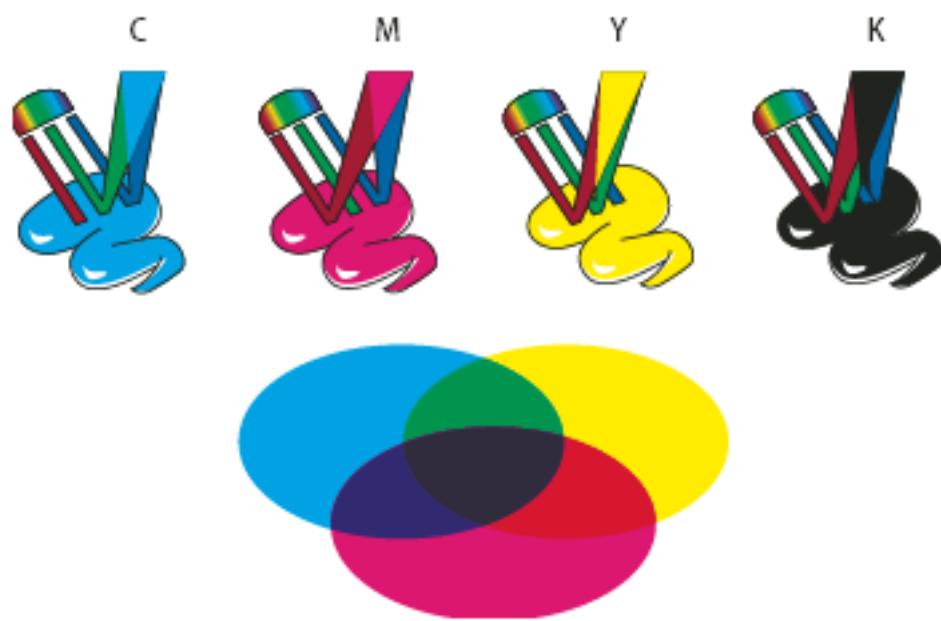
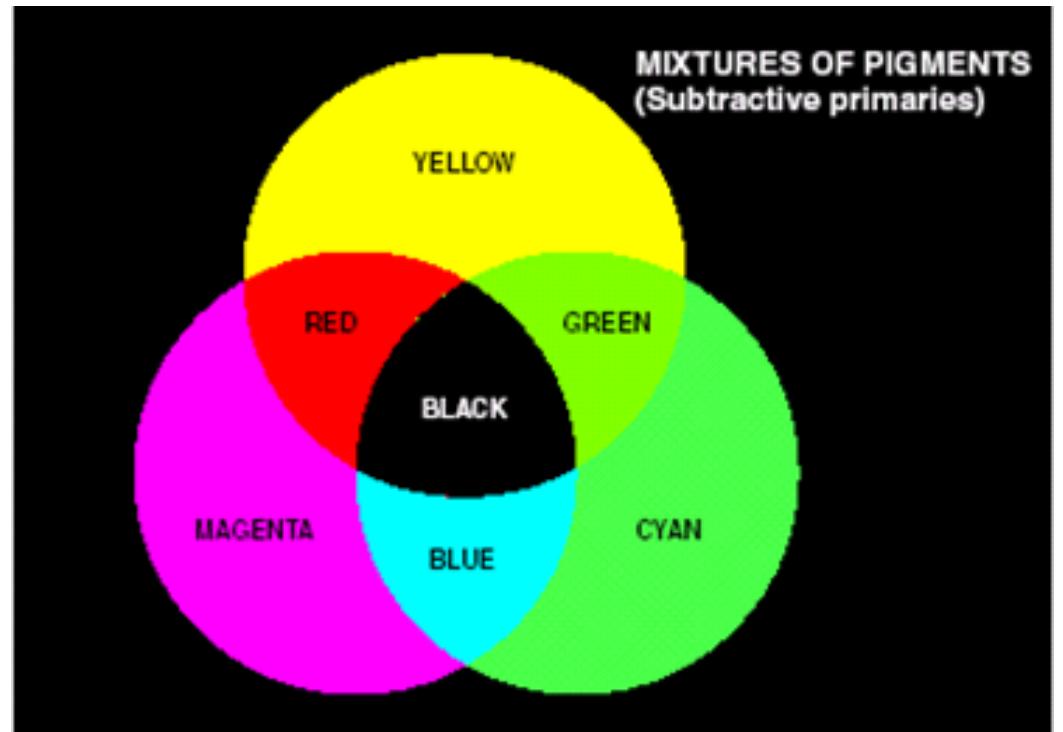


Fig. 5.4: Additive color mixing. (From [GW18])



(a)



(b)

Fig. 5.5: Subtractive color mixing. (From [GW18])

Note: It is necessary to differentiate between different primary colors for color mixture:

- **Additive:** Primary colors of light.
- **Subtractive:** Primary colors of pigments or colorants. The primary color is here defined as one that subtracts or absorbs a primary color of light and reflects or transmits the two others.

Applications of color mixtures:

- **Additive:** Color television
- **Subtractive:** Color photography, color printing

Discrimination of colors by using the following characteristics:

- **Hue:** Attribute associated with the dominant wavelength in a mixture of wavelengths (e. g. reddish vs. bluish green)
- **Saturation:** Relative purity or amount of white light mixed with a hue (“pastelity”)
- **Brightness:** Chromatic motion of intensity (e. g. light yellow compared to dark yellows)

- Interpretation:**
- Pure spectrum colors are fully saturated.
 - Colors such as pink (red and white) are less saturated, with the degree of saturation being inversely proportional to the amount of white light added.

Hue and saturation together are called **chromaticity**.

→ Therefore a color may be characterized by its brightness and chromaticity.

In a trichromatic model, the amounts of the primary colors, e. g. red, green, and blue, needed to form any particular color are called **tristimulus values** and are denoted, X , Y , and Z , respectively.

A color is specified by its **trichromatic** coefficients defined as

$$\begin{aligned} x &= \frac{X}{X + Y + Z} \\ y &= \frac{Y}{X + Y + Z} \quad \text{and} \\ z &= \frac{Z}{X + Y + Z} \end{aligned} \tag{5.1}$$

It is noted from these equations that

$$x + y + z = 1 \quad (5.2)$$

Note: A 3D color model is not suitable for color metric purposes due to the shift of layers of constant brightness (see fig. 5.6).

→ Reduction to two dimensions by normalizing to the sum of tristimulus values.

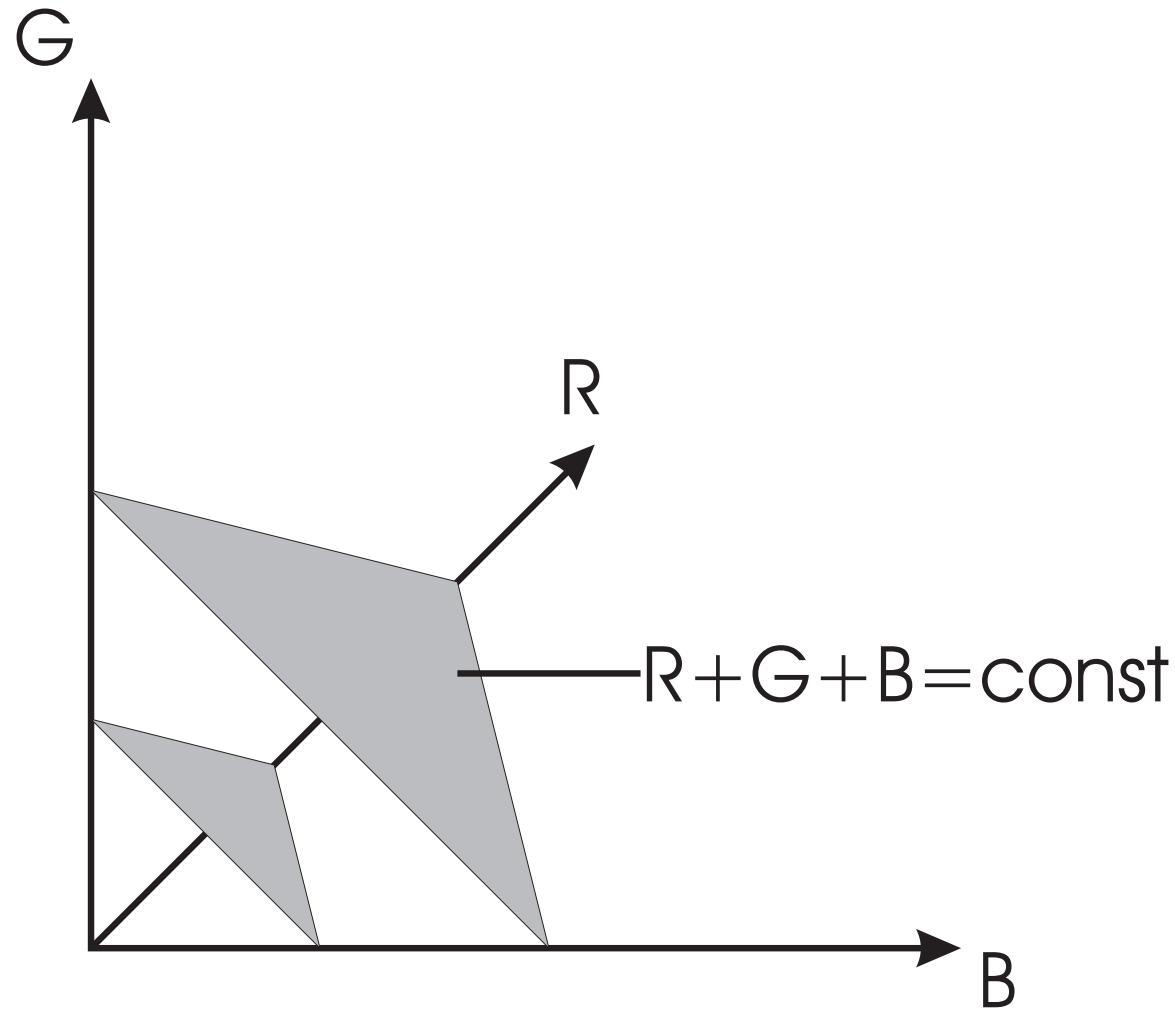


Fig. 5.6: Motivation of the chromaticity diagram

The **CIE chromaticity diagram** (see fig. 5.7) shows color composition as a function of x (red) and y (green).

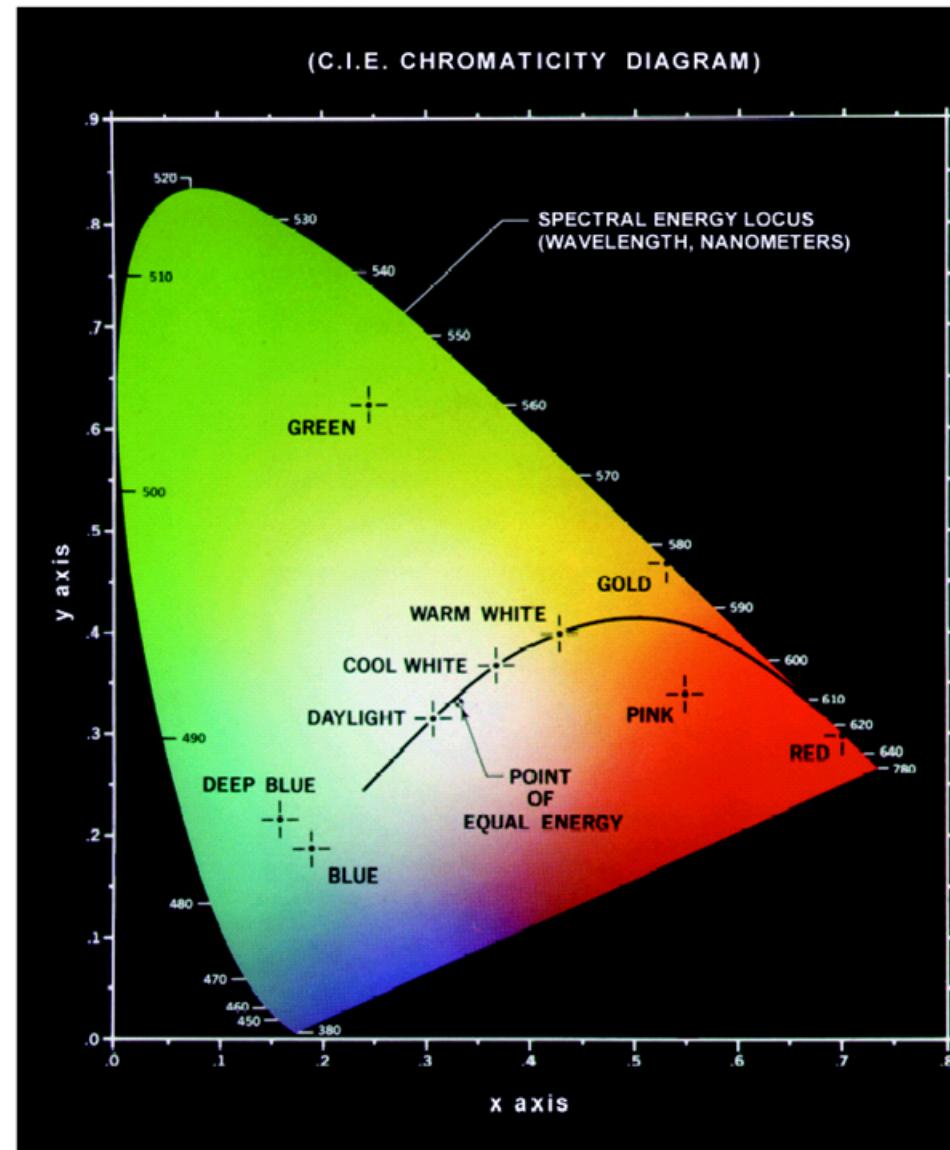


Fig. 5.7: Chromaticity diagram. (From [GW18])

The corresponding value of z (blue) can be derived from eq. (5.2).

- Interpretation:**
- The positions of the various spectrum colors are indicated around the boundary of the tongue-shaped diagram. (“Pure” colors from fig. 5.2)
 - The point of equal energy corresponds to equal fractions of three primary colors ($x = y = 0.33$) :
It represents CIE standard for white light.
 - Any point at the boundary is fully saturated. As a point leaves the boundary and approaches the point of equal energy, more white light is added to the color and it becomes less saturated.
The saturation at the “illuminant” is zero.

- Important:** A straight-line segment joining any two points in the diagram defined all the different color variations that can be obtained by combining these two colors additively.
Extension to three points: The resulting triangle contains all possible variations of the three initial colors (see fig. 5.9).

The gamut for any RGB color system is a triangle in the CIE chromaticity diagram (see fig. 5.8).

Fig. 5.9 describes the color gamut additive (triangle) and subtractive ('cloud') color mixing in the CIE chromaticity diagram.

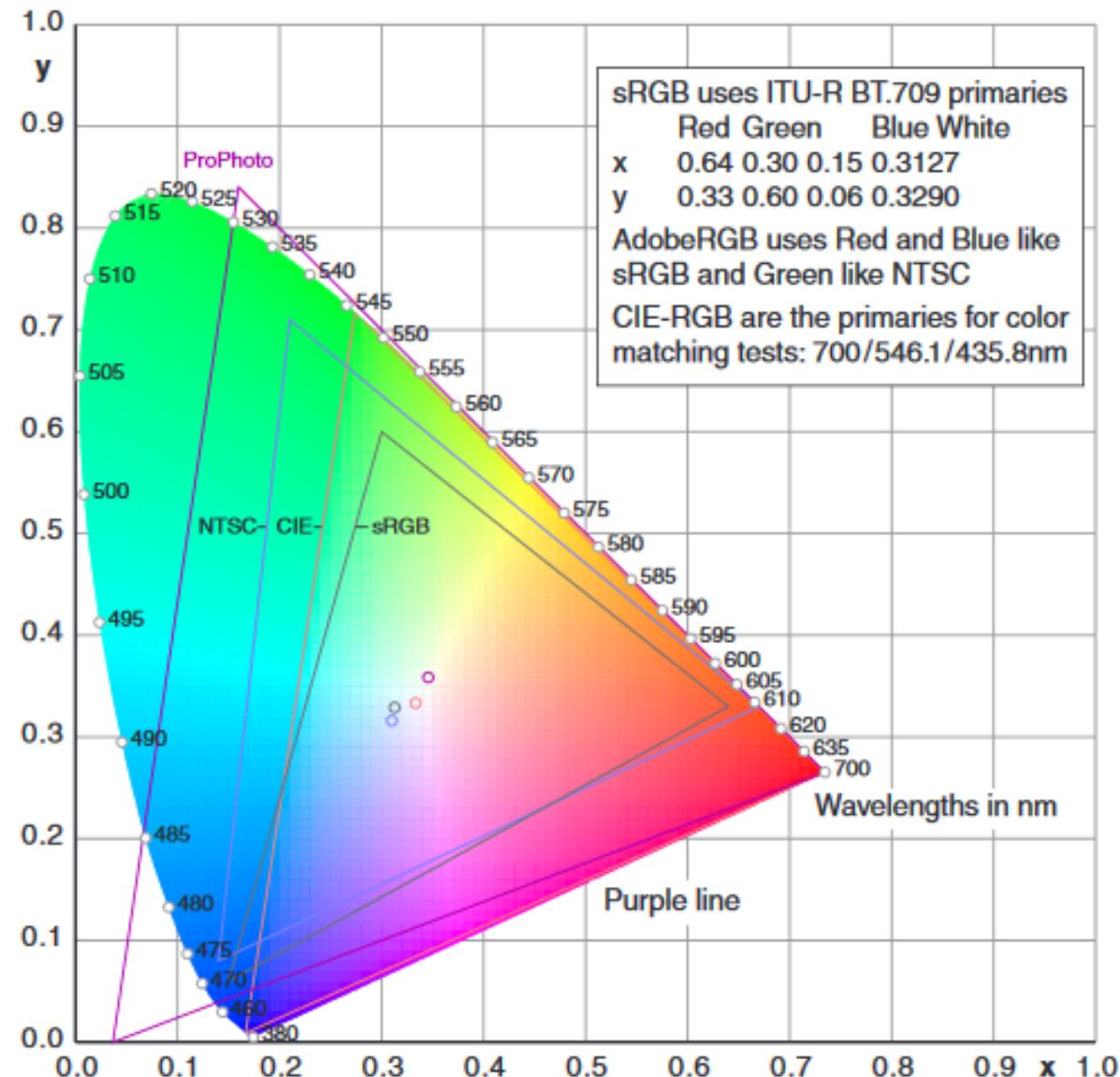


Fig. 5.8: Gamuts for different RGB color spaces. (From [Hof])

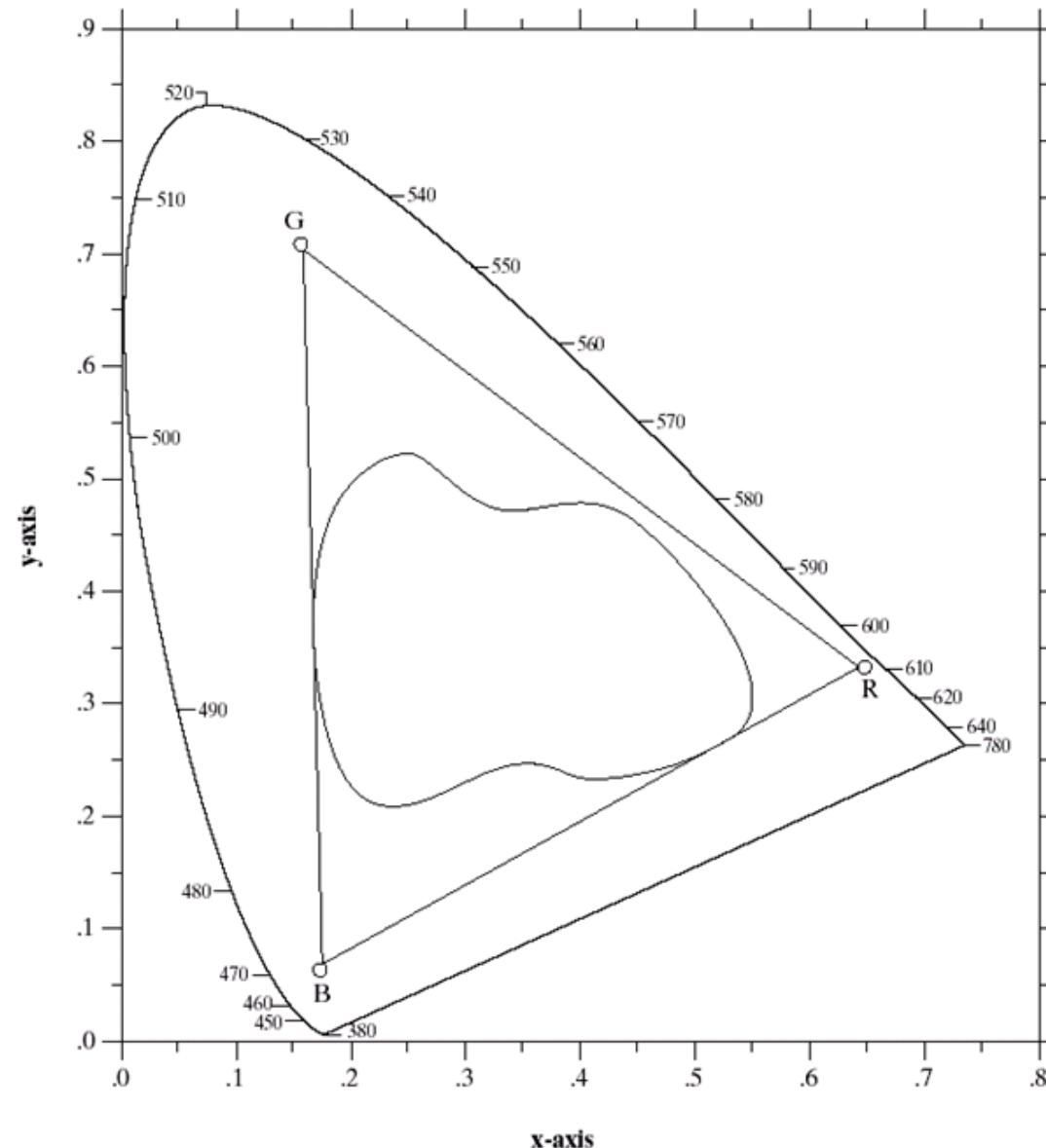


Fig. 5.9: Typical color gamut of color monitors (triangle) and color printing devices (irregular region). (From [GW18])

5.2 Color Spaces

A **color space** (or color model or color system) is a standardized specification of colors.

It consists of a coordinate system and a subspace within where each color is represented by a single point.

The choice of a color space depends on the hardware (such as color monitors or printers) or the application.

In this section, the following color spaces will be introduced:

RGB: Red, green, blue model for color monitors and a broad class of color video cameras.

YUV: Used in the PAL system of television broadcasting.

CMY/CMYK: Cyan, magenta, yellow or cyan, magenta, yellow, black model for color printing.

HSI: The model decouples color and gray-scale information.

Many more specialized color spaces exist in the literature.

5.2.1 RGB color space

Each color is described by its primary spectral components of red, green, and blue.

The color space is based on the cartesian coordinate system.

The color subspace of interest is a cube (see fig. 5.10).

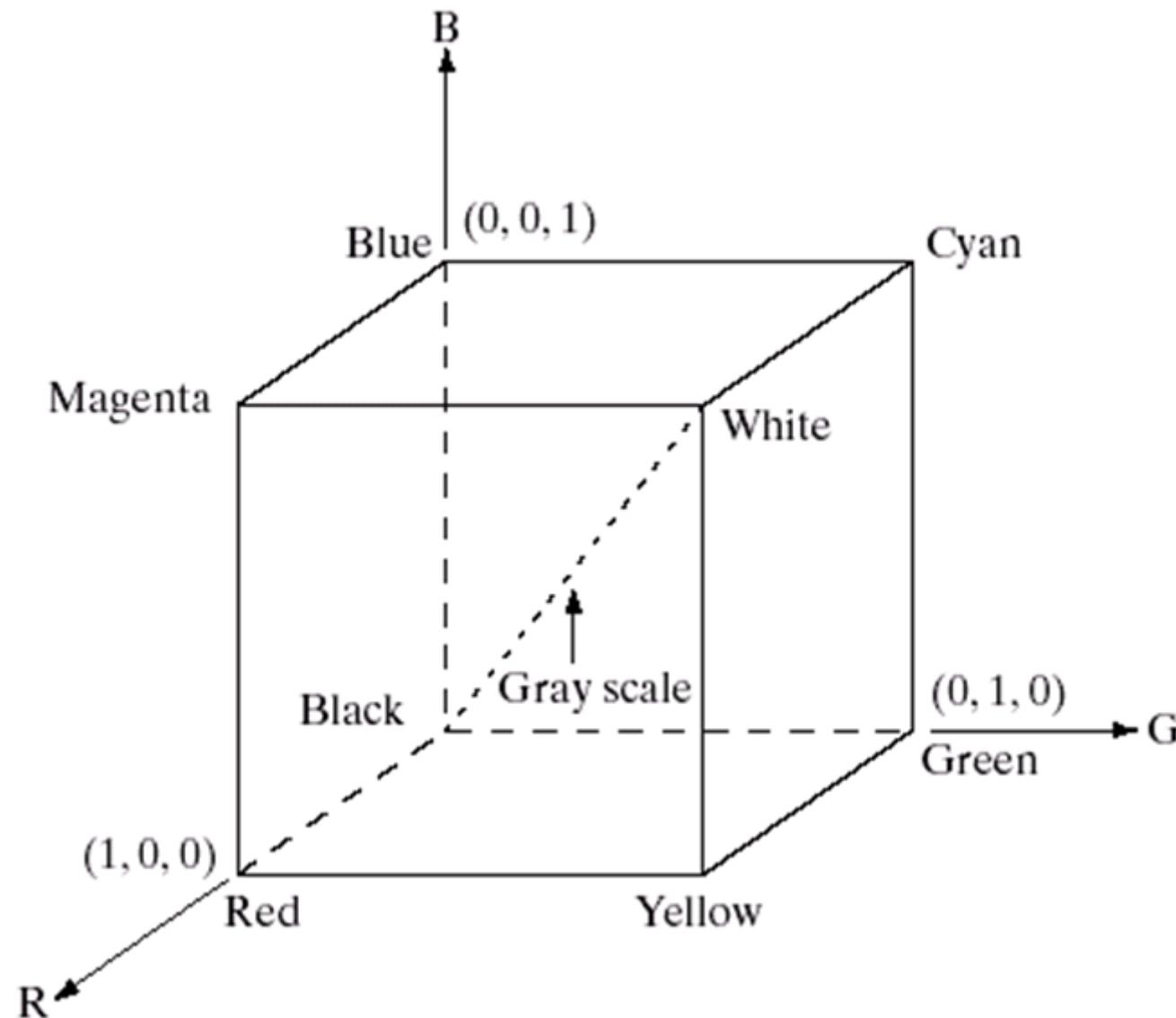


Fig. 5.10: Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point $(1,1,1)$. (From [GW18])

Note: For convenience, all values have been normalized so that the cube shown is the unit cube.

The values of R, G, and B are in the range [0, 1].

An image represented in the RGB color model consists of three component images, one for each primary color.

When fed into a RGB monitor, these three images combine on the phosphor screen to produce a composite color image:

Pixel depth: Number of bits to represent each pixel.

Example 5.1 Full-Color Image

The term is used often to denote a 24-bit RGB color image.

Each Color (that is a triplet of values (R, G, B)) has a depth of 24 bits (3 image planes times the number of bits per plane).

Total number of colors: $(2^8)^3 = 16.777.216$.

Fig. 5.11 shows the corresponding (solid) 24-bit RGB color cube.

*

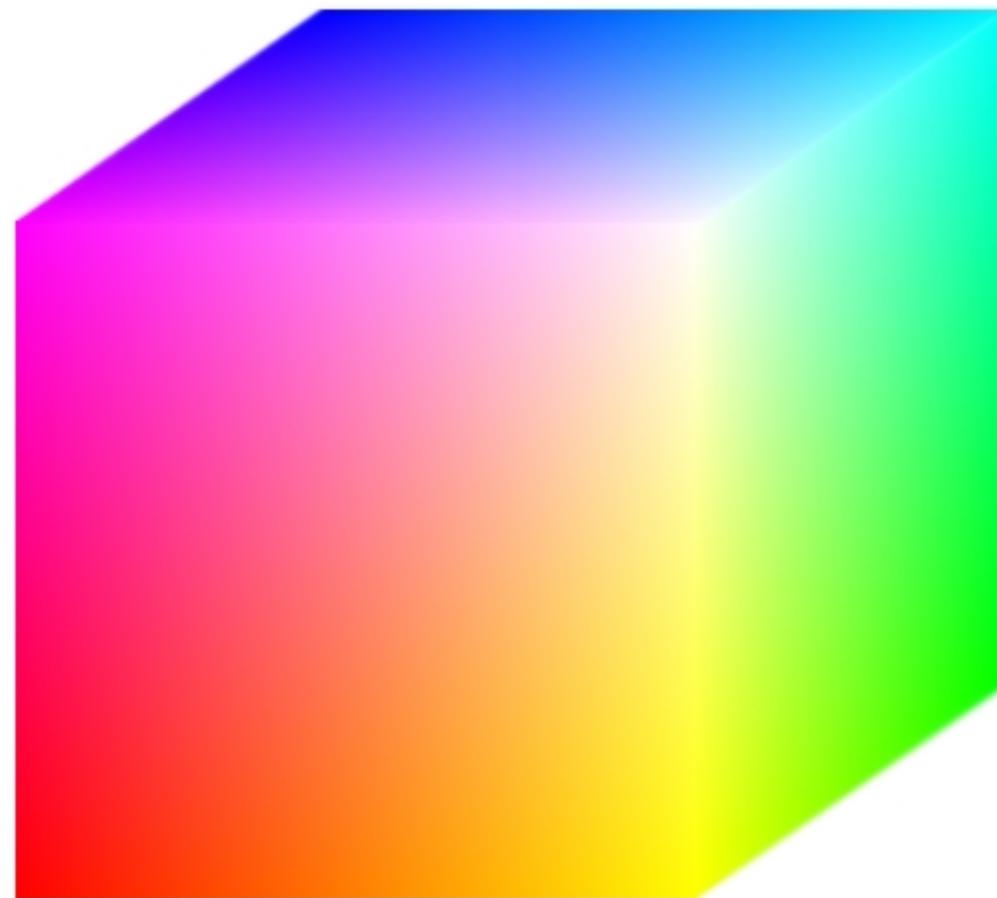


Fig. 5.11: RGB 24-bit color cube. (From [GW18])

These colors can be made visible by generating color planes.

Example 5.2 Cross-sectional plane through the center of the cube and parallel to the GB-plane

Assumption:

- $R = 127$.
- G, B are in the range $[0, 255]$.

Note: Here the actual pixel values are used (as a computer would do) rather than the mathematically normalized values in the range $[0,1]$.

In fig. 5.12 (a) the result is shown by feeding the individual components into a color monitor, while fig. 5.12 (b) shows the three hidden surfaces of the cube in fig. 5.11.

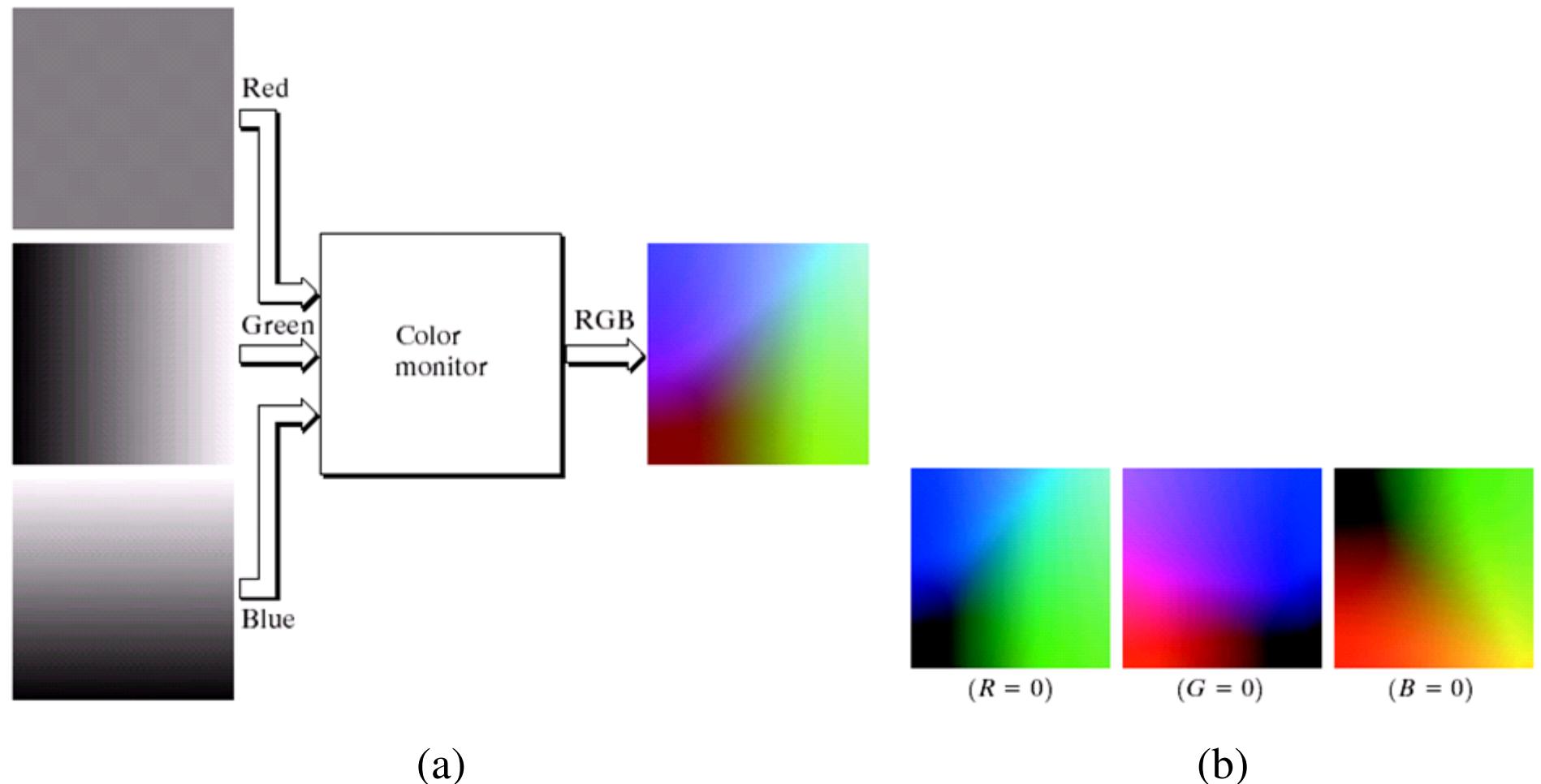


Fig. 5.12: (a) Generating the RGB image of the cross-sectional color plane (127,G,B). (b) The three hidden surface planes in the color cube of fig. 5.11. (From [GW18])

While high-end display cards and monitors provide a reasonable rendition of the colors in a 24 bit color image, some systems still are restricted to 256 colors.

Some applications do not need more than 256 color states.

Using a pseudo-color processing (see section 5.3) a mapping from the color depth of 24 bits to 8 bits is suitable.

The resulting subset is called set of **safe RGB colors** or **all-systems-safe colors**.

In Internet applications, the terms **safe web colors** or **safe browser colors** are used.

Note: Forty of the 256 colors are known to be processed differently by various operating systems.

Only 216 colors are left that are common to most systems.

→ De facto standard for safe colors.

As before, the 216 safe colors are formed by RGB triplets, but each value can be 0, 51, 102, 153, 204, or 255 ($6^3 = 216$).

It is customary to express these values in the hexagonal number system (see table 5.1).

Number System Color Equivalents	
Hex	00 33 66 99 CC FF
Decimal	0 51 102 153 204 255

Table 5.1: Valid values of each RGB component in a safe color. (From [GW18])

Fig. 5.13(a) shows the 216 safe colors, organized by descending RGB values, while fig. 5.14 shows the RGB safe-color cube.

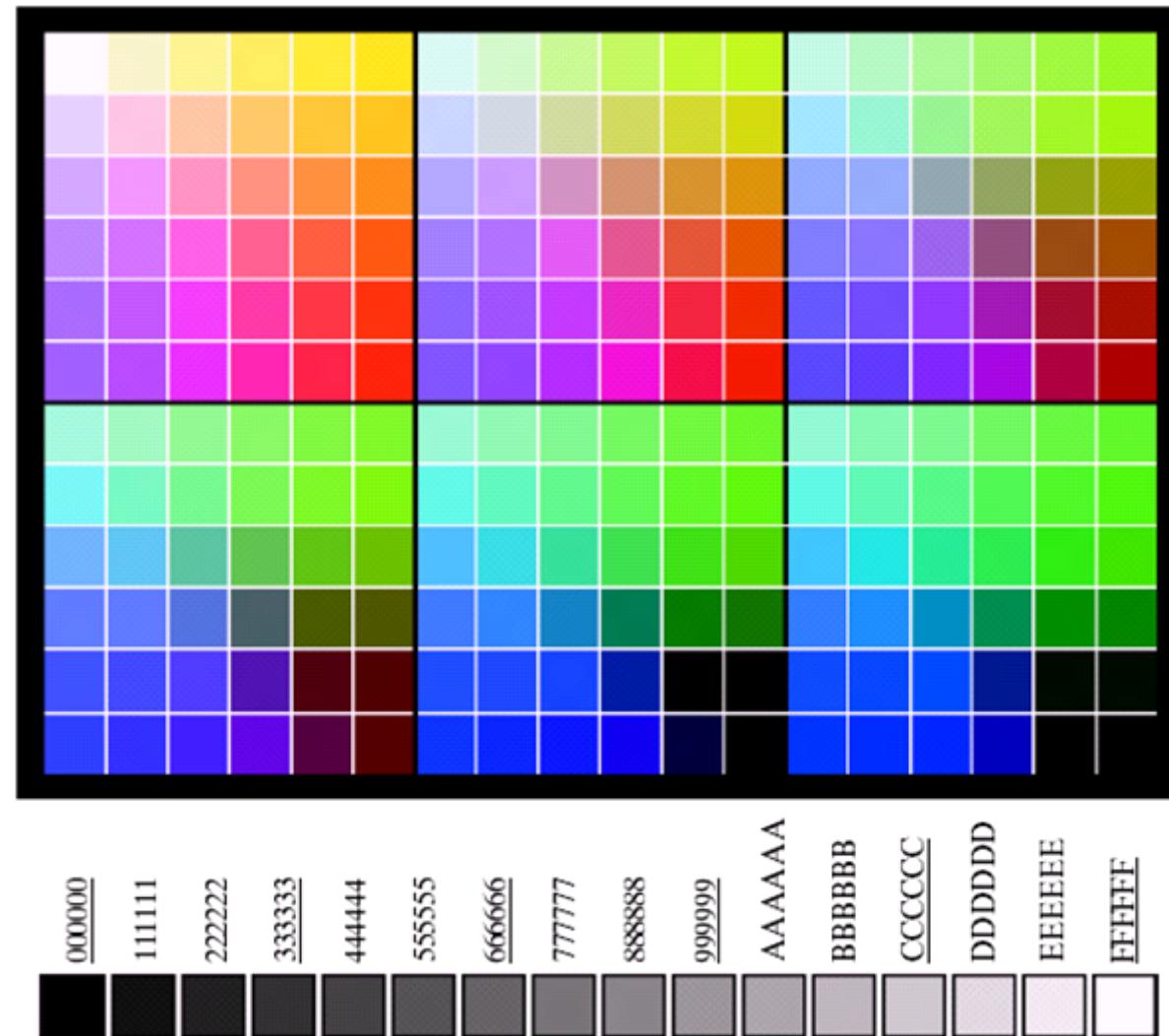


Fig. 5.13: (a) The 216 safe RGB colors. (b) All the grays in the 256-color RGB system (grays that are part of the safe color group are shown underlined). (From [GW18])

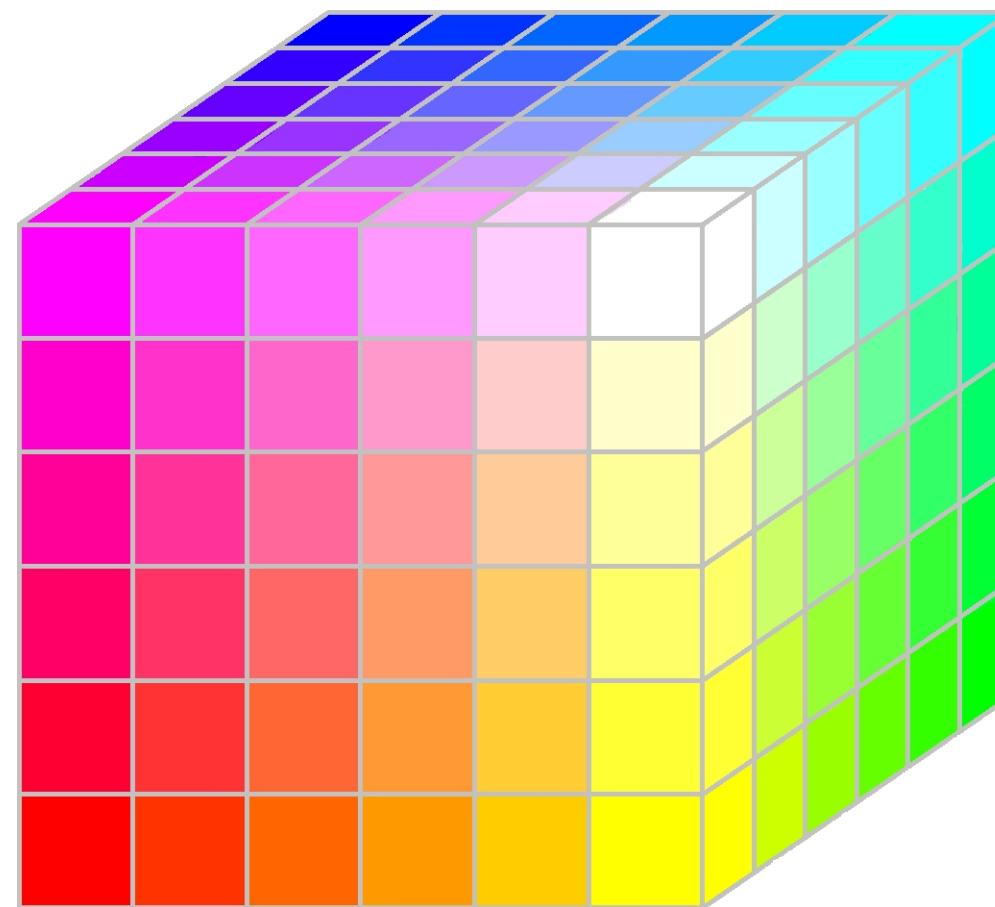


Fig. 5.14: The RGB safe-color cube. (From [GW18])

5.2.2 YUV Color Space

In Europe, analog color television has been based on the YUV model.

Note: The model was originally developed for backward compatibility with black-and-white television.

While the first show of back-and-white television was in 1926, the first general television for color reception took place only in 1956 in USA (Germany: 1967).

Components:

- **Y:** Luminance (brightness)
- **U and V:** Chrominance (color)

YUV signals are generated from an original RGB source (System PAL)³:

$$Y = 0.299 R + 0.587 G + 0.114 B \quad (5.3)$$

$$\begin{aligned} U &= 0.492(B - Y) \\ &= -0.147 R - 0.289 G + 0.436 B \end{aligned} \quad (5.4)$$

$$\begin{aligned} V &= 0.877(R - Y) \\ &= 0.615 R - 0.515 G - 0.100 B \end{aligned} \quad (5.5)$$

Fig. 5.15 shows an example of the UV color plane and fig. 5.16 the YUV color space generated from the RGB color space.

Fig. 5.17 visualizes different views of the YUV color space.

³ The French SECAM system uses slightly different coefficients in its YUV model.

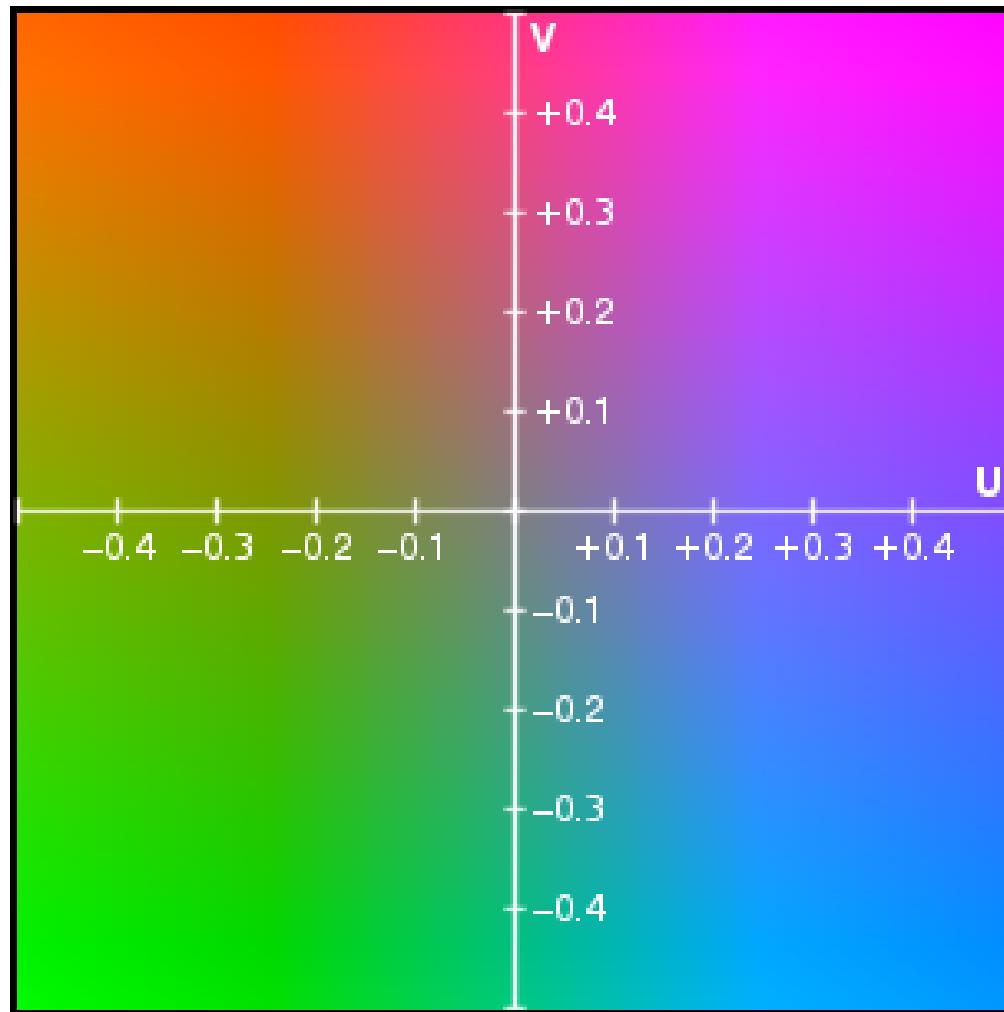


Fig. 5.15: Example of UV color plane, Y value = 0.5, represented within RGB color gamut. (From <http://en.wikipedia.org/wiki/YUV>)

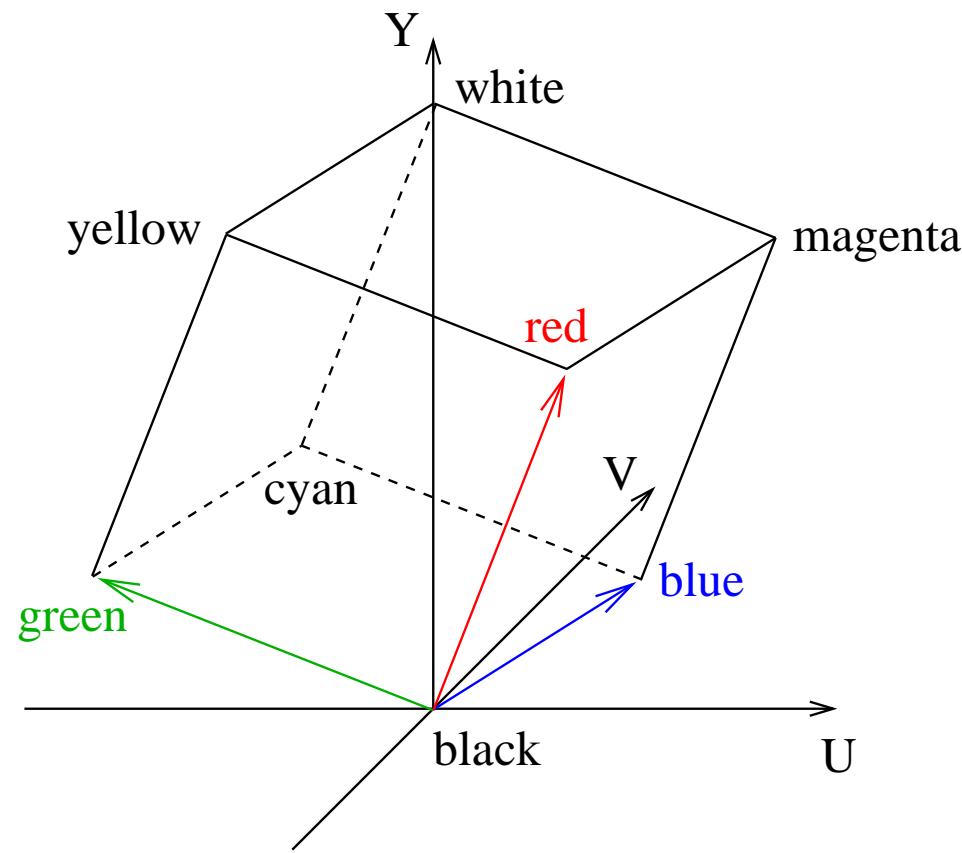


Fig. 5.16: YUV color space derived from the RGB color cube.

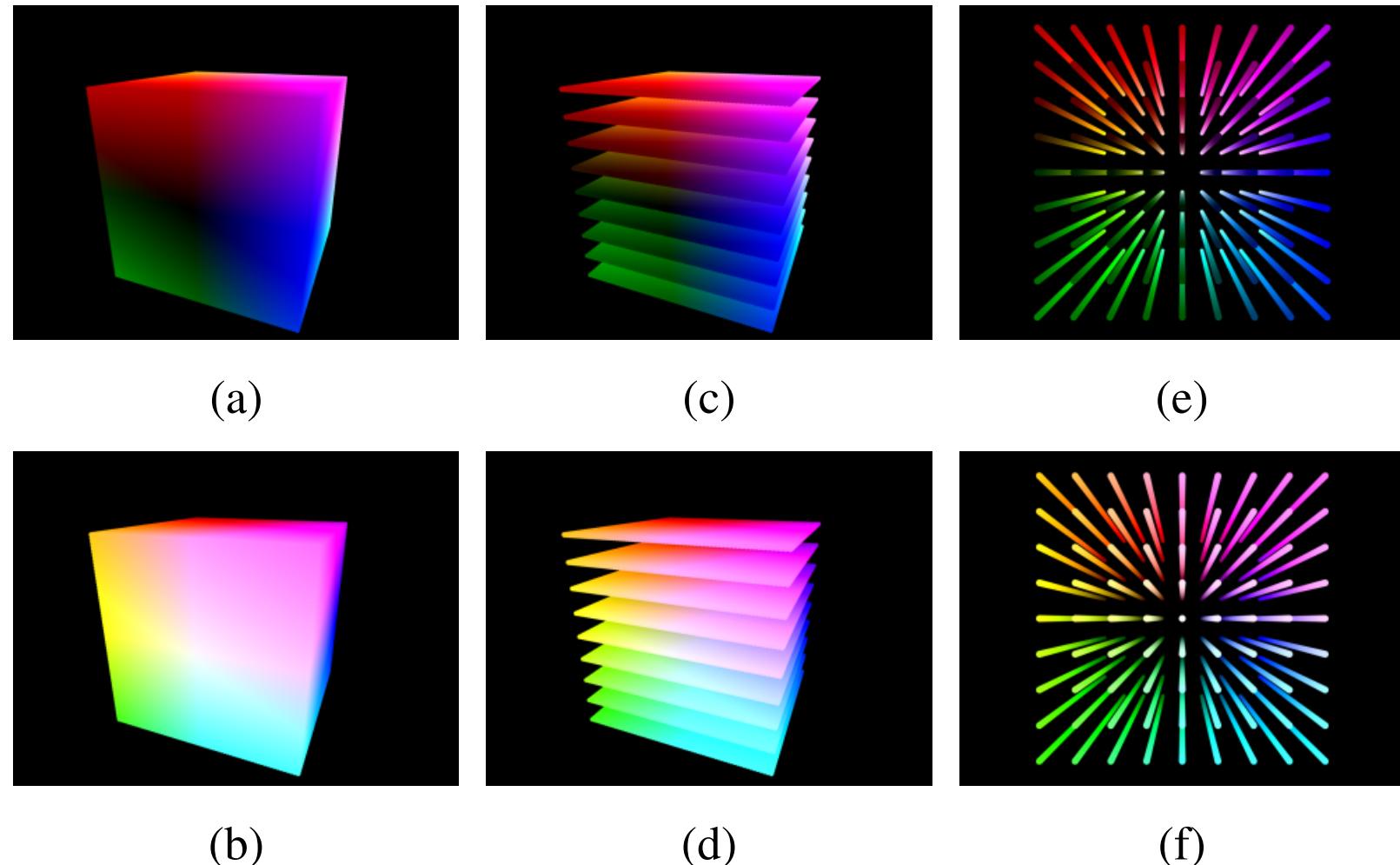


Fig. 5.17: Views of the YUV color space: (a) Look at the dark side ($Y=0$). (b) Bright side ($Y=1$). (c) and (d): Same cubes divided into layers. (e) and (f): Rods at various points of the UV plane, extending through Y . (From <http://softpixel.com/~cwright/programming/colorspace/yuv/>)

On the American continent (USA, Canada, ...) and Southwestern Asia (Japan) the color model is called **YQI**, the system **NTSC**:

$$Y = 0,299 R + 0,587 G + 0,114 B \quad (5.6)$$

$$Q = 0,48(R - Y) + 0,41(B - Y) \quad (5.7)$$

$$I = 0,74(R - Y) - 0,27(B - Y) \quad (5.8)$$

Note: In all models, luminance is calculated in the same manner.

This fact allows the reproduction of black-and-white image on TV sets compiling to different norms.

Besides the backward compatibility, another advantage is that the YUV signal can be easily manipulated to deliberately discard some information in order to reduce band width.

5.2.3 Digital Television and Photography

Color processing is derived from the YUV color model:

- Separation of the RGB signal into an intensity and two color difference signal (B-Y chrominance and R-G chrominance).
- Based on psychovisual findings: Color opponent processes in the human visual system.
- In the digital domain: No separation into PAL, SECAM or NTSC.

Use of human visual perception:

- **Chroma subsampling:** Reduction of resolution of chromaticity channels compared to intensity channel without perceivable loss.
- JPEG encoding can yield high compression rates (up to 50 %).

YCbCr color space:

- First: mapping of the RGB values to the range [0,1] (**Gamma correction.**
New values: R' , B' , G' .
- Second: calculation of the **YPbPr** values.
- Gamma correction affects intensity signal: $Y \rightarrow Y'$.
- Third: Calculation of Y' , C_r and C_b .

$$Y' = K_r R' + (1 - K_r - K_b)G' + K_b B' \quad (5.9)$$

$$P_b = 0.5 \frac{B' - Y'}{1 - K_b} \quad (5.10)$$

$$P_r = 0.5 \frac{R' - Y'}{1 - K_r} \quad (5.11)$$

The constants K_r and K_b are selected with respect to the selected RGB color space.

ITU-R BT.601 conversion for standard definition television:

Constants: $K_r = 0.114$ and $K_b = 0.299$

Digital Y'CbCr:

$$Y' = 16 + (65.481R' + 128.553G' + 24.966B') \quad (5.12)$$

$$P_b = 128 + (-37.797R' - 74.203G' + 112.0B') \quad (5.13)$$

$$P_r = 128 + (112.0R' - 93.786G' - 18.214B') \quad (5.14)$$

5.2.4 CMY and CMYK Color Models

Note: Intuition from one's finger pain suggests

- The primary colors should be red, yellow, and blue, and
- Red and green mix to make yellow.

But again:

Intuition does not apply to color painting because the used pigments mix subtractively.

Features of pigments:

- They remove color from incident light, which is reflected from paper.
- E. g. a red ink is a dye that absorbs green and blue light, while incident red light passes through this dye and is reflected from the paper.

Color spaces for this kind of subtractive matching can be quite complicated.

Simplest case:

- Mixing is *linear*, and
- the CMY space applies (cyan-magenta-yellow)

These primaries subtract a *light* primary from white light:

Cyan: $C = W - R$ (white - red)

Magenta: $M = W - G$ (white - green)

Yellow: $Y = W - B$ (white - blue)

Now: The appearance of mixtures can be evaluated by reference to the RGB color space.

Example 5.3 Mixture of Cyan and Magenta

$$\begin{aligned} C + M &= (W - R) + (W - G) \\ &= R + G + B - R - G \\ &= B \end{aligned}$$

Note: $W + W = W$
→ Ink cannot cause paper to reflect more light than it does uninked.

Devices that deposit colored pigments on paper (color printers or copiers) require

- CMY data input or
- Perform an RGB to CMY conversion internally.

RGB to CMY conversion:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (5.15)$$

The range of primaries is 0 % to 100 %:

- 0 %: Plain page
- 100 %: Full hue page

Their mixture yields the CMY color cube depicted in fig. 5.18.

Practical printing devices use at least four inks.

→ **Four-color printing**

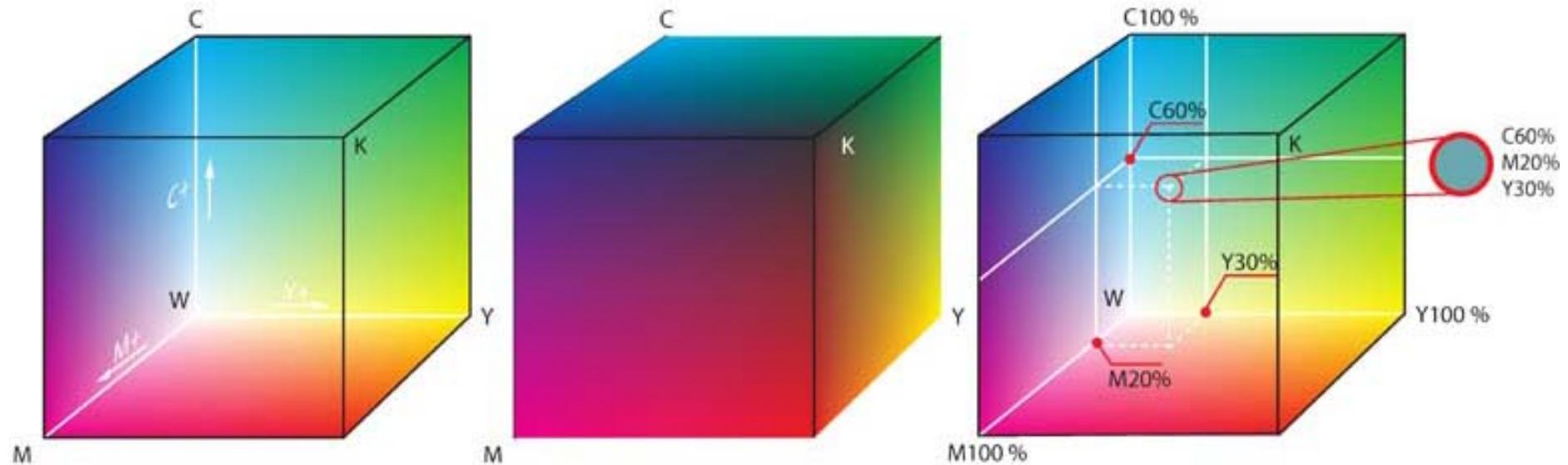


Fig. 5.18: CMYK color cube. (From [<http://de.wikipedia.org/wiki/CMY>])

The key color black is added, because

- Mixing color inks leads to a poor black
→ Produce **true** black
- It is difficult to ensure good registration between the three color inks to avoid colored halos around text.
- Colored inks tend to be more expensive than black inks.

Generally:

It is difficult to produce good results with a color printing process:

- Different inks have different spectral properties.
- The same holds for different printing papers.
- Inks can mix nonlinearly.

High grade inkjet printers use additionally a light magenta and a light cyan.

→ **Six-color printing**

5.2.5 HSI Color Model

So far: Consideration of color spaces with the following features:

- Ideally suited for hardware implementations
- Easy conversion of color spaces

Now: Perceptual color space which is intuitive

→ **HSI model.**

Color description by the components

- **Hue**
- **Saturation**
- **Intensity (brightness)**

(compare sec. 5.1).

Development of new color models from the RGB color space:

- Extract the feature intensity from the RGB color cube in fig. 5.11 and put it on the black vertex $(0, 0, 0)$, with the white vertex $(1, 1, 1)$ directly above it (see fig. 5.19).
- Considering a plane perpendicular to the intensity axis
 - the saturation increases with the distance to the intensity axis
- The hue value can be generated from a RGB point using a plane defined by three points (in fig. 5.19(b)): black, white, and cyan).

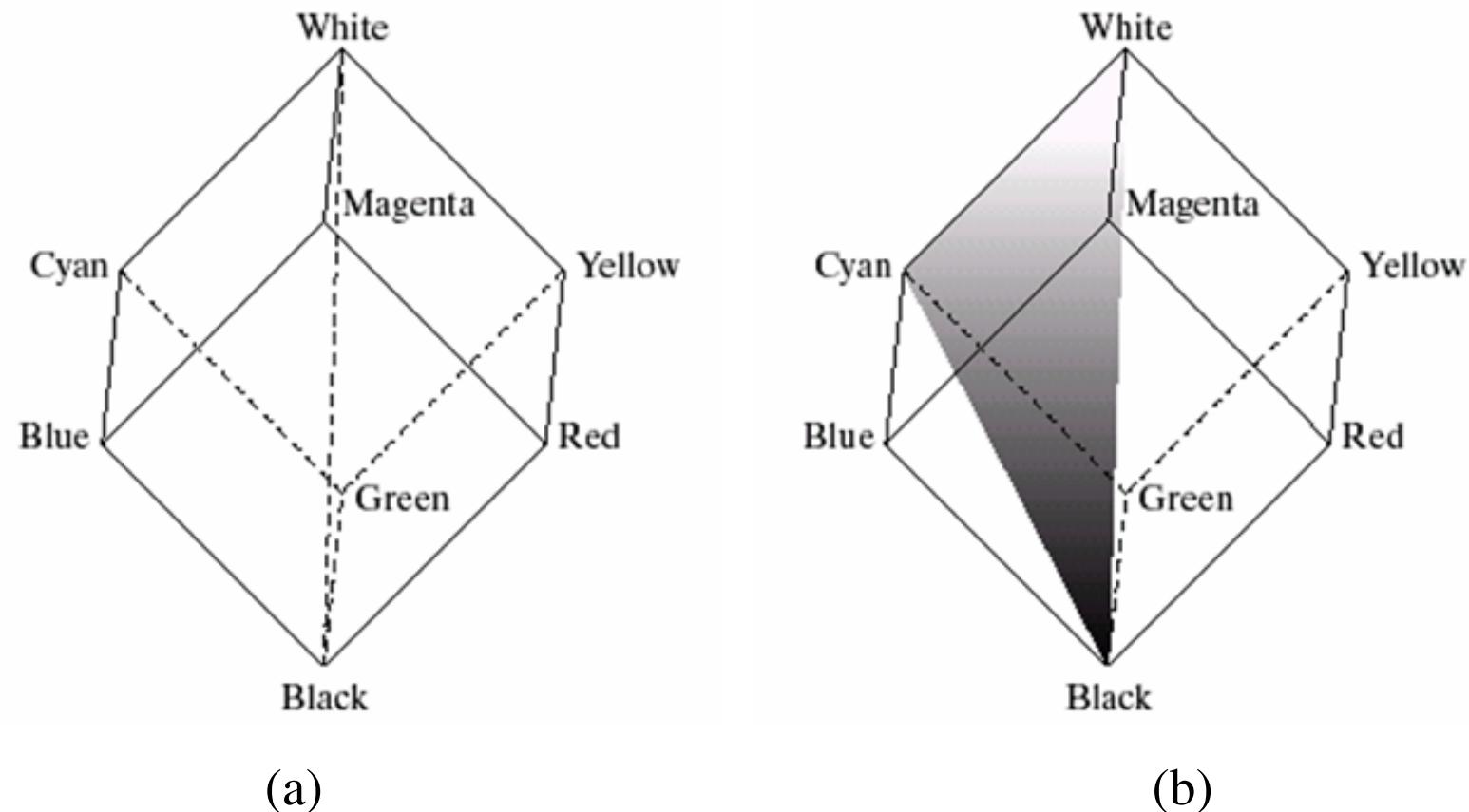


Fig. 5.19: Computational relationships between the RGB and HSI color models. (From [GW18])

Note: Same conclusion as in section 5.1:

All colors generated by three colors lie in the triangle by the colors.

Remember: The HSI space is represented by a vertical intensity axis and by the loci of color points that lie perpendicular to this axis.

Hue can be described by an angle from a reference point (usually red):

- The primary colors are separated by 120° .
- The secondary colors are 60° from the primaries.
- Therefore the secondary colors are separated by 120° , also.

The HSI planes can be defined in terms of a hexagon, a triangle, or a circle (see fig. 5.20(b), (c), and (d), respectively.)

The shape chosen does not matter: Using a geometric transformation, the shapes can be warped into one of the others.

Fig. 5.21 shows the HSI model based on colors triangles and circles.

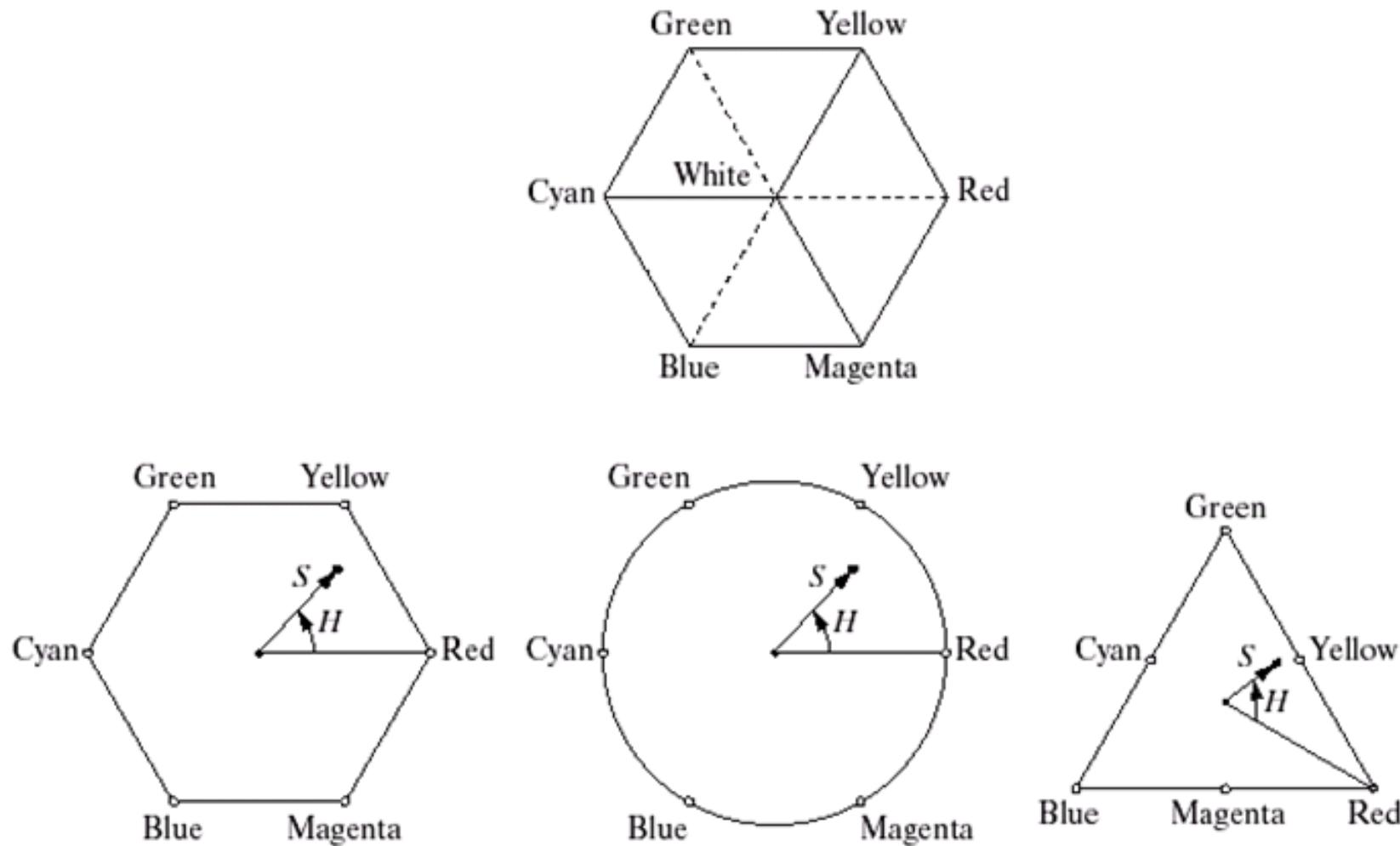


Fig. 5.20: Hue and saturation in the HSI color model. The dot is an arbitrary color point. The angle from the red axis gives the hue, and the length of the vector is the saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axes. (From [GW18])

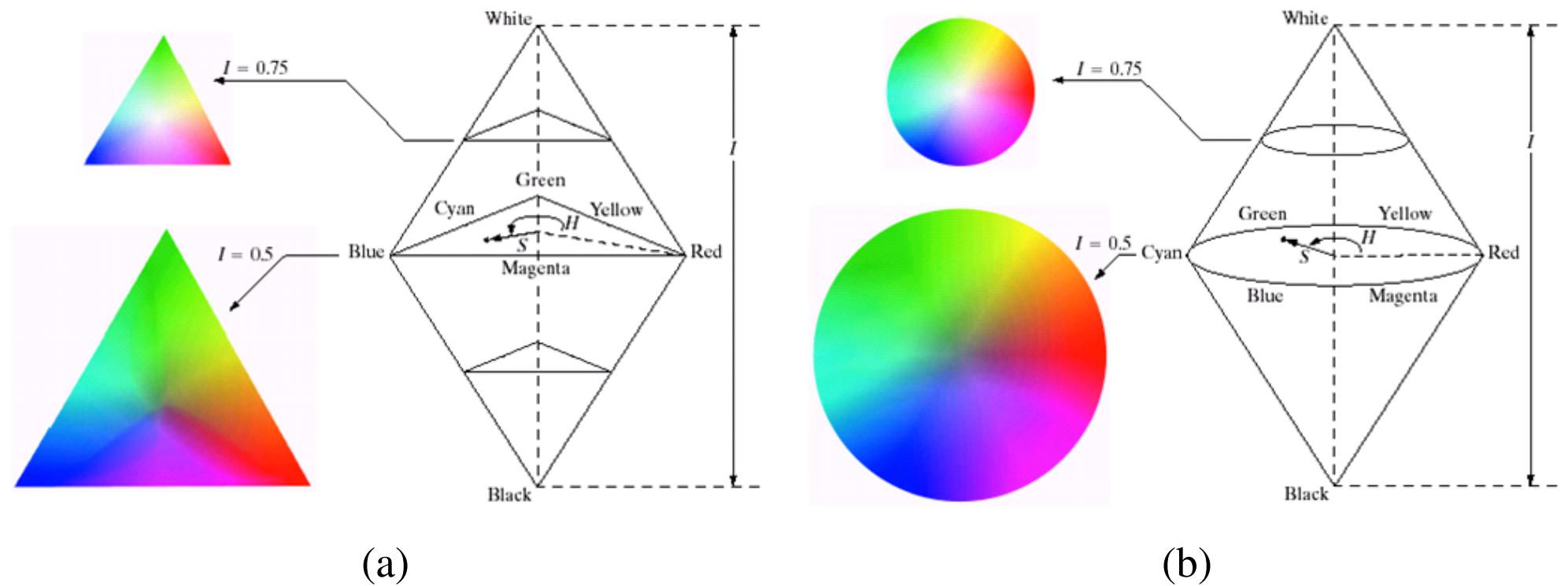


Fig. 5.21: The HSI color model based on (a) triangular and (b) circular color planes. The triangles and circles are perpendicular to the vertical intensity axis. (From [GW18])

Note: The HSI color space which can be viewed graphically as a double cone or double hexcone is similar to the HSV color space (hue, saturation, value) depicted as a color cone or hexcone.

The HSI color model is preferred because it better reflects the intuitive motion of “saturation” and “intensity” as two independent parameters.

Converting Colors from RGB to HSI

Given an image in RGB color format, the HSI components can be computed as follows:

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360^\circ - \theta & \text{if } B > G \end{cases} \quad (5.16)$$

with

$$\theta = \arccos \left(\frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{\frac{1}{2}}} \right) \quad (5.17)$$

$$S = 1 - \frac{3}{R + G + B} [\min(R, G, B)] \quad (5.18)$$

$$I = \frac{1}{3}(R + G + B) \quad (5.19)$$

- Hint:**
- The RGB values have been normalized to the range $[0, 1]$.
 - Hue has to be normalized to the interval $[0, 1]$, also, by dividing by 360° all values from eq. (5.16).

Converting Colors from HSI to RGB

Given the HSI values in the interval $[0, 1]$ for conversion to RGB, the three sectors of interest, corresponding to the 120° intervals in the separation of primaries, have to be regarded.

Start with multiplying H by 360° , which returns the hue to its original range of $[0^\circ, 360^\circ]$.

Conversion:

- RG sector ($0^\circ \leq H < 120^\circ$)

$$B = I(1 - S) \quad (5.20)$$

$$R = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad (5.21)$$

$$G = 3I - (R + B) \quad (5.22)$$

- GB sector ($120^\circ \leq H < 240^\circ$)

→ first: subtraction: $H' = H - 120^\circ$

$$R = I(1 - S) \quad (5.23)$$

$$G = I \left[1 + \frac{S \cos H'}{\cos(60^\circ - H')} \right] \quad (5.24)$$

$$B = 3I - (R + G) \quad (5.25)$$

- BR sector ($240^\circ \leq H \leq 360^\circ$)

→ first: subtraction: $H' = H - 240^\circ$

$$G = I(1 - S) \quad (5.26)$$

$$B = I \left[1 + \frac{S \cos H'}{\cos(60^\circ - H')} \right] \quad (5.27)$$

$$R = 3I - (G + B) \quad (5.28)$$

Fig. 5.22 shows the HSI components for the RGN color cube in fig. 5.22:

Hue (fig. 5.22(a)):

- Discontinuity in a value along the 45° line in the front (red) plane of the cube.
- Interpretation:
Start in the frontplane of fig. 5.11, select the middle point in the plane, start at that point, and draw a path to the right until return to the starting point.

Path: $Y \rightarrow G \rightarrow C \rightarrow B \rightarrow M \rightarrow R$

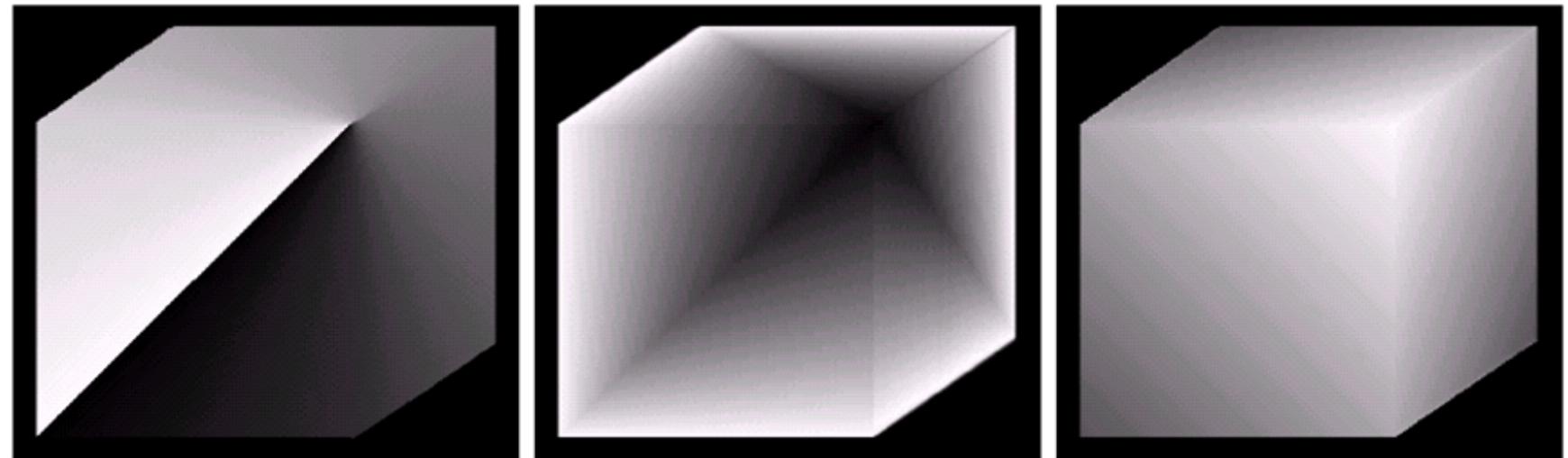
The hue values along this path increases from 0° to 360°
→ Discontinuity at the starting point.

Saturation (fig. 5.22(b)):

The image shows progressively dark values toward the white vertex of the RGB cube, indicating the colors become less and less saturated as they approach white.

Intensity (fig. 5.22(c)):

Every pixel is the average of the RGB values at the corresponding pixels in fig. 5.11.



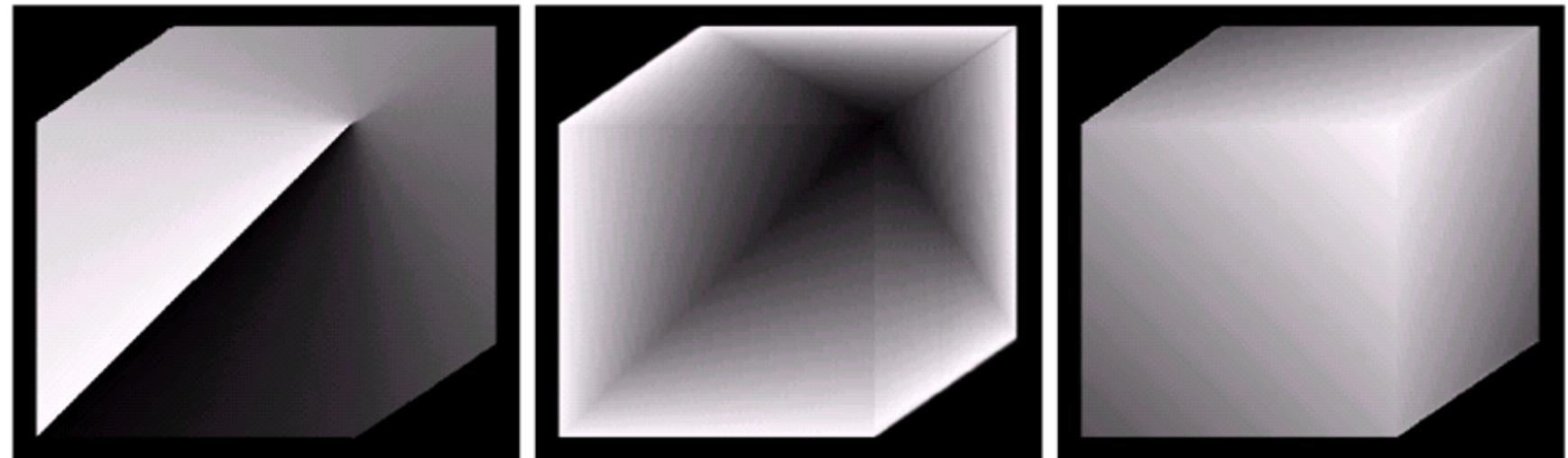


Fig. 5.22: HSI components of the image in fig. 5.11. (a) Hue, (b) saturation, and (c) intensity images. (From [GW18])

Manipulating HSI Component Images

Fig. 5.23(a) shows the primary and secondary RGB colors.

For illustration, in fig. 5.23(b), (c), and (d) the corresponding HSI-components are depicted.

The independent control of the perceptual quantities H, S, and I is illustrated in fig. 5.24:

- (a): The pixels in fig. 5.23(b) corresponding to green and blue are set to H=0.
- (b): The saturation of the cyan region in fig. 5.23(c) is reduced by half (S).
- (c): The intensity of the central white region of fig. 5.23(d) is reduced by half (I)
- (d): The result of converting fig. 5.24(d) back to RGB.

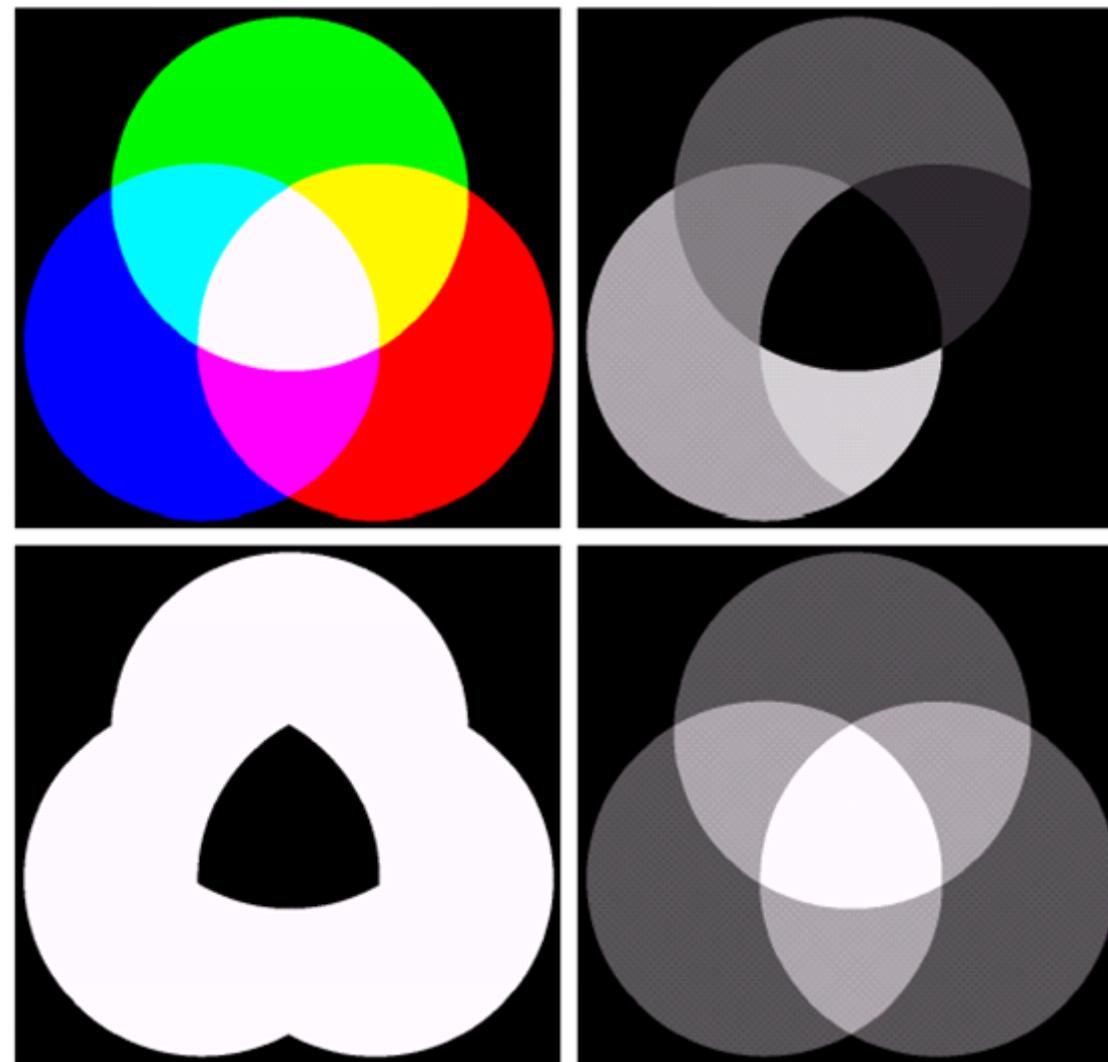


Fig. 5.23: (a) RGB image and the components of its corresponding HSI image: (b) hue, (c) saturation, and (d) intensity. (From [GW18])

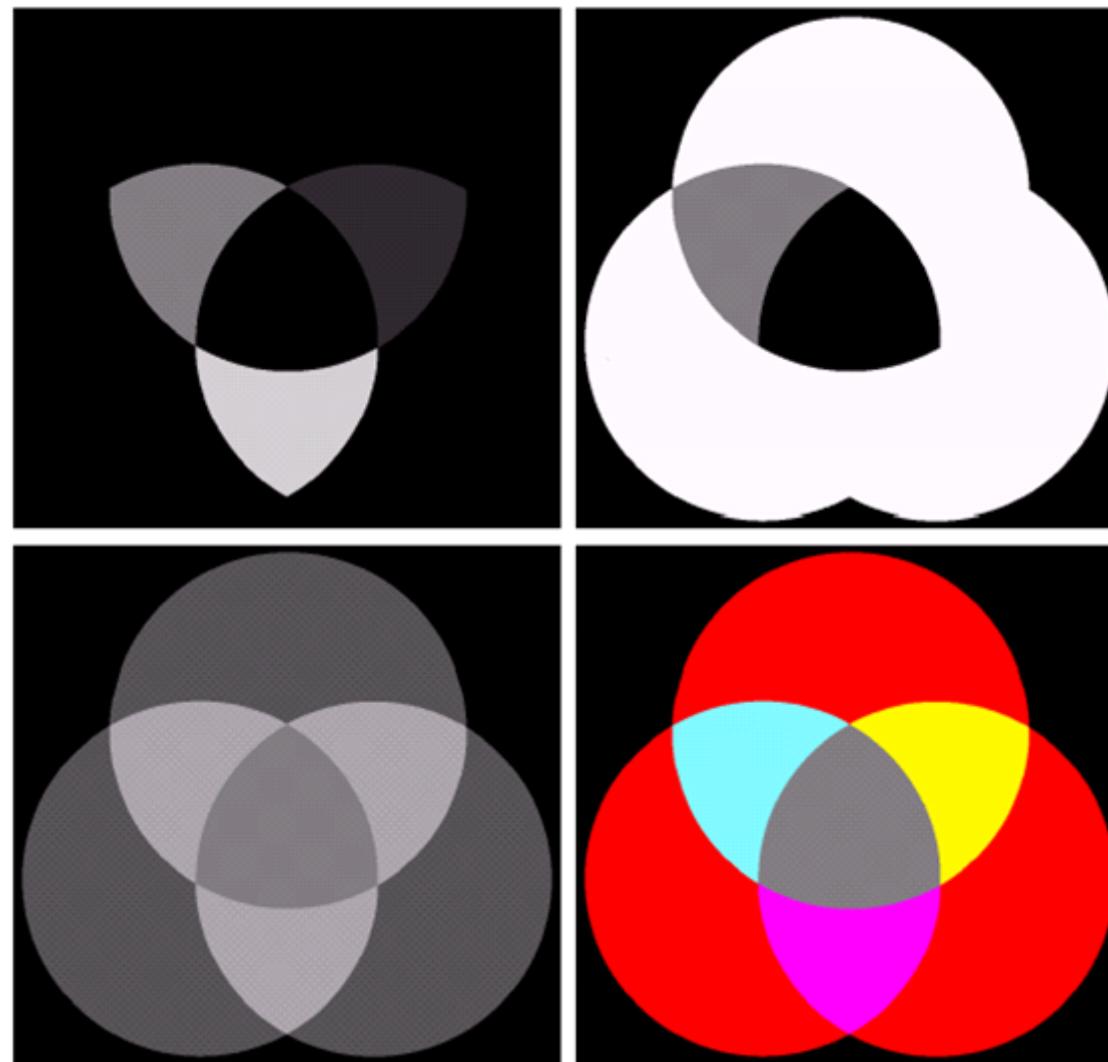


Fig. 5.24: (a)-(c) Modified HSI component images. (d) Resulting RGB image. (See fig. 5.23 for the original HSI images.) (From [GW18])

5.3 Pseudo Color Image Processing

Pseudo color (or false color) image processing consists of assigning colors or gray values to an image based on a specific criterion.

5.3.1 Intensity Slicing

An image is interpreted as a 3D function (intensity versus spatial coordinates).

Slices can be introduced parallel to the coordinate plane (see fig. 5.25).

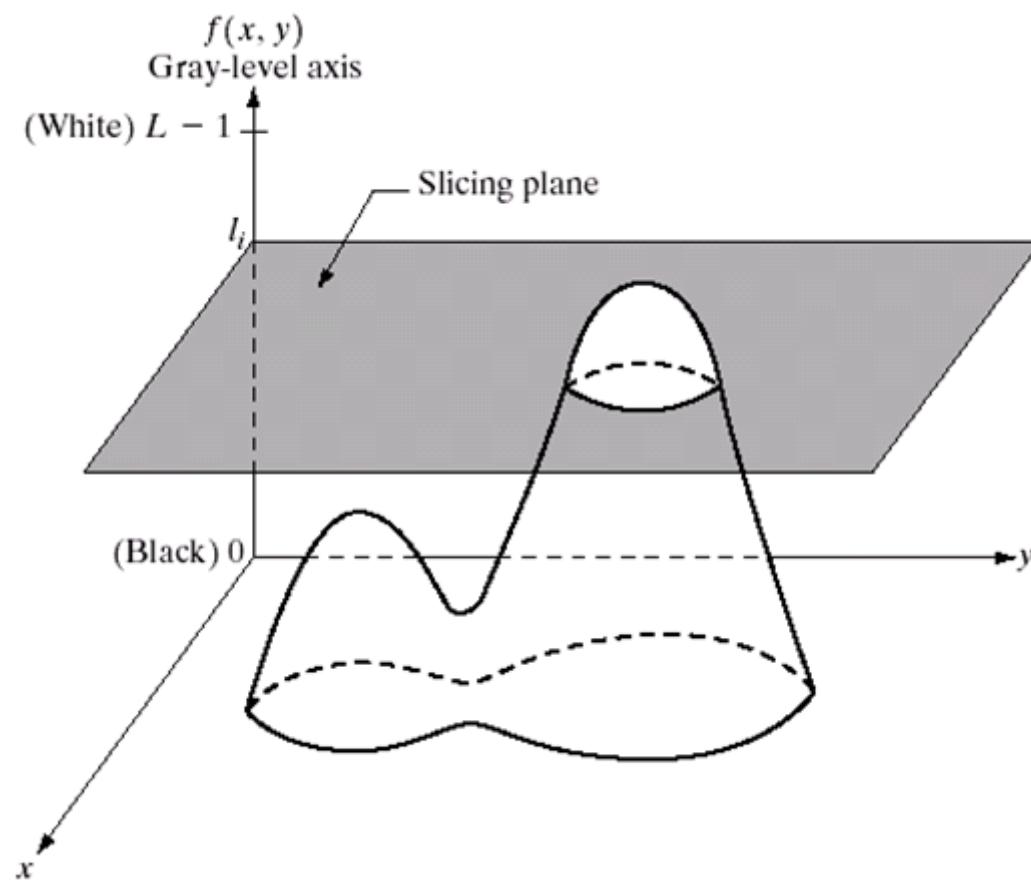


Fig. 5.25: Geometric interpretation of the intensity-slicing technique. (From [GW18])

Fig. 5.26 shows an alternative representation.

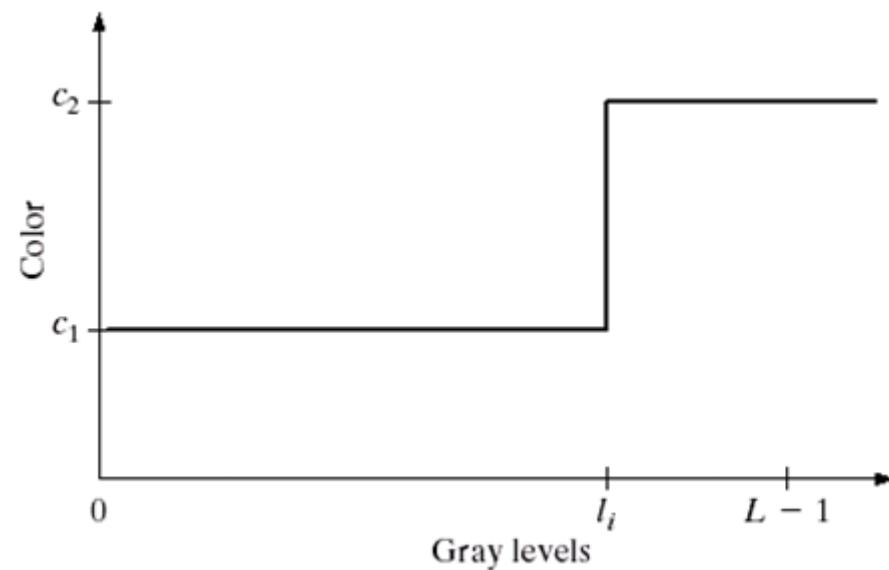
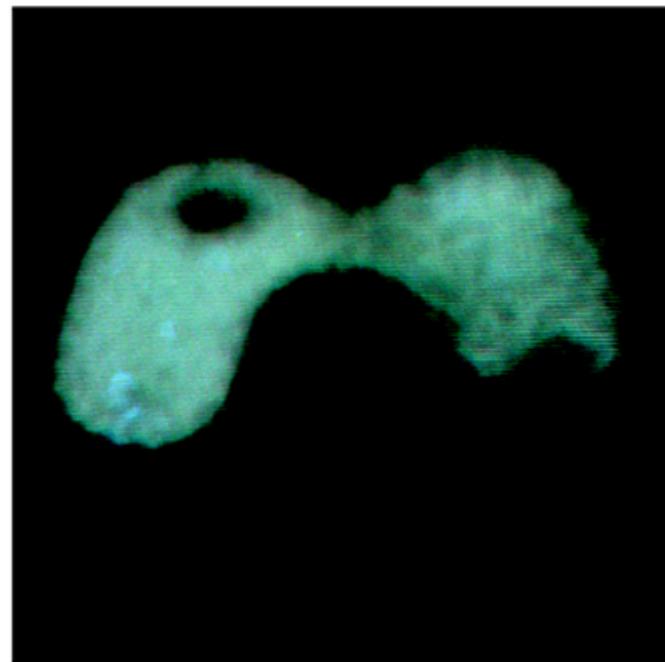
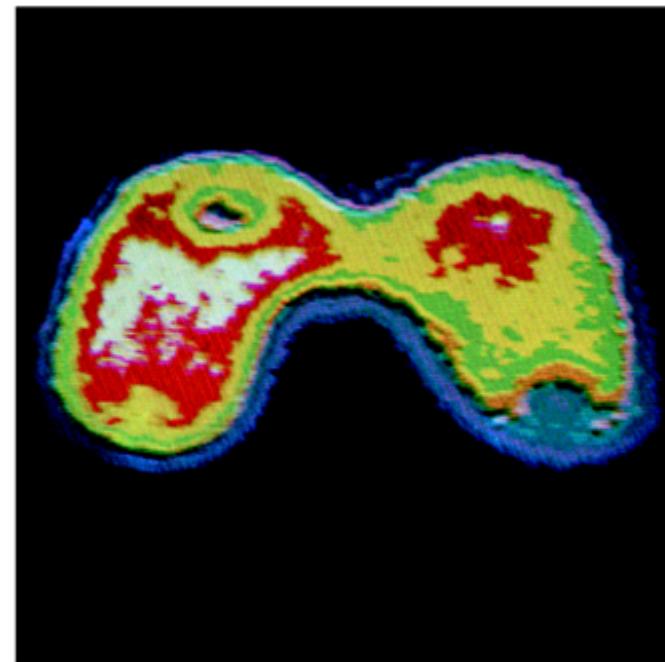


Fig. 5.26: An alternative representation of the intensity-slicing technique. (From [GW18])

An application of intensity slicing is shown in fig. 5.27 (thyroid szintigram).



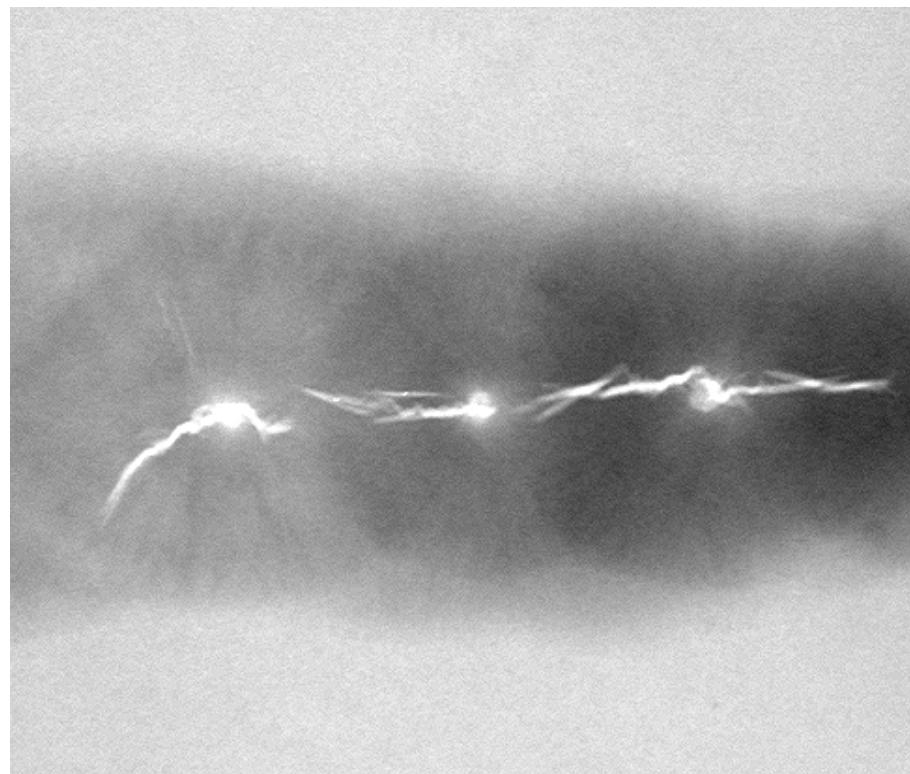
(a)



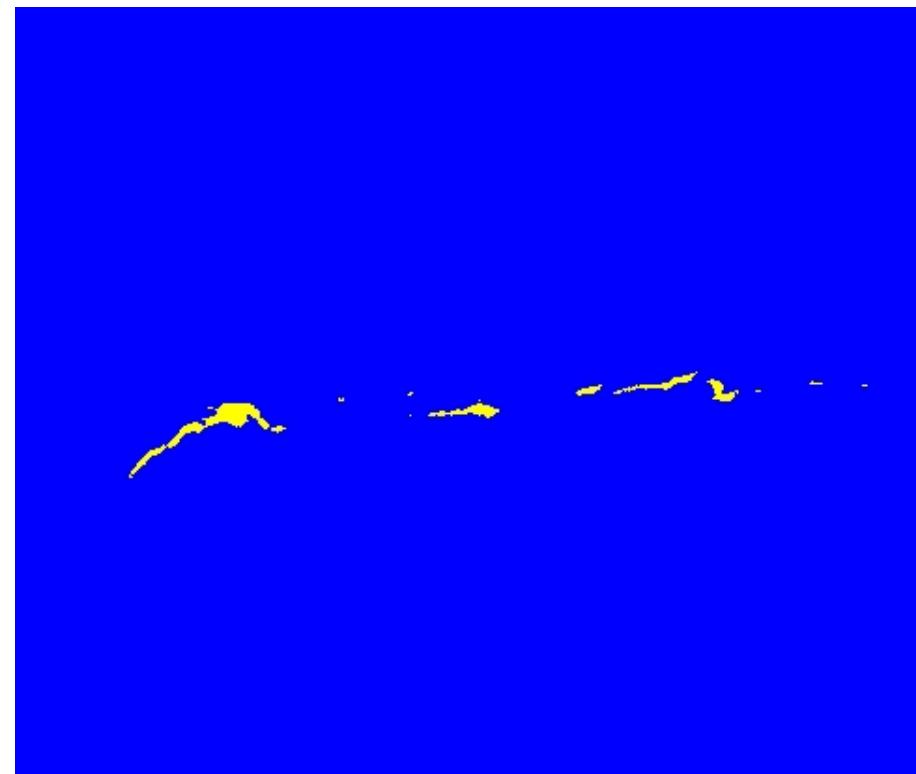
(b)

Fig. 5.27: (a) Monochrome image of the Picker Thyroid Phantom. (b) Result of density slicing into eight colors. (From [GW18])

The interpretation of a weld is illustrated in fig. 5.28.



(a)



(b)

Fig. 5.28: (a) Monochrome X-ray image of a weld. (b) Result of color coding. (From [GW18])

Fig. 5.29 shows results from the TRMM (Tropical Rainfall Measuring Mission) satellite, which uses, among others, three sensors for rain detection (precipitation radar, a microwave imager, a visible and infrared scanner).

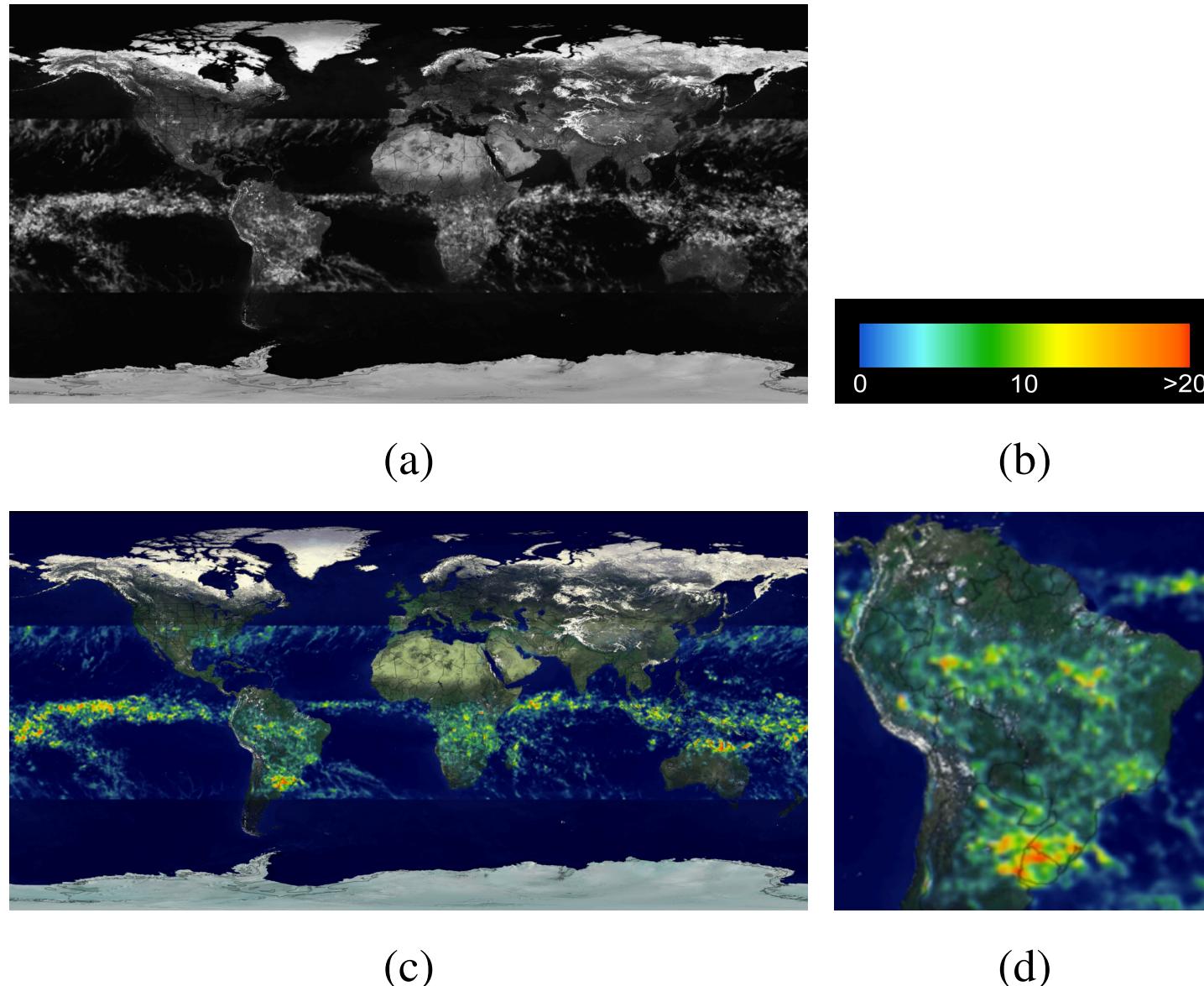


Fig. 5.29: (a) Gray-scale image in which intensity (in the lighter horizontal band shown) corresponds to average monthly rainfall. (b) Colors assigned to intensity values. (c) Color-coded image. (d) Zoom of the South America region. (From [GW18])

5.3.2 Gray Level to Color Transformation

Now: More general transformations than simple slicing.

Fig. 5.30 shows the basic approach.

The analysis of monochrome images of baggage from an airport X-ray scanning system (see fig. 5.31) can be extremely facilitated when using sinusoidal transformation functions, as shown in fig. 5.32.

Adjusting the phase and frequency accordingly allows the detection of an explosive in fig. 5.31(b), while a wrong choice hinders the detection.

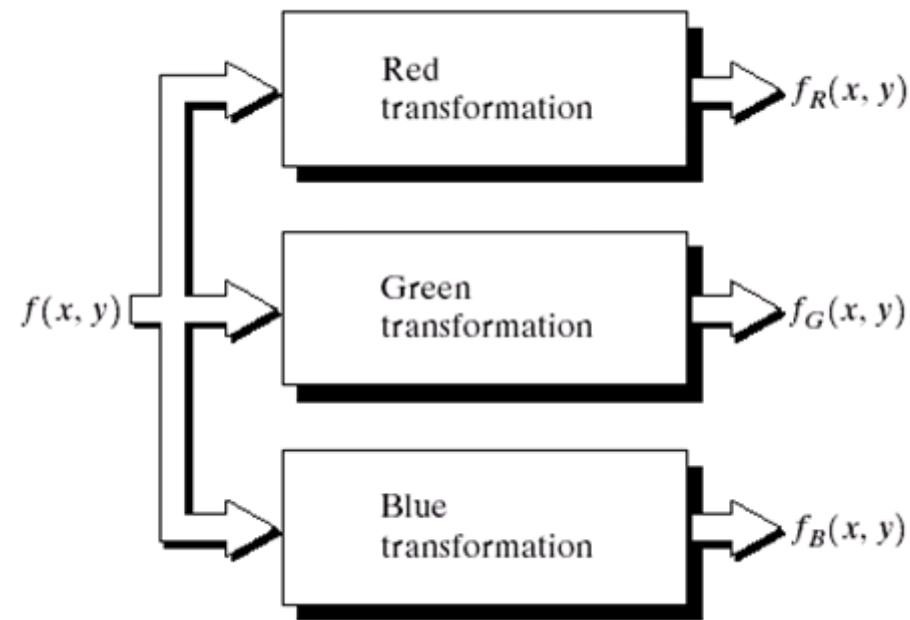


Fig. 5.30: Functional block diagram for pseudo-color image processing. f_R , f_G , and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor. (From [GW18])

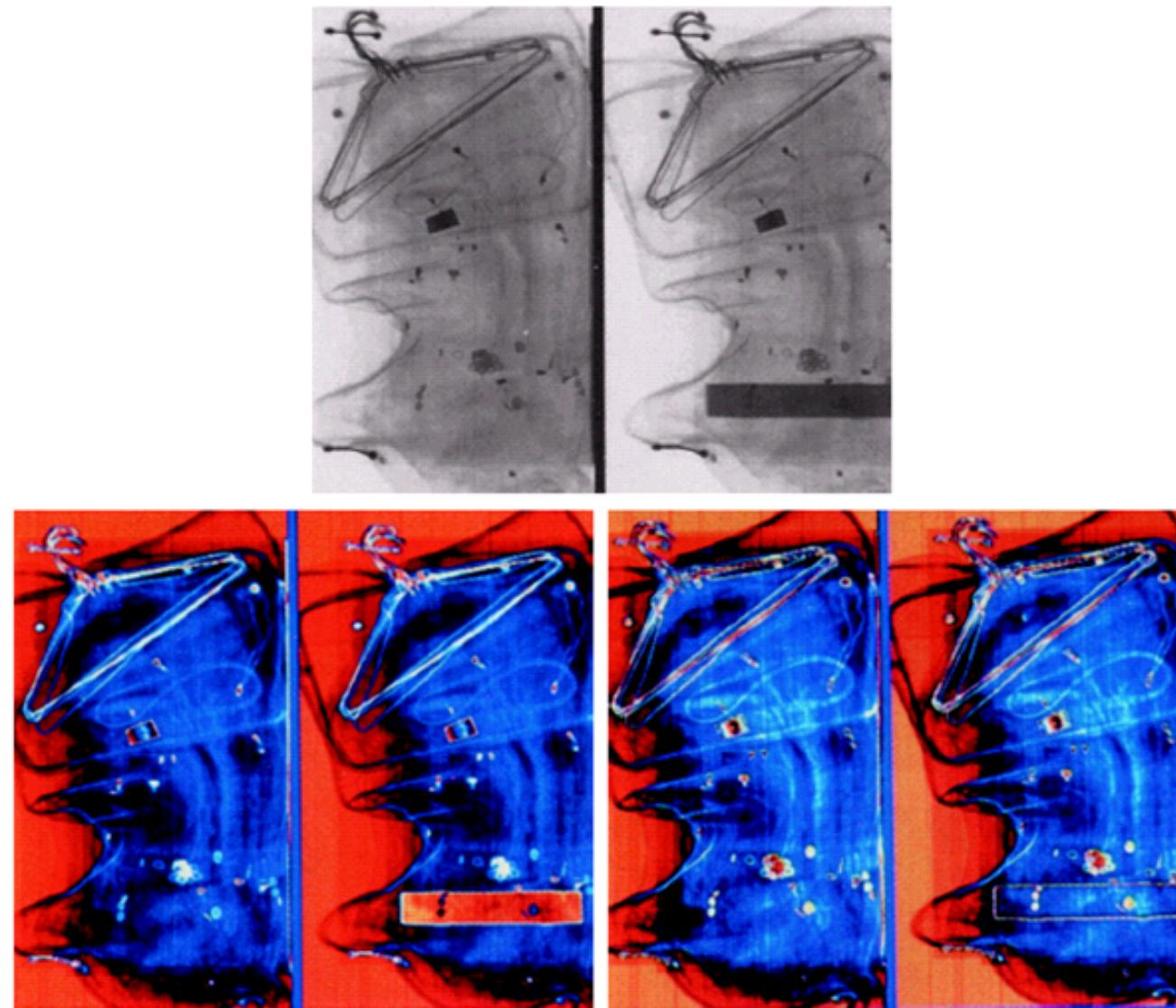
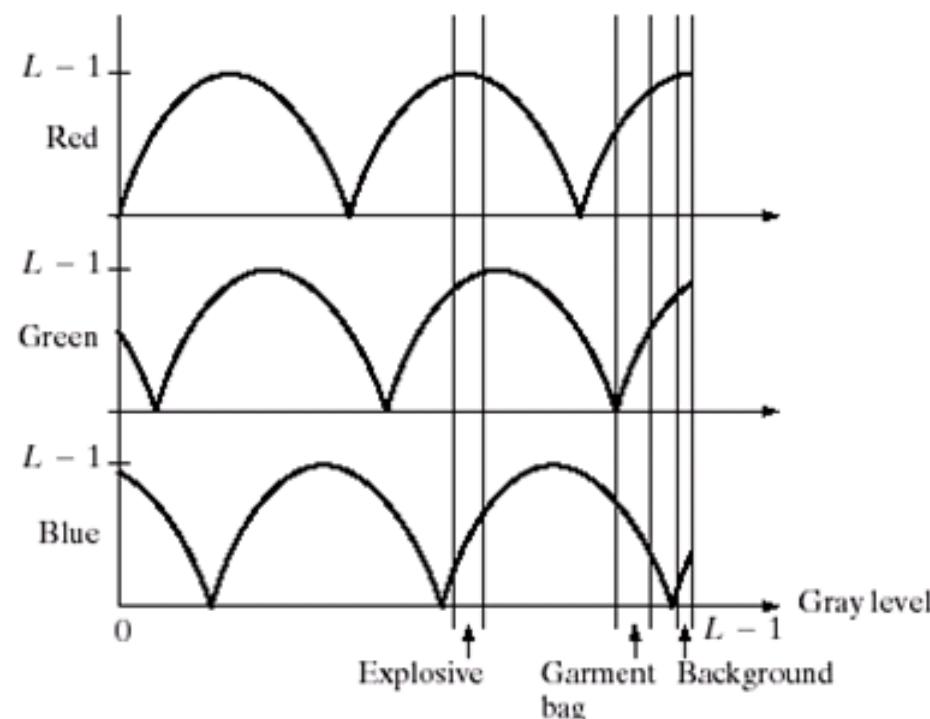
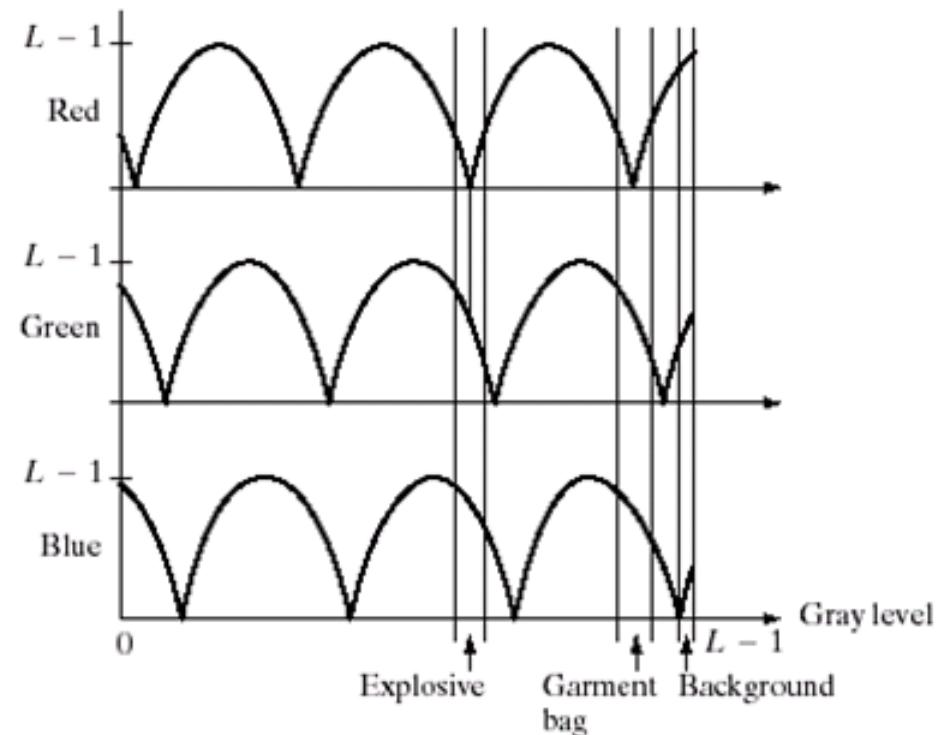


Fig. 5.31: Pseudo-color enhancement by using the gray-level to color transformations in fig. 5.32. (From [GW18])



(a)



(b)

Fig. 5.32: Transformation functions used to obtain the images in fig. 5.31 (a) and (b). (From [GW18])

While in the approach in fig. 5.30 only one monochrome was used, it is often useful to base the analysis on several monochrome images combined to a single color composite, as shown in fig. 5.33.

For the set of spectral satellite images of Washington, D.C., including the Potomac River, a clever combination of transformations allows discrimination of bio mass (in red) and human-made entities in the image (see fig. 5.34).

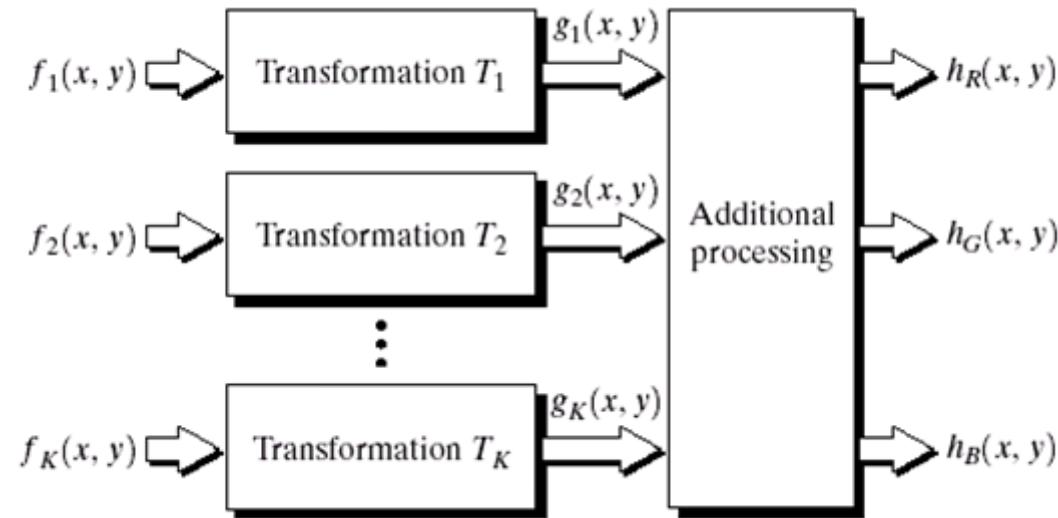


Fig. 5.33: A pseudo-color approach used when several monochrome images are available. (From [GW18])

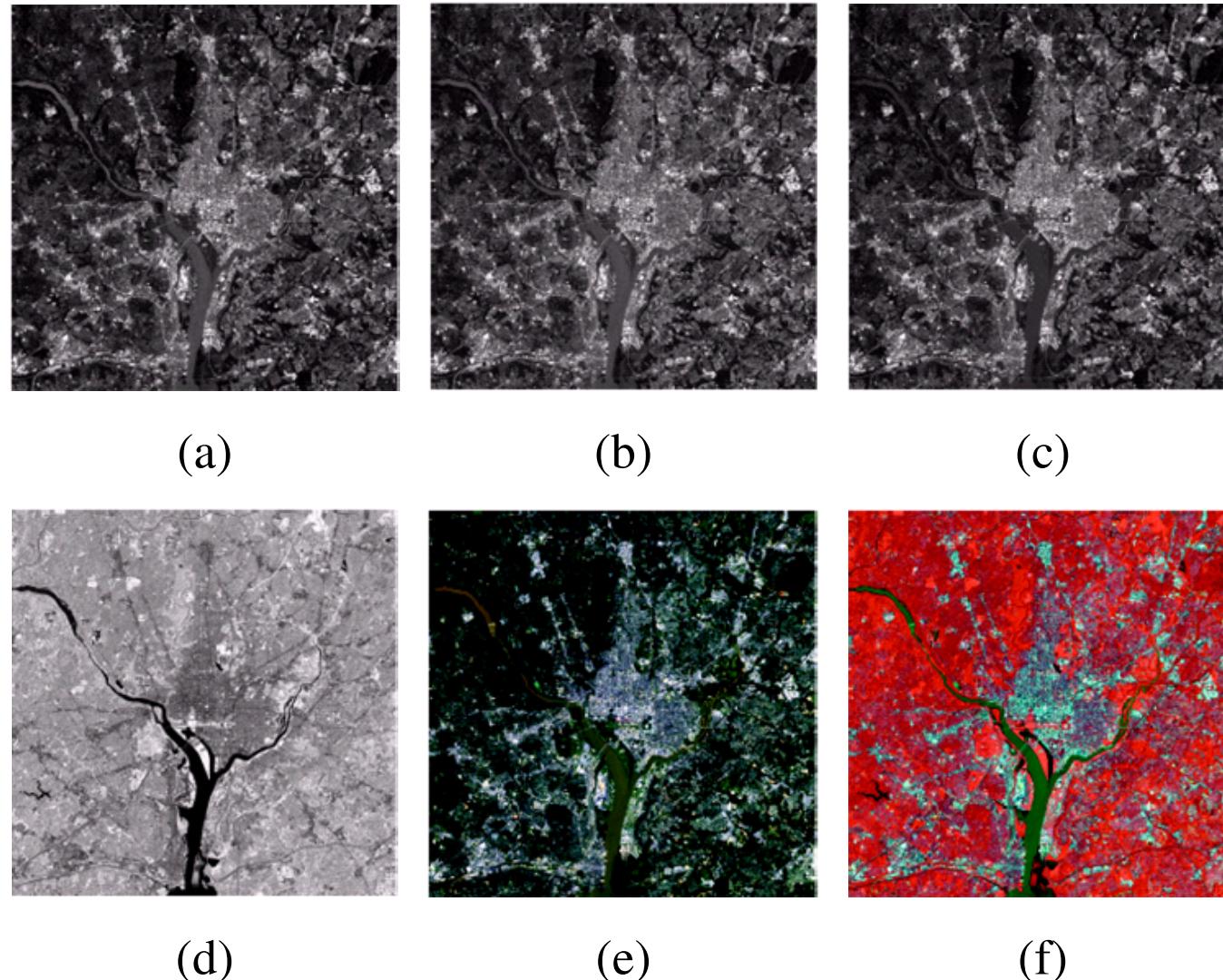
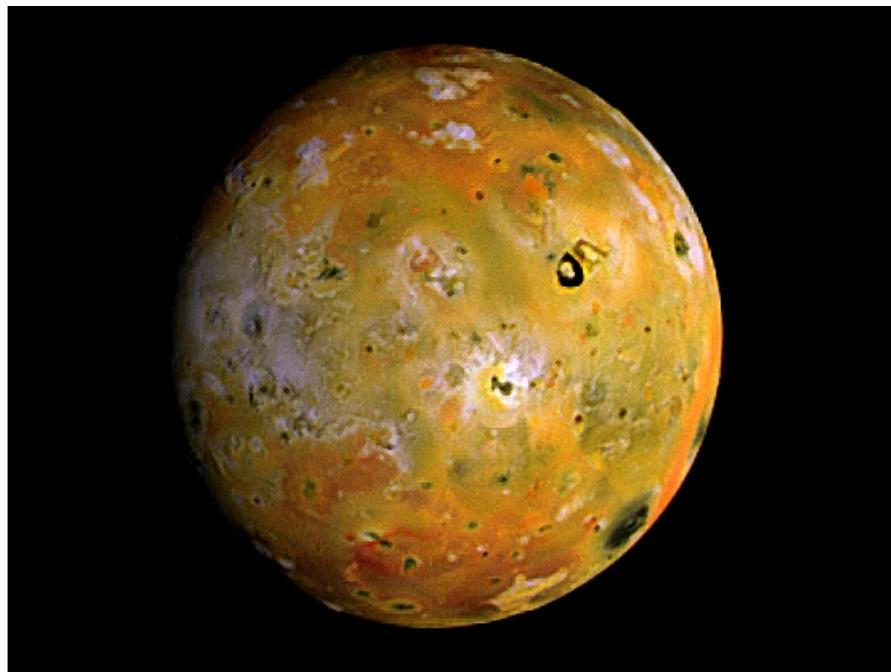


Fig. 5.34: (a)-(d) Images in bands 1-4 in fig. 1.13 (see table 1.10). (e) Color composite image obtained by treating (a), (b), and (c) as the red, green, and blue components of an RGB image. (f) Image obtained in the same manner, but using in the red channel the near-infrared image in (d). (From [GW18])

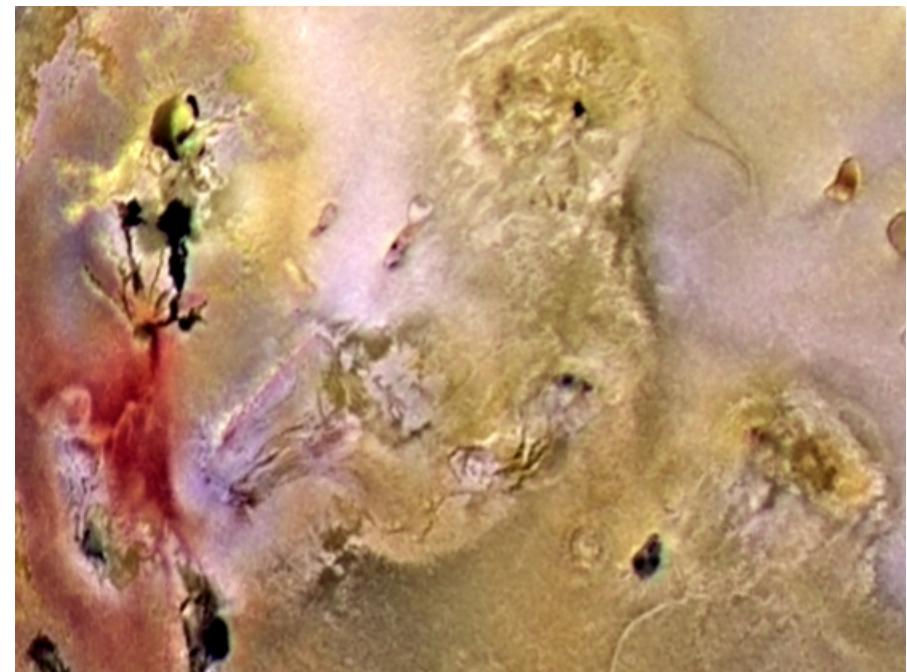
Pseudocolor image processing can even help to illustrate phenomena that are beyond the human normal sensing capabilities.

Fig. 5.35(b) shows the newly ejected output from an active volcano on Io, a Jupiter moon. The surrounding yellow materials are older sulphur deposits.

The result was obtained again by several images from different sensors of the Galileo space-craft.



(a)



(b)

Fig. 5.35: (a) Pseudo-color rendition of Jupiter Moon Io. (b) A close-up. (From [GW18])

5.4 Full-Color Image Processing

Pixels in full-color images have at least three components and are therefore denoted as vectors:

$$\vec{\mathbf{c}}(x, y) = \begin{bmatrix} c_r(x, y) \\ c_g(x, y) \\ c_b(x, y) \end{bmatrix} = \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix} \quad (5.29)$$

where $\vec{\mathbf{c}}(x, y)$: arbitrary vector in the RGB space

It holds furthermore for an image of size $M \times N$:

$$0 \leq x \leq M - 1, \quad 0 \leq y \leq N - 1$$

Alternatives for full-color image processing:

- Processing each image component separately, using standard gray-scale image processing techniques
- Direct processing in the color vector space
 - new approaches required.

Two conditions necessary for equivalent per-color component *and* vector-based processing:

- Process must be applicable to scalars *and* vectors.
- Operation on each component of a vector must be independent of other components.

Fig. 5.36 illustrates the two alternatives using the example of neighborhood spatial processing of gray-scale and full-color images.

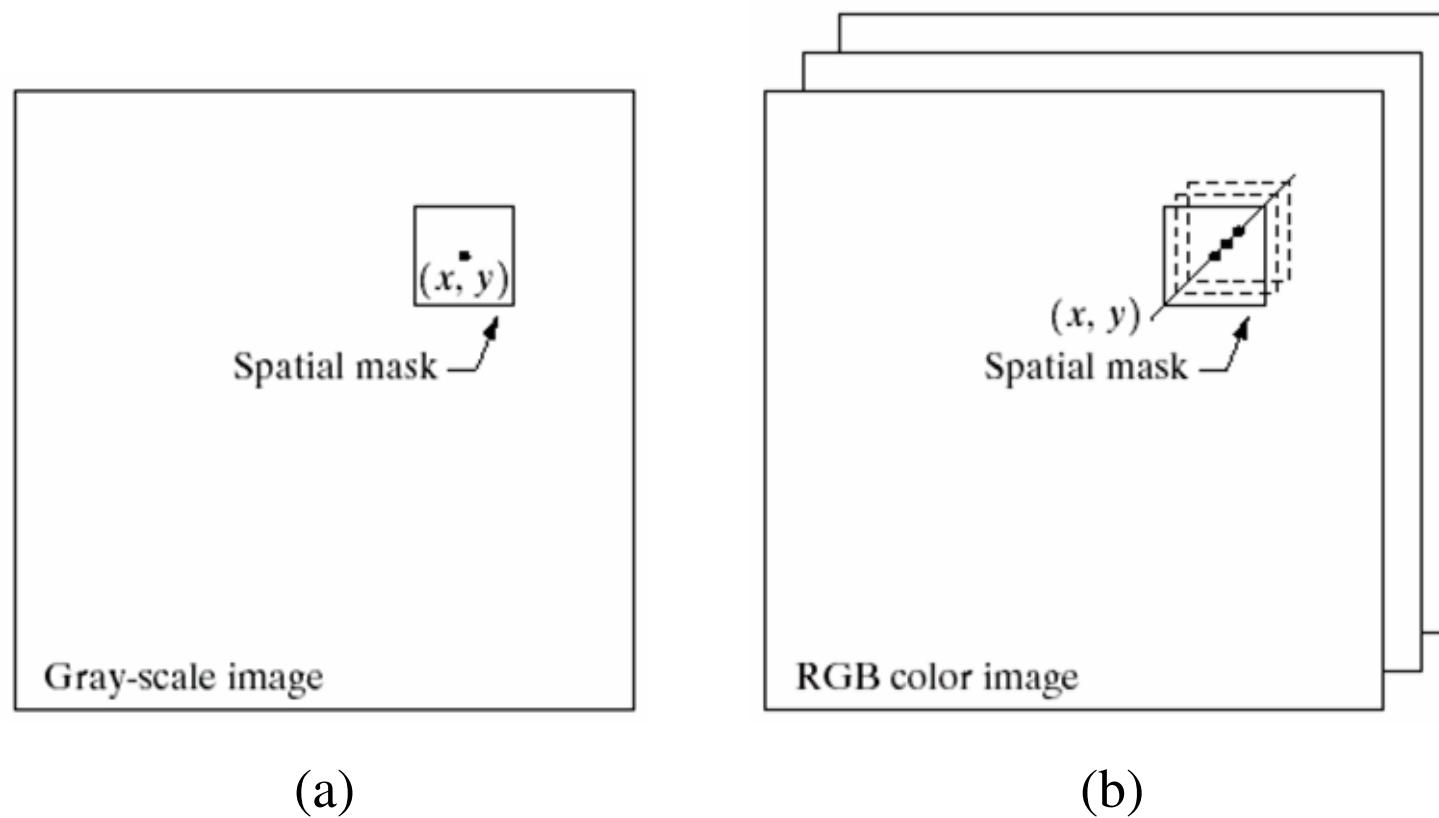


Fig. 5.36: Spatial masks for (a) gray-scale and (b) RGB color images. (From [GW18])

5.4.1 Color Transformations

While in section 5.2 the conversion between components of color models have been presented, now transformations **within** the context of a single color model will be explained.

Use of eq. (3.1):

$$g(x, y) = T[f(x, y)] \quad (5.30)$$

where

$f(x, y)$: color input image

$g(x, y)$: color output image

T : operator on f over a spatial neighborhood of (x, y)

Interpretation: The pixels are triplets or quartets from the chosen color space.

Analogous to the approach in sec. 3.2:

Restriction of attention to transformations:

$$s_i = T_i(r_1, r_2, r_3) \quad (5.31)$$

where

r_i : color component of $f(x, y)$ of any point (x, y)

s_i : color component of $g(x, y)$ of any point (x, y)

$i = 1, 2, \dots, n$

T_i : set of transformation functions

Example 5.4 Full-Color Image and its Components

Fig. 5.37(a) shows a high-resolution color image that was digitized from a large format ($4'' \times 5''$) color negative.

The first row on the right side of the image shows the CMYK scan.

The CMYK image is converted to RGB (third row).

Particularly interesting is the conversion to the HSI color space using the eqn. (5.19)-(5.22).

Note: Hue component:

- Discontinuity in the HSI space between 360° and 0°
- The hue value is undefined for a saturation of 0 (white, black, and pure gray)

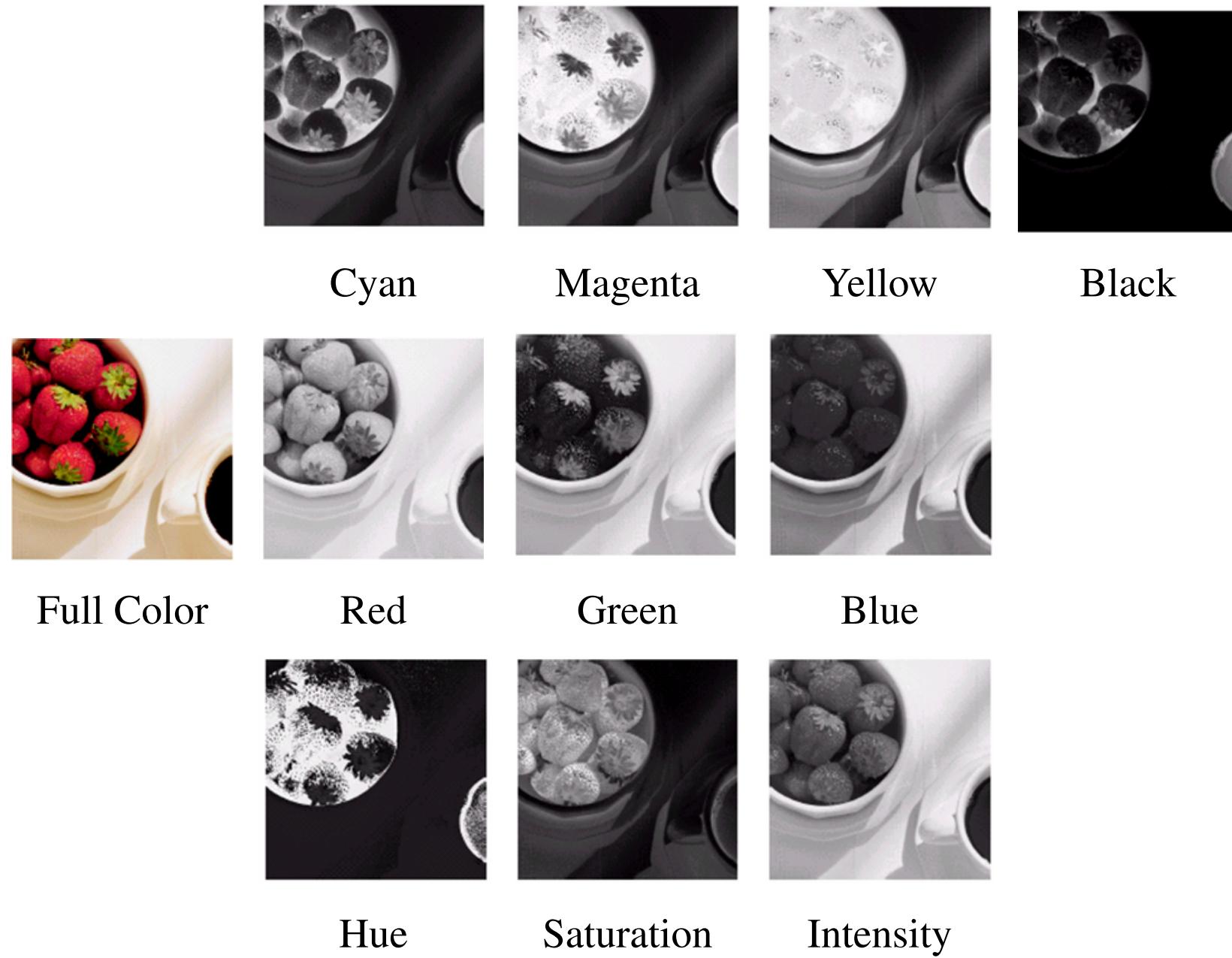


Fig. 5.37: A full-color image and its various color-space components. (From [GW18])

Note: Any color space can be used in conjunction with eq. (5.31).

In practice, some operations are better suited for a specific model.

Example 5.5 Modify the intensity of the images in fig. 5.37(a)

Approach:

$$g(x, y) = k f(x, y) \quad (5.32)$$

with $0 < k < 1$

(a) HSI color space:

$$S_3 = k r_3 \quad (5.33)$$

→ only the intensity component r_3 is modified.

(b) RGB color space:

$$S_i = k r_i, \quad i = 1, 2, 3 \quad (5.34)$$

→ all three components must be transformed.

(c) CMY color space:

$$S_i = k r_i + (1 - k), \quad i = 1, 2, 3 \quad (5.35)$$

→ similar set of transformations as for the RGB color space.

- Interpretation:**
- HSI transformation involves the fewest number of operations,
 - but the conversion of an RGB or CMY input image to the HSI space more than offsets the advantages (at least in this case).
 - The output of that transformations in (a), (b), and (c) is the same.

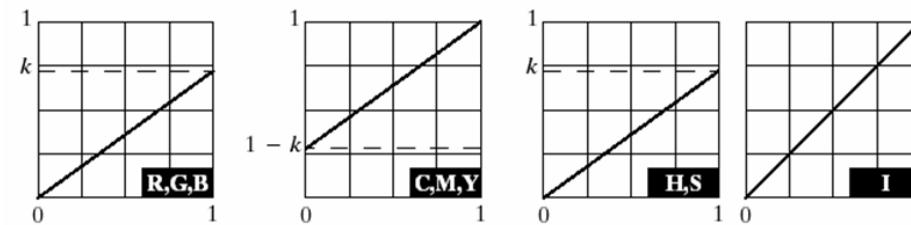
Important: The eqn. (5.33) to (5.35) depend only on one component within its color space.

Fig. 5.38 visualizes the example.



(a)

(b)



(c)

Fig. 5.38: Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30%. (c) Required RGB, CMY, and HSI transformation. (From [GW18])

5.4.2 Color Complements

The hues directly opposite one another on the **color circle**⁴ (see fig. 5.39) are called **color complements**.

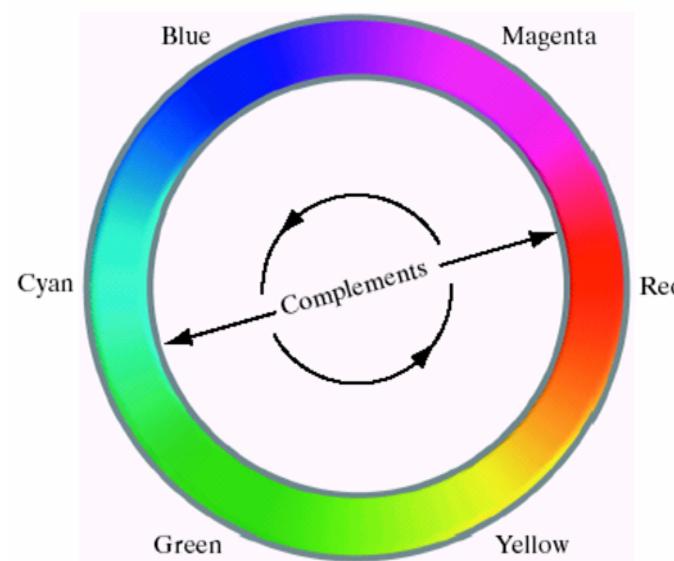


Fig. 5.39: Complements on a color circle. (From [GW18])

⁴Isaac Newton had been the first to join the ends of the color spectrum to form the color circle.

Example 5.6

Color Complement Transformations

Fig. 5.40(a) and (c) show the image in fig. 5.37(a) and its color complement.

The RGB transformations utilized are shown in fig. 5.40(b).

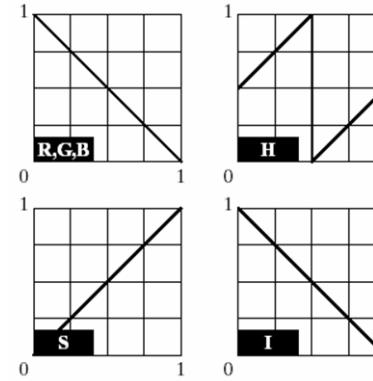
Note: The transformations are identical to the gray-scale negative transformation in section 3.2.

The computation of the HSI complements is more demanding:

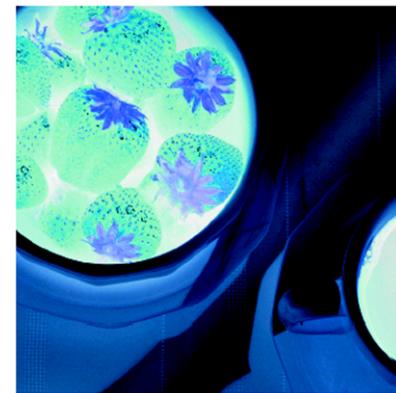
- The used transformations are depicted in fig. 5.40(b).
- The approximation of RGB component using the HSI transformations can be found in fig. 5.40(d).



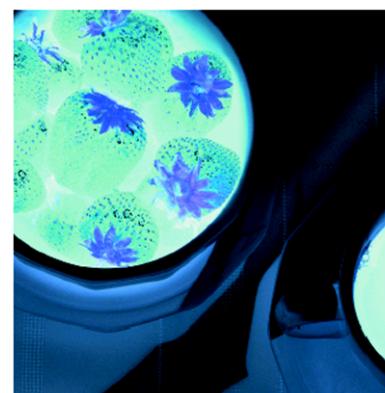
(a)



(b)



(c)



(d)

Fig. 5.40: Color complement transformation. (a) Original image. (b) Complement transformation function. (c) Complement of (a) based on the RGB mapping functions. (d) An approximation of the RGB complement using HSI transformations. (From [GW18])

5.4.3 Color Slicing

A specific range of colors in an image shall be highlighted to image segmentation purposes.

- Ideas:**
- (1) Display the colors of interest, so that they stand out from the background.
 - (2) Use the region defined by the colors as a mask for further processing.

Approach: Extension of the gray-level slicing techniques in sec. 3.2.4.

Note: The required transformations are more complex than those before.

Each pixel's transformed color components have to be functions of all n original pixel's color components.

Simple way to “slice” a color image:

- Map all colors outside some range of interest to a nonprominent neutral color (e. g. ‘middle gray’ (0.5, 0.5, 0.5)).
- (Hyper-)Cube to enclose the interesting colors:

$$S_i = \begin{cases} 0.5 & \text{if } [(r_j - a_j) > \frac{W}{2}] \text{ any } 1 \leq j \leq n \\ r_i & \text{otherwise} \end{cases} \quad (5.36)$$

where

for $i = 1, 2, \dots, n$

W : Width of cube

$(a_1, a_2, a_3, \dots, a_n)$: Center of the cube

- Sphere to specify the interesting colors:

$$S_i = \begin{cases} 0.5 & \text{if } \sum_{j=1}^n (r_j - a_j)^2 > R_0^2 \\ r_i & \text{otherwise} \end{cases} \quad (5.37)$$

where

for $i = 1, 2, \dots, n$

R_0 : Radius of enclosing sphere

$(a_1, a_2, a_3, \dots, a_n)$: Components of its center (i. e. prototypical color)

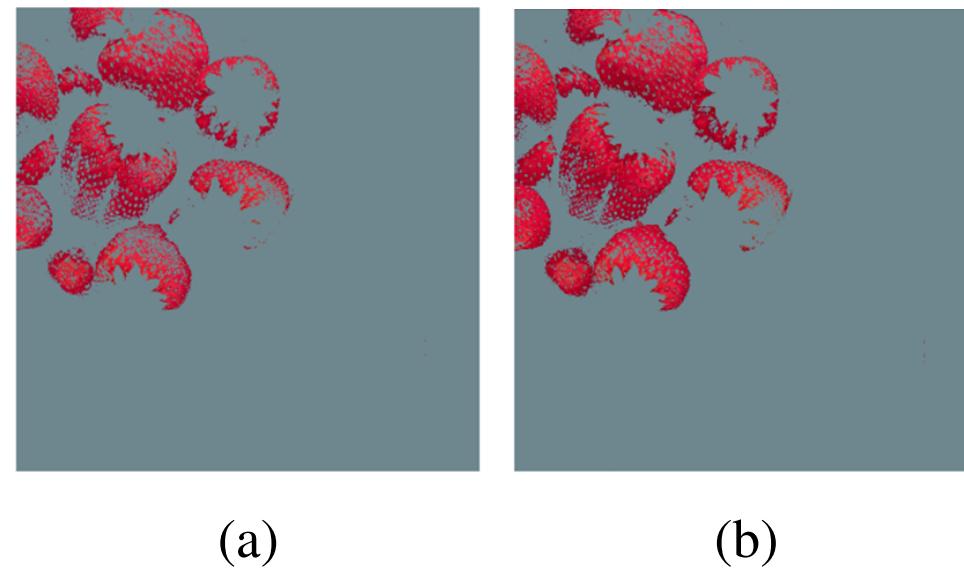
Example 5.7 Color Slicing

The edible part of the strawberries in fig. 5.37(a) can be separated by color slicing.

- Assumption:**
- Prototypical color: $(0.6863, 0.1608, 0.1922)$
 - $W = 0.2549$
 - $R_0 = 0.1765$

The results for the use of a cube and sphere are depicted in fig. 5.41.

Interpretation: The results for the sphere-based transformation are slightly better.



(a)

(b)

Fig. 5.41: Color slicing transformation that detects (a) reds within an RGB cube of width $W = 0.2549$ centered at $(0.6863, 0.1608, 0.1922)$, and (b) reds within an RGB sphere of radius 0.1765 centered at the same point. Pixels outside the cube and sphere were replaced by color $(0.5, 0.5, 0.5)$. (From [GW18])

5.4.4 Tone and Color Corrections

Color transformations nowadays lie in the scope of every pc user.

Important: The effectiveness of any transformation is judged ultimately in print.

While the transformations are developed, refined, and evaluated on monitors, it is necessary to maintain a high degree of **color consistency** between the monitors used and the eventual output devices.

Best results can be accomplished with a **device independent color model** that relates the color gamuts (see sec. 5.1, fig. 5.9).

Furthermore:

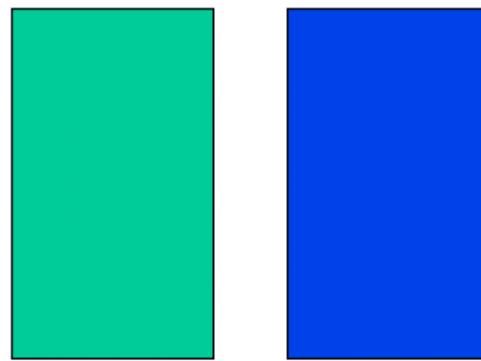
Neither the 3D color space produced by displaying the tristimulus X, Y, Z values in cartesian coordinates (see eq. (5.1)) nor the 2D (x, y) chromaticity diagram (see eq. (5.7)) are

- visually uniform
 - Equal distances in these spaces do not represent equally perceptible differences between color stimuli. (see McAdams ellipses in fig. 5.42) and
- colorimetric
 - Colors perceived as matching are encoded identically.

In 1976:

CIE introduced a new color space (**CIELAB** or CIE $L^*a^*b^*$) whose coordinates are non-linear functions of X, Y , and Z .

The color space includes all perceivable colors. It exceeds e.g. the RGB and the CMYK color spaces.



Test patches

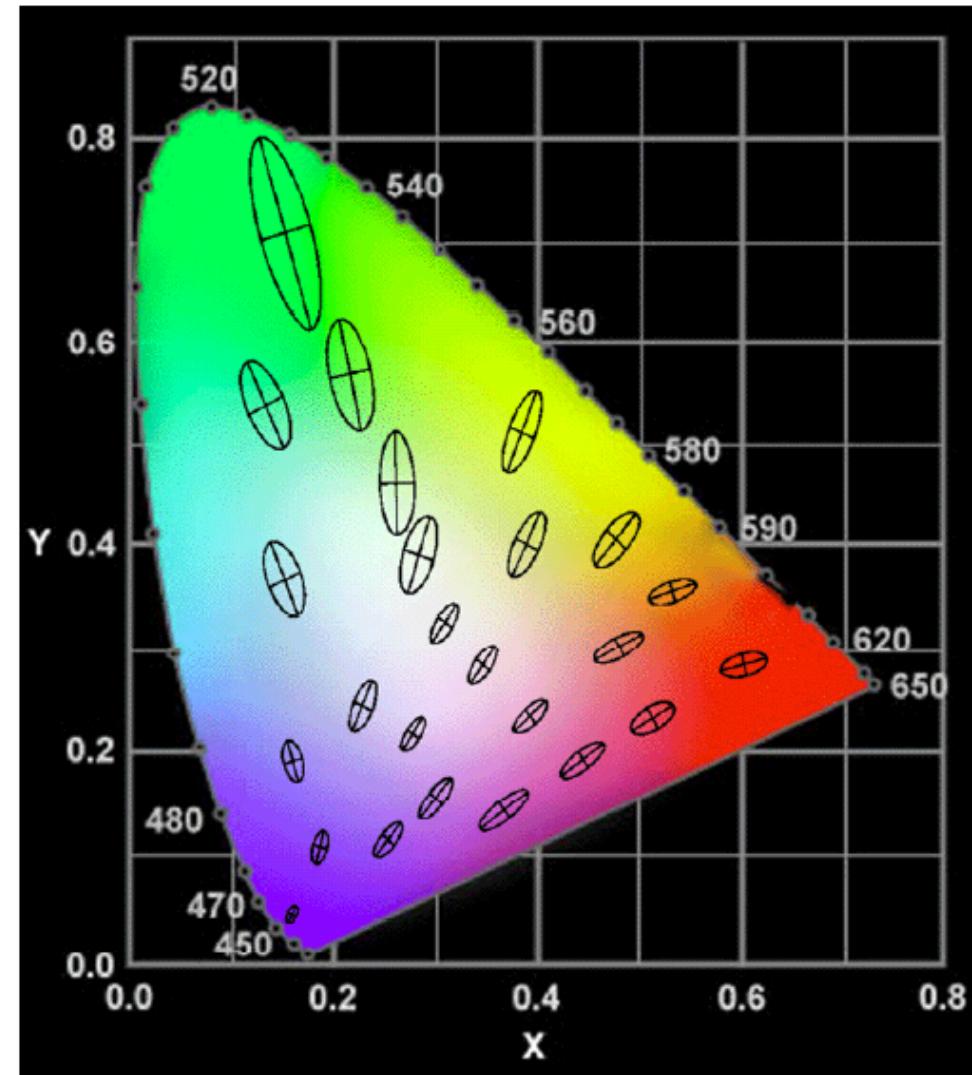


Fig. 5.42: McAdams ellipses. (From [http://graphics.ethz.ch/teaching/gdv1/GDV1_WS04/lecture/02_colors_1.pdf])

It is a color-opponent color space with the dimensions:

- L^* : lightness
where $L = 0$: black and $L = 100$: white
and
- a^* and b^* : color-opponent dimensions
 - a^* : red-green axis (negative values: green, positive one: red)
 - b^* : yellow-blue axis (negative values: blue, positive one: yellow)

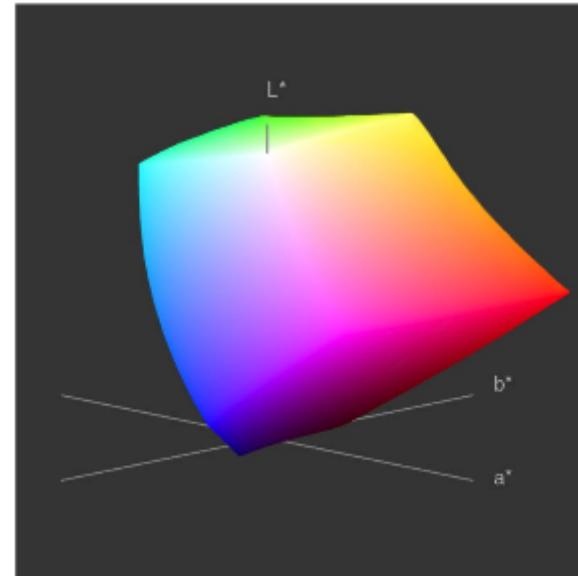


Fig. 5.43: Visualization of the CIELab color space. (From [Hof])

CIE $L^*a^*b^*$ model:

$$L^* = 116 h \left(\frac{Y}{Y_w} \right) - 16 \quad (5.38)$$

$$a^* = 500 \left[h \left(\frac{X}{X_w} \right) - h \left(\frac{Y}{Y_w} \right) \right] \quad (5.39)$$

$$b^* = 200 \left[h \left(\frac{Y}{Y_w} \right) - h \left(\frac{Z}{Z_w} \right) \right] \quad (5.40)$$

where

$$h(q) = \begin{cases} \sqrt[3]{q}, & q > 0,008856 \\ 7,787q + \frac{16}{116}, & q \leq 0,008856 \end{cases} \quad (5.41)$$

X_w, Y_w, Z_w : Reference white tristimulus values

(typically white of a perfectly reflecting diffuser under CIE standard D65 illumination, defined by $x=0.3127$ and $y=0.3290$ in the CIE chromaticity diagram)

- Interpretation:**
- Not directly displayable format.
 - Gamut encompasses entire visible spectrum.
 - Excellent decoupling of intensity (represented by lightness (L^*) and color (a^*b^*)).

Applications of the CIE $L^*a^*b^*$ color space:

- Image manipulation (tone and contrast editing),
- Image compression.

Principal benefit of a calibrated imaging systems:

→ They allow tonal and color imbalances to be corrected interactively and independently (see examples 5.8 and 5.9).

Example 5.8 Tonal Transformations

Idea: Adjustment of an image's brightness and contrast to provide maximum detail over a suitable range of intensities, while the colors are not changed.

In the RGB or CMY(K) spaces:

→ mapping of all color components with same transformation function.

Fig. 5.44 shows typical transformations used for correcting three common tonal imbalances
- flat, light, and dark images.

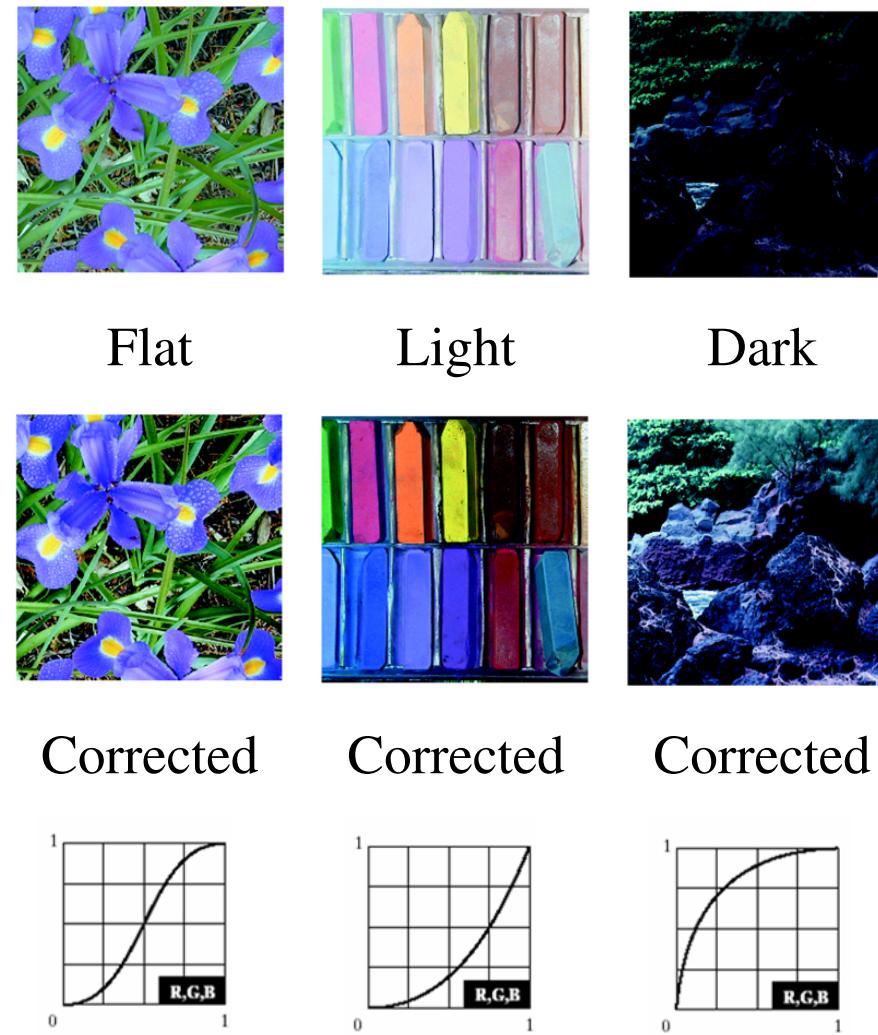


Fig. 5.44: Tonal correction for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues. (From [GW18])

Example 5.9 Color balancing corrections for CMYK color images

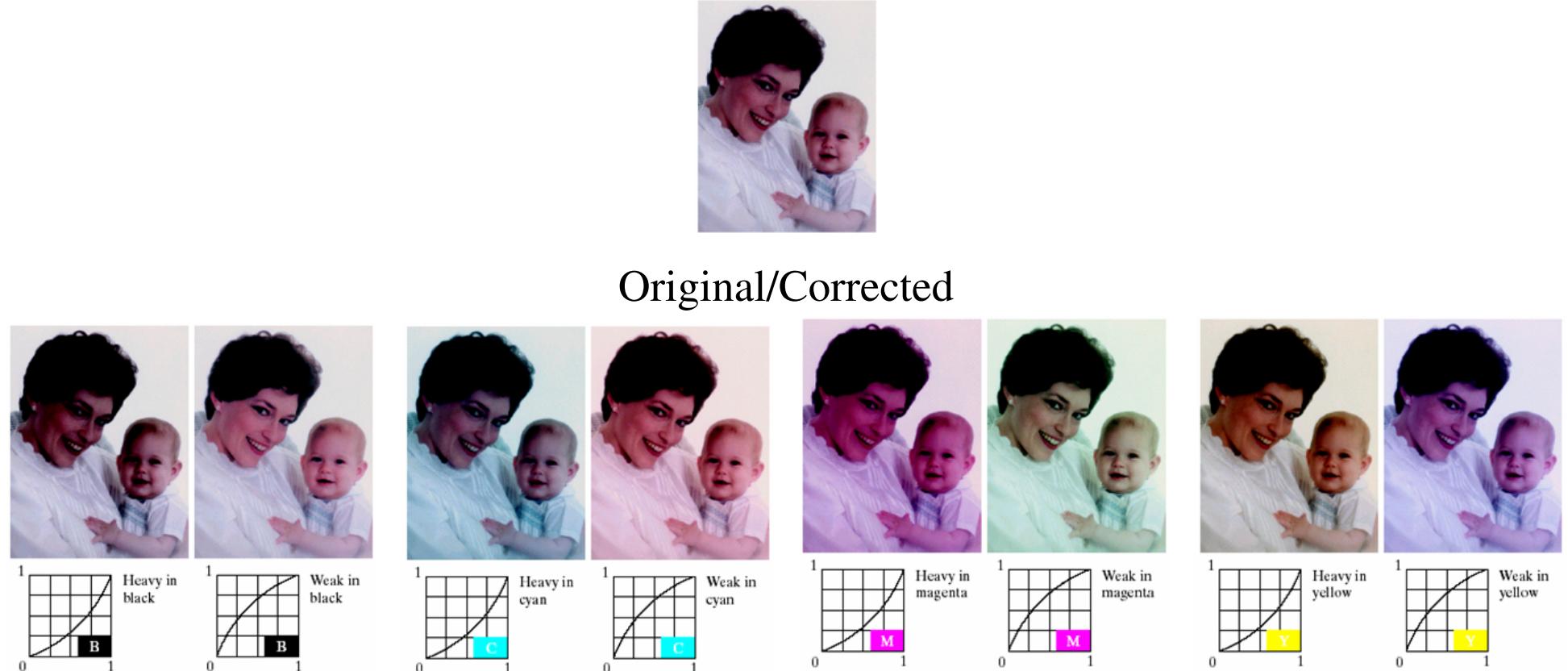


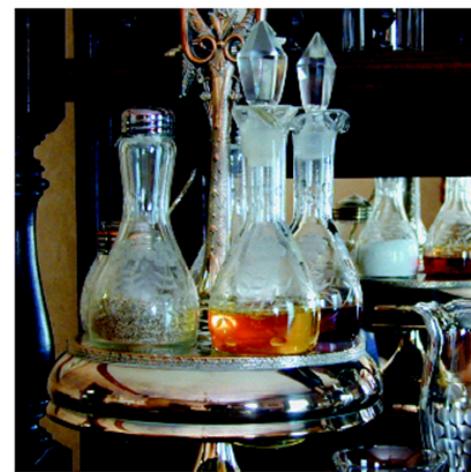
Fig. 5.45: Color balancing corrections (From [GW18])

5.4.5 Histogram Processing

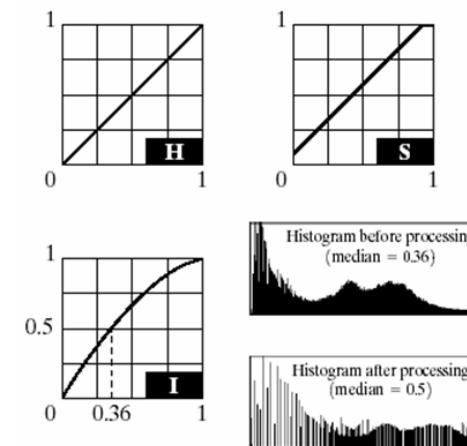
Idea: Histogram balancing (recall section 3.3.1) automatically determines a transformation that seeks to produce an image with an uniform histogram of intensity values.

The HSI color space is ideally suited for this type of approach.

Fig. 5.46 visualizes an example.



(a)



(b)



(c)



(d)

Fig. 5.46: Histogram equalization (followed by saturation adjustment). (From [GW18])

5.5 Smoothing and Sharpening

5.5.1 Color Image Smoothing

With reference to fig. 3.38 and the discussion in section 3.5, gray-level image smoothing can be viewed as a spatial filtering operation in which the coefficients of the filtering mask are all 1's.

The average of the RGB component vectors in a neighborhood, centered at (x, y) is:

$$\vec{\bar{\mathbf{c}}}(x, y) = \frac{1}{K} \sum_{x, y \in S} \vec{\mathbf{c}}(x, y) \quad (5.42)$$

It follows from eq. (5.29):

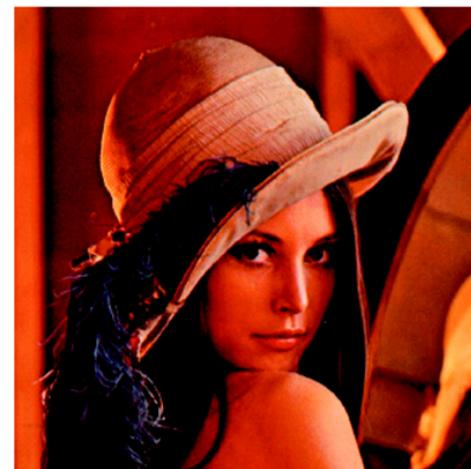
$$\vec{\bar{\mathbf{c}}}(x, y) = \begin{bmatrix} \frac{1}{K} \sum_{x, y \in S} R(x, y) \\ \frac{1}{K} \sum_{x, y \in S} G(x, y) \\ \frac{1}{K} \sum_{x, y \in S} B(x, y) \end{bmatrix} \quad (5.43)$$

Example 5.10 Smoothing of the Test image Lena

Fig. 5.47 and 5.48 show the components of the test image in RGB and HSI color space, respectively.

Fig. 5.49 compares both of the color spaces regarding smoothing.

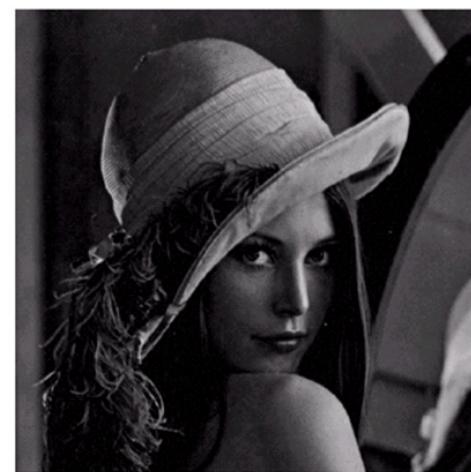
*



(a)



(b)



(c)



(d)

Fig. 5.47: (a) RGB image. (b) Red component image. (c) Green component image. (d) Blue component image. (From [GW18])

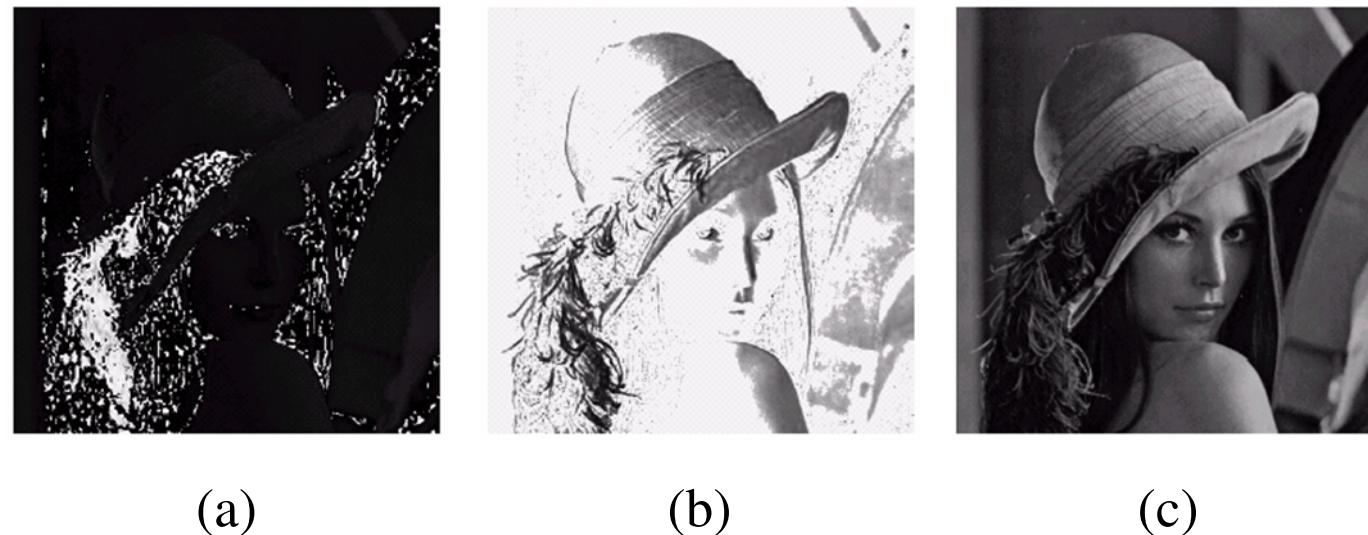


Fig. 5.48: HSI components of the RGB color image in Fig 5.47. (a) Hue. (b) Saturation. (c) Intensity. (From [GW18])



Fig. 5.49: Image smoothing with a 5×5 averaging mask. (a) Result of processing each RGB component image. (b) Result of processing the intensity component of the HSI image and converting to RGB. (c) Difference between the two results. (From [GW18])

- Interpretation:**
- In HSI color space it is sufficient to perform the smoothing in the intensity channel.
 - The results in fig. 5.49a und b are not identical:
 - the average of two pixels of differing color is a mixture of two colors, not either of the original colors.
 - Smoothing in the HSI color space maintains the original color (hue and saturation)

Note: The differences would increase as the size of the smoothing marks increases.

5.5.2 Color Image Sharpening

Use of the Laplacian (see section 3.6.2)

$$\nabla^2[\vec{\mathbf{c}}(x, y)] = \begin{bmatrix} \nabla^2 R(x, y) \\ \nabla^2 G(x, y) \\ \nabla^2 B(x, y) \end{bmatrix} \quad (5.44)$$

Interpretation: The Laplacian of a full-color image can be computed by computing the Laplacian of each component separately.

Example 5.11 Sharpening of a test image

Fig. 5.50(a) shows the result for the RGB color space.

The Laplacians of the RGB components in fig. 5.47 are combined to produce the sharpened full-color result.

Fig. 5.50(b) shows the sharpening of the intensity component in fig. 5.48(c).

Fig. 5.50(c) summarizes again the differences of the operation in both color spaces.

*



Fig. 5.50: Image sharpening with the Laplacian. (a) Result of processing each RGB channel. (b) Result of processing the intensity component and converting to RGB. (c) Difference between the two results. (From [GW18])

5.6 Color Segmentation

Short look on color segmentation for the sake of continuity.

Segmentation: Process that partitions an image into regions.

5.6.1 Segmentation in HSI Color Space

Segmentation based on color

→HSI color space (color is conveniently represented in the hue channel)

The saturation channel can be used to mark parts of an image in order to isolate regions of interest in the hue channel.

Example 5.12 Image Segmentation in the HSI Space

Task: Segmentation of reddish region in the lower left of fig. 5.51 (a).

Note: Image generated by pseudocolor methods, but it can be segmented as a full-color image without loss of generality.

Fig. 5.51 (b)-(d) show the HSI component images.

Note: Comparing of fig. 5.51 (a) and (b) yields:

Region of interest has high intensity values of hue

→ Colors on the blue-magenta side of red (compare fig. 5.20).

Idea: Building a mask from the saturation component by introducing a threshold equal to 10% of the maximum value (see fig. 5.51 (e)).

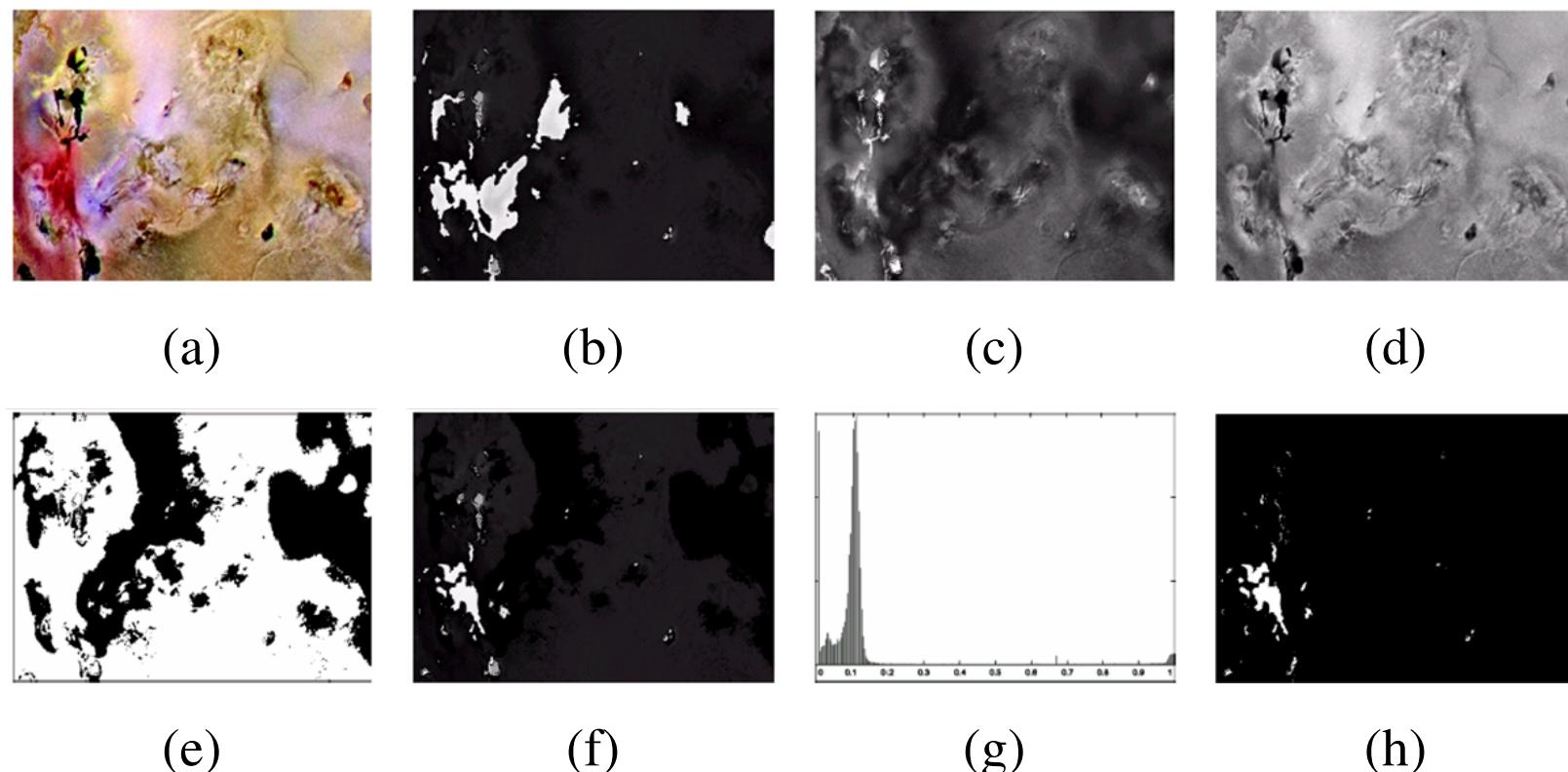


Fig. 5.51: Image segmentation in HSI space. (a) Original. (b) Hue. (c) Saturation. (d) Intensity. (e) Binary saturation mask (black = 0). (f) Product of (b) and (e). (g) Histogram of (f). (h) Segmentation of red components in (a). (From [GW18])

Fig. 5.51 (f) shows product of the mask with the hue image.

Analyzing of the histogram of (f) (see fig. 5.51 (g)) shows grouping at the very high of the gray scale (near 1.0).

Applying a threshold of 0.9 to the product image results in the binary image shown in fig. 5.51 (h).

Spatial localization of white parts of (h) identifies reddish parts of the original image.

Interpretation: Results shown are about the best this method can do in identifying the region of interest in the original image.



5.6.2 Segmentation in RGB Vector Space

Note: Though the HSI space is more intuitive, segmentation is a field in which better results generally are obtained by using RGB color vectors.

A colored region of interest can be described by an estimate of an "average" color vector \vec{a} .

Objective: Classify each RGB pixel in the entire image as having a color in the specified range or not.

Assume: \vec{z} denotes an arbitrary point in the RGB space.

Idea: Measurement of similarity of \vec{z} and \vec{a}
→ Distance between them has then a threshold D_0 .

Easiest approach: Use of Euclidean distance

Several formulations possible:

- Sphere:

$$\begin{aligned}
 D(\vec{\mathbf{z}}, \vec{\mathbf{a}}) &= \| \vec{\mathbf{z}} - \vec{\mathbf{a}} \| \\
 &= [(\vec{\mathbf{z}} - \vec{\mathbf{a}})^T (\vec{\mathbf{z}} - \vec{\mathbf{a}})]^{\frac{1}{2}} \\
 &= [(z_R - a_R)^2 + (z_G - a_G)^2 + (z_B - a_B)^2]^{\frac{1}{2}}
 \end{aligned} \tag{5.45}$$

(see fig. 5.52(a))

- Elliptical body:

Useful generalization: Distance measure of the form

$$D(\vec{\mathbf{z}}, \vec{\mathbf{a}}) = [(\vec{\mathbf{z}} - \vec{\mathbf{a}})^T C^{-1} (\vec{\mathbf{z}} - \vec{\mathbf{a}})]^{\frac{1}{2}} \tag{5.46}$$

where **C: Covariance matrix**⁵

⁵ Computation of covariance matrix: see (From [GW18])

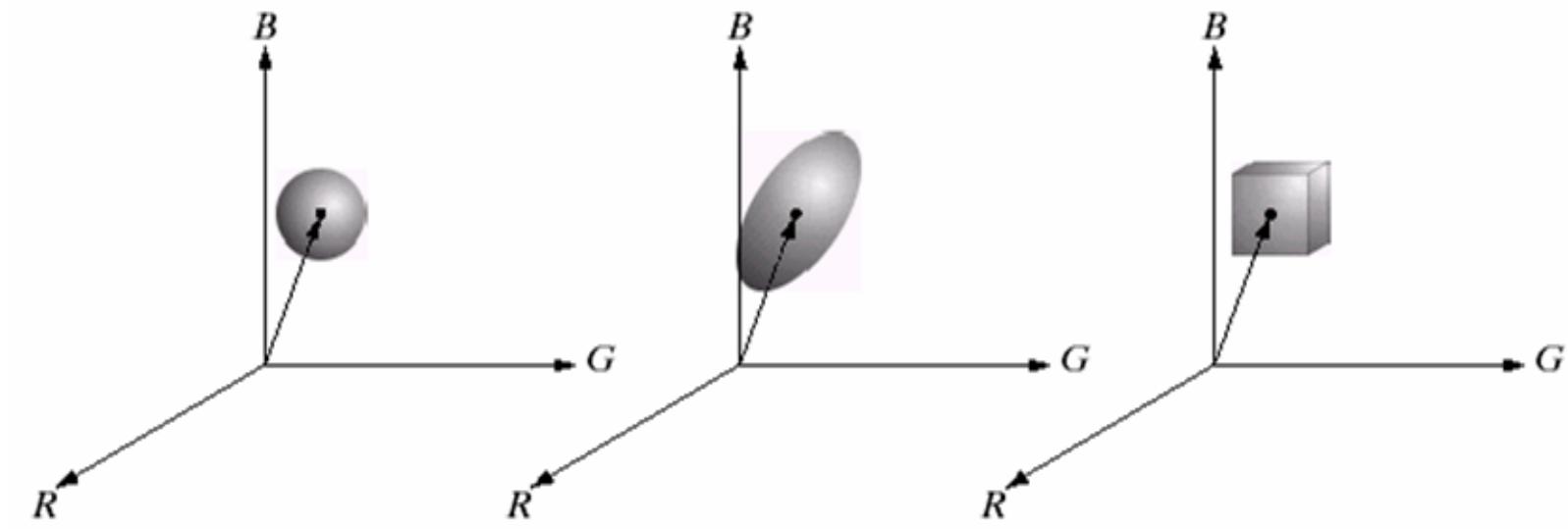


Fig. 5.52: Three approaches for enclosing data regions for RGB vector segmentation. (From [GW18])

Important property: Principal axes are oriented in the direction of maximum data spread.

Note: If $C = I$: Equation (5.46) reduces to equation (5.45).
(see fig. 5.52 (b))

- Cube:

To avoid root computations the distance squared can be used (the distances are positive and monotonic).

Computations in equations (5.45) and (5.46) remain computationally expensive
→ Use of a bounding box on \vec{a} and its dimensions along each of the color axes chosen proportional to the standard deviations of the samples along each of the axis.

Note: Computation of the standard deviations is done only once using sample color data.

Example 5.13 Image Segmentation in the RGB Space

Reconsideration of task in example 5.12:

1. Computation of average vector \vec{a} using the pixels with the rectangle in fig. 5.53 (a)
2. Computation of the standard deviations of the RGB values of those samples.
3. Centering of box at \vec{a}
4. Selection of dimensions of box:
1.25 times the standard deviations
 $\rightarrow \sigma_R, \sigma_G, \sigma_B$
 \rightarrow dimension of the box along the R axis extends from $(a_r - 1.25\sigma_R)$ to $(a_r + 1.25\sigma_R)$
segmentation of the next image

The results are shown in fig. 5.53 (b).

Note: Segmented region extends the rectangle of step 1.

Comparing of fig. 5.53 (b) with 5.51 (h) shows that segmentation in the RGB space is more accurate.

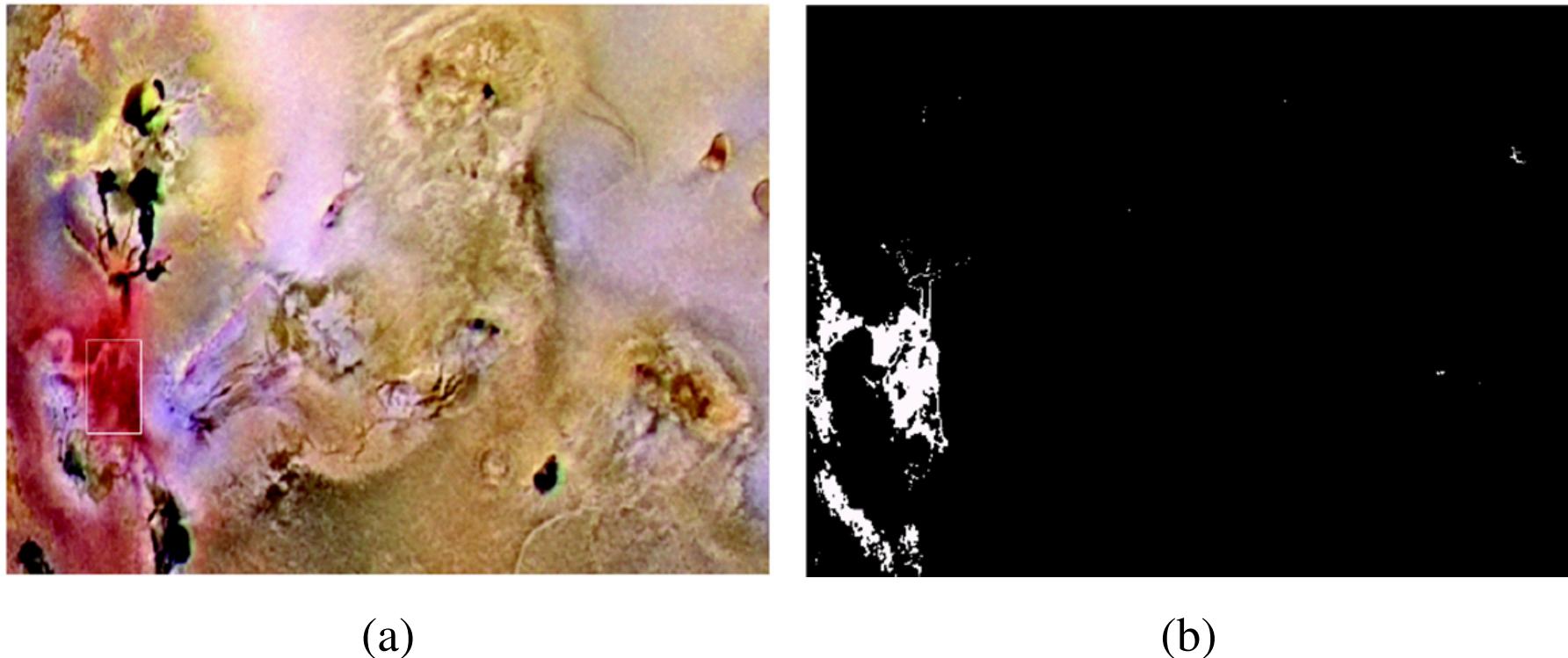


Fig. 5.53: Segmentation in RGB space. (a) Original image with colors of interest shown enclosed by a rectangle. (b) Result of segmentation in RGB vector space. Compare with Fig. 5.51(h) (From [GW18])

5.6.3 Color Edge Detection

In section 3.6.3:

- edge detection by gradient operators, but
- the gradient was not defined for vector quantities.

Example 5.14 Edge Detection

Given: $2 M \times M$ color images in fig. 5.54 (M odd)

Consideration of the gradient at point $(\frac{M+1}{2}, \frac{M+1}{2})$

→ Value is the same in both cases.

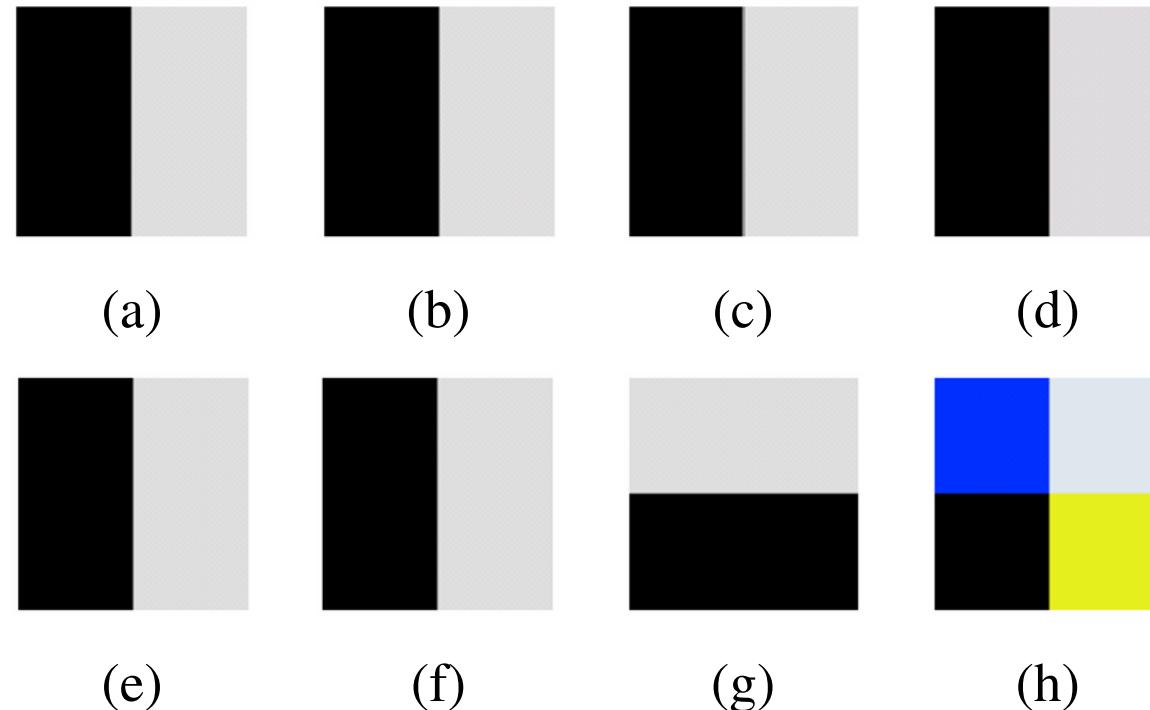


Fig. 5.54: (a)-(c) R, G, and B component image and (d) resulting RGB color image. (f)-(g) R, G, and B component image and (h) resulting RGB color image. (From [GW18])

Interpretation: The processing of three individual planes to form the composite gradient image can yield erroneous results.

Extension the concept of a gradient to vector functions⁶:

Assume: Three unit vectors along the R, G, and B axis of RGB color space: \vec{e}_R , \vec{e}_G , \vec{e}_B

Define:

$$\vec{u} = \frac{\partial R}{\partial x} \vec{e}_R + \frac{\partial G}{\partial x} \vec{e}_G + \frac{\partial B}{\partial x} \vec{e}_B \quad (5.47)$$

$$\vec{v} = \frac{\partial R}{\partial y} \vec{e}_R + \frac{\partial G}{\partial y} \vec{e}_G + \frac{\partial B}{\partial y} \vec{e}_B \quad (5.48)$$

⁶Method presented here is one of several possibilities. This was proposed by Di Zenzo (1986).

Compute: Dot products of the new vectors:

$$g_{xx} = \vec{\mathbf{u}} \cdot \vec{\mathbf{u}} = \vec{\mathbf{u}}^T \vec{\mathbf{u}} = \left| \frac{\partial R}{\partial x} \right|^2 + \left| \frac{\partial G}{\partial x} \right|^2 + \left| \frac{\partial B}{\partial x} \right|^2 \quad (5.49)$$

$$g_{yy} = \vec{\mathbf{v}} \cdot \vec{\mathbf{v}} = \vec{\mathbf{v}}^T \vec{\mathbf{v}} = \left| \frac{\partial R}{\partial y} \right|^2 + \left| \frac{\partial G}{\partial y} \right|^2 + \left| \frac{\partial B}{\partial y} \right|^2 \quad (5.50)$$

$$g_{xy} = \vec{\mathbf{u}} \cdot \vec{\mathbf{v}} = \vec{\mathbf{u}}^T \vec{\mathbf{v}} = \frac{\partial R}{\partial x} \frac{\partial R}{\partial y} + \frac{\partial G}{\partial x} \frac{\partial G}{\partial y} + \frac{\partial B}{\partial x} \frac{\partial B}{\partial y} \quad (5.51)$$

It can be shown that the direction of maximum rate of change of $\vec{\mathbf{c}}(x, y)$ is given by the angle:

$$\Theta = \frac{1}{2} \arctan \left(\frac{2g_{xy}}{g_{xx} - g_{yy}} \right) \quad (5.52)$$

And value of the rate of change at point (x, y) is

$$F(\Theta) = \sqrt{\frac{1}{2}[(g_{xx} + g_{yy}) + (g_{xx} - g_{yy}) \cos(2\Theta) + 2 g_{xy} \sin(2\Theta)]} \quad (5.53)$$

It holds:

$$\tan \alpha = \tan(\alpha \pm \pi)$$

If Θ_0 is a solution to eq. (5.52), than so is $\Theta_0 \pm \frac{\pi}{2}$.

Furthermore: $F(\Theta) = F(\Theta + \pi)$

→ Computation of F only for value of Θ in the interval $[0, \pi]$.

Interpretation: Eq. (5.52) provides two values 90° apart.

→ Equation associates with each point (x, y) a pair of orthogonal directions (F_{max} and F_{min}).⁷

Hint: Partial derivatives for implementing g_{xx} , g_{yy} , and g_{xy} can be computed e. g. using the Sobel operators (see 3.6.3).

⁷For details of computation please refer to paper of Di Zenzo.

In fig. 5.55 the results for the vector gradient approach is compared to the composite gradient approach in example 5.14.

- Interpretation:**
- Both approaches yield reasonable results.
 - Little more extra detail when using the vector-based approach.
 - Trade-off between computational burden and quality of results required.

Fig. 5.56 shows the three component gradient images which were used to obtain fig. 5.55(c) (after adding and scaling).

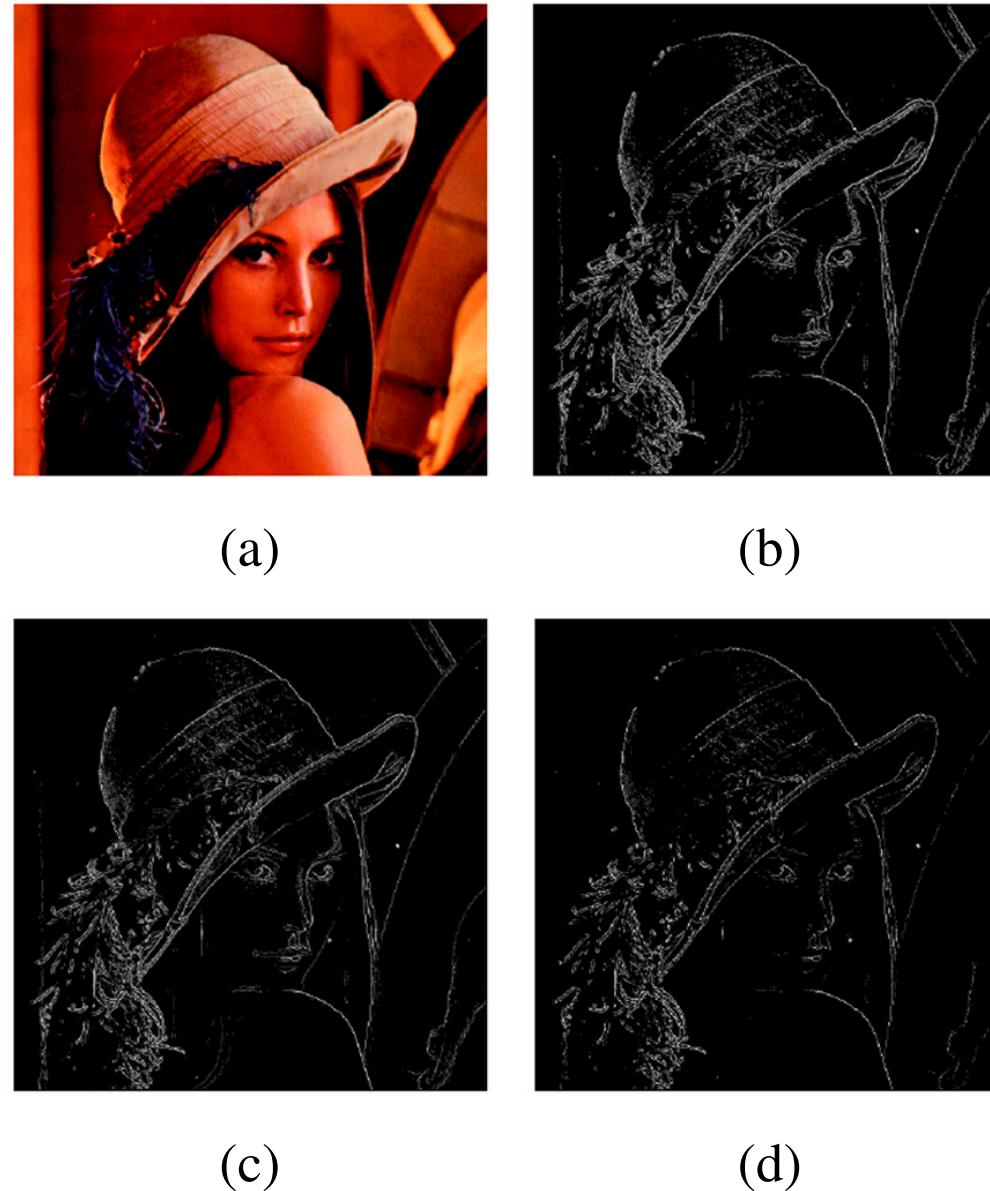


Fig. 5.55: (a) RGB image. (b) Gradient computed in RGB color vector space. (c) Gradients computed on a per-image basis and then added. (d) Difference between (b) and (c). (From [GW18])

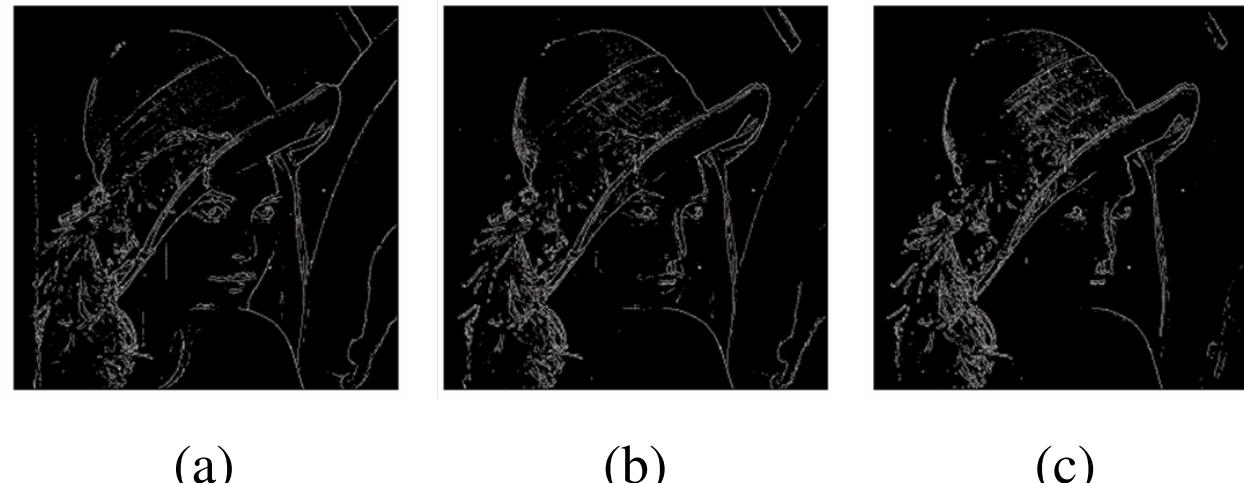


Fig. 5.56: Component gradient images of the color image. (a) Red component, (b) green component, and (c) blue component. These three images were added and scaled to produce the image. (From [GW18])

5.7 Color Image Compression

Note: Number of bits to represent a color image is three or four times greater than the number employed in the representation of gray levels.

→ Data compression is very important for storage and transmission of color images (see chapter 6).

Example 5.15 Color Image Compression

Fig. 5.57(a) shows a 24-bit RGB full color image.

Fig. 5.57(b) is reconstructed from a (lossy) compressed and subsequently decompressed approximation of it (JPEG 2000, compression rate: 230/1).



(a)



(b)

Fig. 5.57: Color image compression. (a) Original RGB image. (b) Result of compressing and decompressing the image in (a). (From [GW18])

Appendix

A.1 Noise

Noise: Random background events which accompany every real signal, but are not part of the ideal signal.

Wide range of sources:

- Variations in the detector sensitivity,
- Environmental variations,
- Discrete nature of radiation,
- Transmission errors,
- Quantization errors,
- etc.

Hint: Event irrelevant scene details (e. g. surface reflectance textures) can be treated as noise.

Two categories of noise:

- Independent noise
- Noise which is dependent on the image data.

Image independent noise can often be described by an additive noise model:

$$f(i, j) = s(i, j) + n(i, j) \quad (\text{A.1.1})$$

where

$f(i, j)$: recorded image

$s(i, j)$: true image

$n(i, j)$: noise

Features of noise are:

- Mean (often: zero-mean), and
- Variance σ_n^2

The impact of the noise on the image can be described by the **signal to noise ratio (SNR)**:

$$\begin{aligned} SNR &= \frac{\sigma_s}{\sigma_n} \\ &= \sqrt{\frac{\sigma_f^2}{\sigma_n^2} - 1} \end{aligned} \tag{A.1.2}$$

where

σ_f : variance of recorded image

σ_s : variance of true image

In many cases:

Additive noise is evenly distributed of the frequency domain (i. e. **white noise**), whereas an image contains mainly low frequency information.

Additive Stationary Gaussian Noise:

Each pixel has an additive noise value which is chosen independently from the same Gaussian probability distribution.

The standard deviation is a parameter of the model, which is intended to describe thermal noise in cameras (see fig. A.1.1).

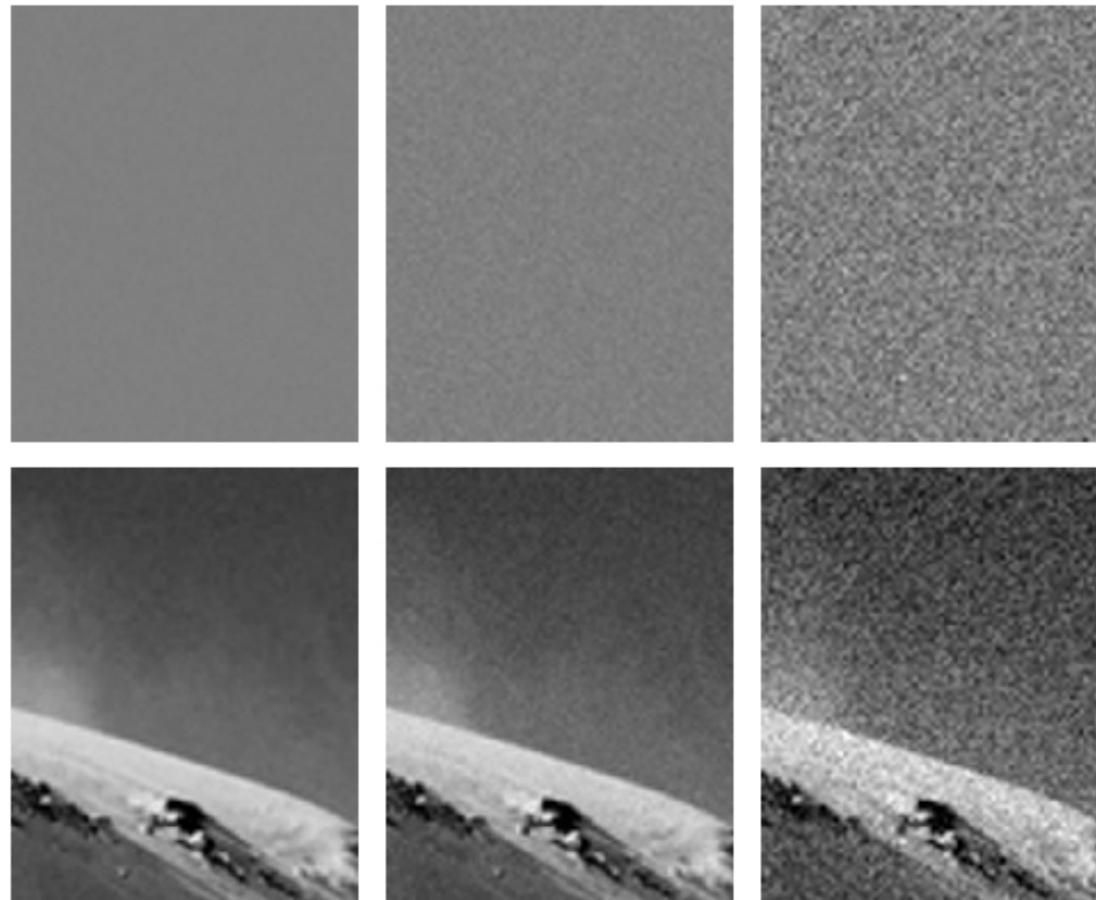


Fig. A.1.1: Stationary additive Gaussian noise. (a)-(c) Three realizations of the noise process. Images have half the range of brightness to show both negative and positive values of noise. From left to right: the noise has the standard deviation $\frac{1}{256}$, $\frac{4}{256}$, $\frac{16}{256}$ of the full range of brightness. This corresponds roughly to bits zero, two, and five of a camera with $k = 8\text{ bit}$. (d)-(f) shows this noise added to an image. In each case values below zero or above full range have been adjusted to zero or the maximum value accordingly. (From [FP11])

Noise:

- Occurs in all recorded images to a certain extent.
- Due to discrete nature of radiation ("counting photons").
- Under some assumptions, this noise can be modeled with an independent, additive model, where $n(i, y)$ has a zero-mean Gaussian distribution described by its standard deviation σ , or variance σ^2 .

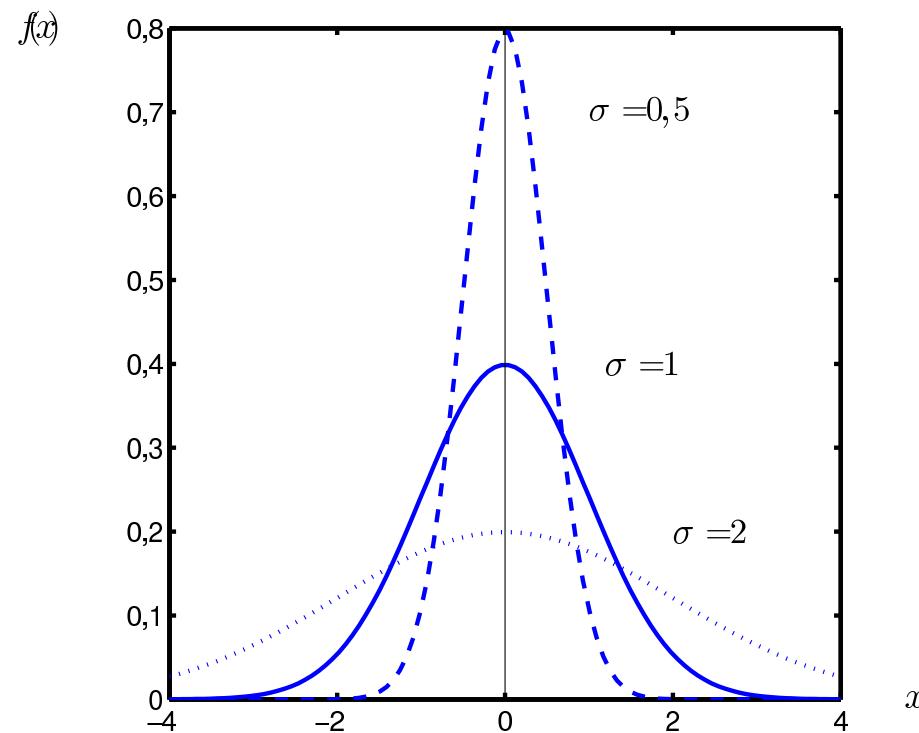


Fig. A.1.2: 1D Gaussian distributions with mean 0 and standard deviations 0,5, 1, and 2.

The Gaussian distribution can be expressed by

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{(-\frac{1}{2} \times \frac{x^2}{\sigma^2})} \quad (\text{A.1.3})$$

Hint: It is wrong to assume that noise does not contain information; e. g. it is possible to estimate the camera temperature by taking pictures in a dark room with the lens cap on.

Salt-and-Pepper Noise:

- Also called intensity spikes or **speckle**.
- This noise is caused by errors in the data transmission:
 - Corrupted pixels are either set to the maximum value ("white") or have single bits flipped over.
 - In rare cases, pixels are set alternatively to zero or to the maximum value (giving the image the salt-and-pepper appearance).
 - Unaffected pixels remain unchanged.

Data Dependent Noise:

- E. g. arising when monochromatic radiation is scattered from a surface whose roughness is of the order of a wavelength, causing wave interference which results in image **speckle**.
- Can be modeled with a multiplicative, or non-linear model.

A.2 Theorems of 2D Fourier Transform

Property	Expression(s)
Fourier transform	$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j 2\pi(ux/M + vy/N)}$
Inverse Fourier transform	$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j 2\pi(ux/M + vy/N)}$
Polar representation	$F(u, v) = F(u, v) e^{-j\phi(u, v)}$
Spectrum	$ F(u, v) = [R^2(u, v) + I^2(u, v)]^{1/2}, R=\text{Real}(F) \text{ and } I=\text{Imag}(F)$
Phase angle	$\phi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right]$
Power spectrum	$P(u, v) = F(u, v) ^2$
Average value	$\bar{f}(x, y) = F(0, 0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$
Translation	$f(x, y) e^{j 2\pi(u_0 x/M + v_0 y/N)} \circledcirc \bullet F(u - u_0, v - v_0)$ $f(x - x_0, y - y_0) \circledcirc \bullet F(u, v) e^{-j 2\pi(ux_0/M + vy_0/N)}$ <p>When $x_0 = u_0 = M/2$ and $y_0 = v_0 = N/2$, then</p> $f(x, y) (-1)^{x+y} \circledcirc \bullet F(u - M/2, v - N/2)$ $f(x - M/2, y - N/2) \circledcirc \bullet F(u, v) (-1)^{u+v}$ $f(x, y) e^{j 2\pi(u_0 x/M + v_0 y/N)} \circledcirc \bullet F(u - u_0, v - v_0)$ $f(x - x_0, y - y_0) \circledcirc \bullet F(u, v) e^{-j 2\pi(ux_0/M + vy_0/N)}$

Table A.1: Summary of some important properties of the 2-D Fourier transform. (From [GW18])

Property	Expression(s)
Conjugate symmetry	$F(u, v) = F^*(-u, -v)$ $ F(u, v) = F(-u, -v) $
Differentiation	$\frac{\partial^n f(x, y)}{\partial x^n} \circ \bullet (ju)^n F(u, v)$ $(-jx)^n f(x, y) \circ \bullet \frac{\partial^n F(u, v)}{\partial u^n}$
Laplacian	$\nabla^2 f(x, y) \circ \bullet -(u^2 + v^2) F(u, v)$
Distributivity	$\mathcal{F}\{f_1(x, y) + f_2(x, y)\} = \mathcal{F}\{f_1(x, y)\} + \mathcal{F}\{f_2(x, y)\}$ $\mathcal{F}\{f_1(x, y) \cdot f_2(x, y)\} \neq \mathcal{F}\{f_1(x, y)\} \cdot \mathcal{F}\{f_2(x, y)\}$
Scaling	$a f(x, y) \circ \bullet a F(u, v)$ $f(ax, by) \circ \bullet \frac{1}{ ab } F(u/a, v/b)$
Rotation	$x = r \cos\Theta, \quad y = r \sin\Theta, \quad u = \omega \cos\Phi, \quad v = \omega \sin\Phi$ $f(r, \Theta + \Theta_0) \circ \bullet F(\omega, \Phi + \Theta_0)$
Periodicity	$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N)$ $f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N)$

Table A.1: (continued). (From [GW18])

Property	Expression(s)
Separability	See eqs. () and (). Separability implies that we can compute the 2D transform of an image by first computing 1D transforms along each row of the image, and then computing a 1D transform along each column of this intermediate result. The reverse, columns and then rows, yields the same result.
Computation of the inverse Fourier transform using a forward transform algorithm	$\frac{1}{MN}f^*(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v)e^{-j 2\pi(ux/M+vy/N)}$ <p>This equation indicates that inputting the function $F^*(u, v)$ into an algorithm designed to compute the forward transform (right side of the preceding equation) yields $f^*(x, y)/MN$. Taking the complex conjugate and multiplying this result by MN gives the desired inverse.</p>
Convolution	$f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n)$
Correlation	$f(x, y) \circ h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n)h(x + m, y + n)$
Convolution theorem	$f(x, y) * h(x, y) \circlearrowleft F(u, v)H(u, v)$ $f(x, y)h(x, y) \circlearrowleft F(u, v) * H(u, v)$
Correlation theorem	$f(x, y) \circ h(x, y) \circlearrowleft F^*(u, v)H(u, v)$ $f^*(x, y)h(x, y) \circlearrowleft F(u, v) \circ H(u, v)$

Table A.1: (continued). (From [GW18])

Property	Expression(s)
Some useful FT pairs:	
<i>Impulse</i>	$\delta(x, y) \circledcirc \bullet 1$
<i>Gaussian</i>	$A\sqrt{1\pi}\sigma e^{-2\pi^2\sigma^2(x^2+y^2)} \circledcirc \bullet Ae^{-(u^2+v^2)/2\sigma^2}$
<i>Rectangle</i>	$\text{rect}[a, b] \circledcirc \bullet ab \frac{\sin(\pi ua)\sin(\pi vb)}{(\pi ua)(\pi vb)} e^{-j\pi(ua+vb)}$
<i>Cosine</i>	$\cos(2\pi u_0 x + 2\pi v_0 y) \circledcirc \bullet \frac{1}{2}[\delta(u + u_0, v + v_0) + \delta(u - u_0, v - v_0)]$
<i>Sine</i>	$\sin(2\pi u_0 x + 2\pi v_0 y) \circledcirc \bullet j\frac{1}{2}[\delta(u + u_0, v + v_0) - \delta(u - u_0, v - v_0)]$

Table A.1: (continued). (From [GW18])

A.3 Excursion: Basics of Complex Numbers

Assumption: \mathbb{C} : field of complex numbers

- Definition:**
- $w \in \mathbb{C}$: n^{th} **root of unity** if $w^n = 1$.
 - $w \in \mathbb{C}$ meaning primitive n^{th} root of unity if $w^n = 1$, but $w^k \neq 1$ for all $k \in [1, \dots, n - 1]$.

Example A.16 Calculating Roots of Unity I

Given: $n = 4$

Wanted: All 4^{th} roots of unity

Solution: Primitive 4^{th} root of unity: j

All 4^{th} roots of unity: $j^0 = 1, \quad j^1 = j, \quad j^2 = -1, \quad j^3 = -j$

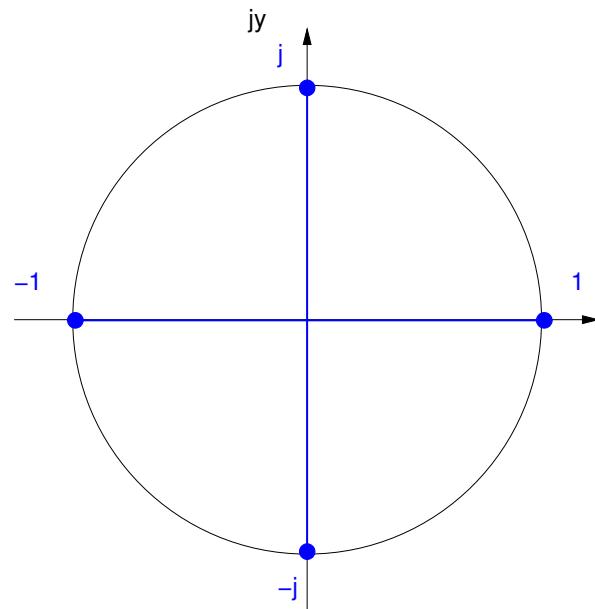


Fig. A.3.1: Unit circle in the complex plane with 4^{th} roots of unity.

Example A.17 Calculating Roots of Unity II

Given: $n = 6$

Wanted: All 6^{th} roots of unity

Solution: Primitive 6^{th} root of unity: $\cos\left(\frac{2\pi}{6}\right) + j \sin\left(\frac{2\pi}{6}\right)$

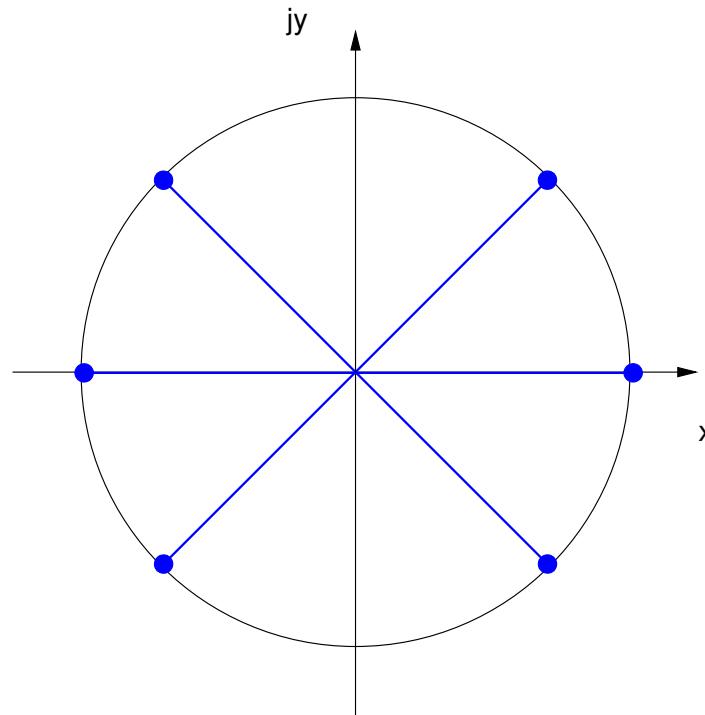


Fig. A.3.2: Unit circle in the complex plane with 6^{th} roots of unity.

Generally: n^{th} root of unity:

$$w = \cos\left(\frac{k 2\pi}{n}\right) + j\left(\sin\left(\frac{k 2\pi}{n}\right)\right) \quad (\text{A.3.1})$$

where

$w \in \mathbb{C}$ and

$$k \in \{0, \dots, n - 1\}$$

Attention: If k is **relatively prime** to n , then w is primitive.

$$= e^{\frac{j 2\pi k}{n}} \quad (\text{A.3.2})$$

Example A.18 Higher Roots of Unity

See [media/animated-gifs/einheitswurzeln/rootsu.gif].



Properties of n^{th} roots of unity:

1. There are exactly n -different n^{th} roots of unity exist in \mathbb{C} .

These can be represented as the powers of a primitive n^{th} root of unity w :

$$w^0, w^1, w^2, \dots, w^{n-1}$$

2. Every integer power of an n^{th} root of unity is again an n^{th} root of unity because the following applies:

$$(w^k)^n = w^{kn} = (w^n)^k = 1^k = 1$$

Hint: Property 2) is also valid for negative k s.

3. If n is an even number, the following holds true for every primitive root of unity:

$$w^{\frac{n}{2}} = -1$$

Proof: $(w^{\frac{n}{2}})^2 = w^n = 1$
 $\rightarrow w^{\frac{n}{2}}$ is the 2^{nd} root of unity (1 or -1).

Thus it applies that $w^{\frac{n}{2}} \neq 1$ because w is primitive.

Hence it follows that $w^{\frac{n}{2}} = -1$.

4. If n is an even number then the square w^2 of a primitive n^{th} root of unity is also a primitive $(\frac{n}{2})^{th}$ root of unity.

Proof: a) $(w^2)^{\frac{n}{2}} = w^n = 1$
 b) Assumption: w^2 is not primitive
 $\rightarrow \exists k \in \{1, \dots, \frac{n}{2} - 1\}$ with $(w^2)^k = 1$

Thus it follows that $w^{2k} = 1$ with $2k < n$:

Contradiction because w is primitive.

5. If w is a primitive n^{th} root of unity then w^{-1} is also a primitive n^{th} root of unity.

Proof: a) $(w^{-1})^n = w^{-n} = \frac{1}{w^n} = 1$

b) Assumption: w^{-1} is not primitive

$$\rightarrow \exists k \in \{1, \dots, n-1\} \text{ with } (w^{-1})^k = w^{-k} = 1$$

$$\text{Thus it follows that } w^k = \frac{1}{w^{-k}} = \frac{1}{1} = 1$$

\rightarrow Contradiction to w being primitive.

6. For $w \in \mathbb{C}$: For the conjugated number w^* of an n^{th} root of unity w holds: $w^* = w^{-1}$.

Proof:

$$\begin{aligned} w \cdot w^* &= \left[\cos\left(\frac{k2\pi}{n}\right) + j \sin\left(\frac{k2\pi}{n}\right) \right] \cdot \left[\cos\left(\frac{k2\pi}{n}\right) - j \sin\left(\frac{k2\pi}{n}\right) \right] \\ &= \cos\left(\frac{k2\pi}{n}\right)^2 + \sin\left(\frac{k2\pi}{n}\right)^2 \\ &= 1 \end{aligned}$$

Bibliography

- [Ado17] ADOBE. *Verwenden von Transparenz und Hintergrundfarben*. [<https://helpx.adobe.com/de/photoshop-elements/using/using-transparency-mattes.html>] (accessed: 17.10.2019). 2017
- [Agr] AGRELLA, Chance. *Comparison of file sizes*. [http://freerangestock.com/understanding/vector_bitmap/Part3_Bitmap_vs_Vector.html] (accessed: 01.10.2019)
- [(AL18)] (ALS), Advanced Light S. *Advanced Light Source - Quick Facts*. [<http://als.lbl.gov/>] (accessed: 17.10.2019). 2018
- [Bac19a] BACH, Michael. *Hidden figures: Dalmatian dog*. [http://www.michaelbach.de/ot/cog_dalmatian/index.html] (accessed: 17.10.2019). 2019
- [Bac19b] BACH, Michael. *Müller-Lyer illusion*. [http://www.michaelbach.de/ot/sze_muelue/] (accessed: 01.10.2019). 2019
- [Bac19c] BACH, Michael. *Poggendorff illusion*. [http://www.michaelbach.de/ot/ang_poggendorff/index.html] (accessed: 01.10.2019). 2019
- [But11] BUTZ, Tillmann: *Fouriertransformation für Fußgänger*. 7. edition. Vieweg, 2011. – ISBN 978–3–8348–0946–9

- [Cen] CENTER, Electronic Privacy I. *Terahertz imaging in the security domain: Body scanner.* [<http://epic.org/privacy/airtravel/backscatter/>] (accessed: 01.10.2019)
- [Cha15] CHANDLER, Daniel. *Visual Perception.* [<http://visual-memory.co.uk/daniel/Documents/visper/>] (accessed: 17.10.2019). 2015
- [DWD05] DWD, Deutscher W. *Multispectral images taken by METEOSAT (geostationary meteorological satellite).* [<http://www.dwd.de/>] (accessed: 25.04.2005). 2005
- [Esc60] ESCHER, Maurits C. *Ascending and Descending (Lithography).* [http://1.bp.blogspot.com/-FF26Xwu2-0Q/UayeWEABBvI/AAAAAAAWF8/FtnMnCrOnP0/s1600/Escher-Ascending-and-Descending_Lithography_1960.jpg] (accessed: 17.10.2019). 1960
- [FP11] FORSYTH, David A. ; PONCE, Jean: *Computer Vision - A Modern Approach.* 2. edition. Prentice Hall, 2011. – ISBN 0–273–76414–4
- [Gle] GLENN, Linda. *Gestalt images: Significance of colors for perception.* [<http://visual-memory.co.uk/daniel/Documents/visper/visper03.html>] (accessed: 01.10.2019)
- [Gri] GRINGER. *Raster graphics image.* [<http://en.wikipedia.org/wiki/File:Rgb-raster-image.svg>] (accessed: 01.10.2019)
- [GW02] GONZALEZ, Rafael C. ; WOODS, Richard E.: *Digital Image Processing.* 2. edition. Prentice Hall, 2002. – ISBN 0–201–18075–8
- [GW08] GONZALEZ, Rafael C. ; WOODS, Richard E.: *Digital Image Processing.* 3. edition. Prentice Hall, 2008. – ISBN 0–135–05267X

- [GW18] GONZALEZ, Rafael C. ; WOODS, Richard E.: *Digital Image Processing*. 4. edition. Pearson, 2018. – ISBN 9780133356724
- [Hof] HOFFMANN, Gernot. *CIELab Color Space*. [<http://docs-hoffmann.de/cielab03022003.pdf>] (accessed: 01.10.2019)
- [Jäh12] JÄHNE, Bernd: *Digitale Bildverarbeitung: und Bildgewinnung*. 7. edition. Springer, 2012. – ISBN 3–642–04951–6
- [KKS98] KLETTE, Reinhard ; KOSCHAN, Andreas ; SCHLÜNS, Karsten: *Computer Vision*. Vieweg, 1998. – ISBN 9–813–08371–9
- [Kop97] KOPP, Herbert: *Bildverarbeitung interaktiv*. Teubner, 1997. – ISBN 3–519–02995–2
- [Loc] LOCHER, Brent. *Insight to Fourier Transform*. [<http://www.fourier-series.com/f-transform/flash-programs/FourierTrans.html>] (accessed: 01.10.2019)
- [Nag] NAGEL, H. *Traffic sequence showing a taxi in Hamburg*. [http://i21www.ira.uka.de/image_sequences/] (accessed: 01.10.2019)
- [NHFS11] NISCHWITZ, Alfred ; HABERÄCKER, Peter ; FISCHER, Max ; SOCHER, Gundrun: *Computergrafik und Bildverarbeitung: Band II: Bildverarbeitung*. 3. edition. Vieweg, 2011. – ISBN 3–834–81712–0
- [SH91] SAWHNEY, H. S. ; HANSON, A. R.: Identification and 3D description of shallow environmental structure over a sequence of images. In: *In IEEE Computer Soci-*

- ety Conference on Computer Vision and Pattern Recognition (CVPR 91), IEEE, 1991, S. 179–185
- [Sha97] SHAH, Mubarak: *Fundamental of computer vision.* 1997
- [Tön05] TÖNNIES, Klaus D.: *Grundlagen der Bildverarbeitung.* Pearson Studium, 2005.
– ISBN 3-827-37155-4
- [Uni] UNIVERSITY, San Diego S. *The law of common fate.* [<http://www/etc.edu.cn/eet/articles/visualperc1/start.htm>] (accessed: 01.10.2019, no longer available)
- [w3.87] w3.ORG. *Graphics Interchange Format - A standard defining a mechanism for the storage and transmission of raster-based graphics information.* [<https://www.w3.org/Graphics/GIF/spec-gif87.txt>] (accessed: 17.10.2019). 1987
- [WP98] WATT, Alan ; POLICARPO, Fabio: *The Computer Image.* Addison Wesley, 1998. – ISBN 0-201-42298-0

Index

Additive, 5-10

Backscatter X-ray, 1-47

BLPF, 4-59

Blurring, 3-60

Brightness, 2-3, 5-6, 5-13, 5-51

CAT, 1-34

CDF, 3-30

Chroma subsampling, 5-42

Chromaticity, 5-14

CIE, 5-9

CIE $L^*a^*b^*$ model, 5-106

CIE chromaticity diagram, 5-17

CIELAB, 5-103

Closure, 1-72

CMY, 5-23

CMYK, 5-23

Color

all-systems-safe, 5-31

complements, 5-94

consistency, 5-102

continuous spectrum, 5-2

false, 5-67

primary, 5-7

pseudo, 5-67

safe browser, 5-31

safe RGB, 5-31

safe web, 5-31

secondary, 5-10

Color circle, 5-94

Color metrics, 5-6

Color model

device independent, 5-102

Color space, 5-23

Column vector, 3-100

Common fate, 1-76

- Computer vision, 1-12
Computerized axial tomography, 1-34
Computerized tomography, 1-34
Contrast
 enhancement, 3-5
 simultaneous, 1-65
Convolution, 3-57, 4-96
 kernels, 3-57
 masks, 3-57
 theorem, 4-31
Coordinates
 spatial, 1-92
Correlation, 4-96
 auto, 4-98
 cross, 4-96
Covariance matrix, 5-127
CT, 1-34
Cutoff frequency, 4-53
DFT, 4-4
 inverse, 4-4
Digital image, 1-92
Discrete Fourier transform, 4-4
 inverse, 4-4
Distribution function
 cumulative, 3-30
Divide-And-Conquer Strategy, 4-103
Electro magnetic energy spectrum, 1-29
Enhancement
 contrast, 3-5
 image, 3-1
 visualization, 3-88
Euclidean distance, 5-127
Fast Fourier Transform, 4-116
FFT, 4-116
Figure, 1-68
Filter, 3-52, 4-42
 Butterworth lowpass, 4-59
 Gaussian lowpass, 4-64
 Gaussian smoothing, 3-63
 highpass, 4-47

- ideal lowpass, 4-51
- isotropic, 3-85
- linear spatial, 3-53
- lowpass, 4-47
- median, 3-75
- notch, 4-45
- order-statistics, 3-75
- ranking, 3-75
- sharpening, 3-79
- smoothing, 3-60
- spatial, 3-53
- Filter function, 4-42
- Filter transfer function, 4-42
- Fluorescence microscopy, 1-36
- Focal length, 2-9
- Focal point, 2-9
- Fourier
 - spectrum, 4-20
- Fourier transform, 3-53, 4-3
 - discrete, 4-4
 - inverse, 4-3
- Fourier transform pair, 4-3
 - 2D discrete, 4-19
- Frequency
 - component, 4-6
 - domain, 4-6
- Gamma, 3-13
- Gamma correction, 3-13, 5-43
- Gaussian functions, 4-36
- Gestalt Theorie, 1-68
- GLPF, 4-64
- Good continuation, 1-71
- Gradient, 3-100, 3-101
- Gray level, 1-92, 2-3
- Ground, 1-68
- Histogram, 3-20
 - equalization, 3-23, 3-36
 - linearization, 3-36
- HSI, 5-23
- HSI model, 5-51
- Hue, 5-13, 5-51

Illumination, 2-18

Illusions

optical, 1-66

ILPF, 4-51

Image

analysis, 1-1, 1-6

bitmap, 1-14

compression, 3-18

enhancement, 3-1

multippectral, 2-16

plane, 2-5

processing, 1-1, 1-4, 5-67

raster graphics, 1-14

understanding, 1-8

vector graphics, 1-14

Impulse function, 4-33

Intensity, 1-92, 2-3, 5-51

Intensity-level slicing, 3-17

Isotropic filter, 3-85

Kernel, 3-52

Laplacian of Gaussian, 3-96

Laplacian operator, 3-85

Light, 2-3

achromatic, 2-3

chromatic, 2-3

monochromatic, 2-3

LoG, 3-96

Luminance, 2-3, 5-6

Mach bands, 1-63

Machine vision, 1-1

Magnetic resonance imaging, 1-52

Magnitude, 4-7

Masks, 3-52

Mean, 3-45

Median filter, 3-75

Microscopy

fluorescence, 1-36

scanning electron, 1-57

transmission electron, 1-57

Moment, 3-44

- n*th, 3-45
MRT, 1-52
Neighborhood, 3-2
Noise, A-6
 additive stationary Gaussian, A-4
 Data dependent, A-8
 impulse noise, 3-77
 reduction, 3-60
 Salt-and-Pepper, A-7
 white, A-3
NTSC, 5-41
Padded functions, 4-88
Pattern recognition, 1-10
PDF, 3-29
Pels, 1-92
PET, 1-32
Phase angle, 4-7, 4-20
Phase spectrum, 4-7
Photon, 1-31
Picture elements, 1-92
Pixel depth, 5-26
Pixels, 1-92
Point processing, 3-5
Point-spread function, 3-67
Positron emission tomography, 1-32
Power spectrum, 4-8, 4-20, 4-98
Prägnanz, 1-77
Printing
 four-color, 5-48
 six-color, 5-50
Probability density function, 3-29
 uniform, 3-32
Proximity, 1-69
Quantization, 2-21
Radiance, 2-3, 5-6
Random variable, 3-26
Ranking filters, 3-75
Reflectance, 2-18
RGB, 5-23
RGB-Image, 2-17

Roberts gradient operator, 3-103

Rotation invariance, 3-85

Sampling, 2-21

Saturation, 5-13, 5-51

Scanning electron microscope, 1-57

Segmentation, 5-122

SEM, 1-57

Shifting property, 4-34

Signal to noise ratio, A-3

Similarity, 1-70

Smallness, 1-73

SNR, A-3

Sobel operator, 3-106

Spatial filtering, 3-53

Speckle, A-7, A-8

Spectral density, 4-8

Spectrum, 4-7

Splines, 1-14

Standard deviation, 3-46

Standardized observer, 5-2

Statistical error, 3-27

Subtractive, 5-10

Surroundedness, 1-74

Symmetry, 1-75

Systematic error, 3-27

Table lookups, 3-6

TEM, 1-57

Template, 3-52

Thresholding, 3-5

Total absorption, 2-19

Total reflection, 2-19

Transformation function

gray-level, 3-3

intensity, 3-3

mapping, 3-3

single valued and monotonically increasing, 3-25

Transmission electron microscope, 1-57

Trichromatic, 5-14

model, 5-7

Tristimulus values, 5-14

Variable

frequency, 4-20

image, 4-20

spatial, 4-20

transform, 4-20

Variance, 3-46

Visualization of enhancement results, 3-88

wavenumber vector, 2-34

Window, 3-52

Wraparound error, 4-88

YCbCr, 5-43

YQI, 5-41

YUV, 5-23

Zero-crossing, 3-93