The IMAGE PROCESSING Handbook

Fourth Edition

John C. Russ

CRC PRESS

Sony Exhibit 1054
Sony v. Optimum

The IMAGE PROCESSING Handbook

Fourth Edition

John C. Russ

Materials Science and Engineering Department
North Carolina State University
Raleigh, North Carolina



CRC PRESS

Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Russ, John C.
The image processing handbook / John C. Russ. — 4th ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-8493-1142-X (alk. paper)
1. Image processing—Handbooks, manuals, etc. I. Title
TA1637 .R87 2002
621.36'7—dc21

2002073595

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the authors and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

Neither this book nor any part may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, microfilming, and recording, or by any information storage or retrieval system, without prior permission in writing from the publisher.

The consent of CRC Press LLC does not extend to copying for general distribution, for promotion, for creating new works, or for resale. Specific permission must be obtained in writing from CRC Press LLC for such copying.

Direct all inquiries to CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation, without intent to infringe.

Visit the CRC Press Web site at www.crcpress.com

© 2002 by CRC Press LLC

No claim to original U.S. Government works
International Standard Book Number 0-8493-1142-X
Library of Congress Card Number 2002073595
Printed by T.G. Hostench, Spain 3 4 5 6 7 8 9 0
Printed on acid-free paper

Introduction

Image processing is used for two somewhat different purposes:

1. Improving the visual appearance of images to a human viewer
2. Preparing images for measurement of the features and structures present

The techniques that are appropriate for each of these tasks are not always the same, but there is considerable overlap. This book covers methods that are used for both tasks.

To do the best possible job, it helps to know about the uses to which the processed images will be put. For visual enhancement, this means having some familiarity with the human visual process, and an appreciation of what cues the viewer responds to in images. It is also useful to know about the printing or display process, because many images are processed in the context of reproduction or transmission.

The measurement of images generally requires that features be well defined, either by edges or unique brightness, color, texture, or some combination of these factors. The types of measurements that will be performed on entire scenes or individual features are important in determining the appropriate processing steps.

It may help to recall that image processing, similar to food processing or word processing, does not reduce the amount of data present but simply rearranges it. Some arrangements may be more appealing to the senses, and some may convey more meaning, but these two criteria may not be identical nor call for identical methods.

This handbook presents an extensive collection of image processing tools, so that the user of computer-based systems can both understand those methods provided in packaged software, and program those additions that may be needed for particular applications. Comparisons are presented of different algorithms that may be used for similar purposes, using a selection of representative pictures from light and electron microscopes, as well as macroscopic, remote sensing and astronomical images.

In revising the book for this new edition, I tried to respond to some of the comments and requests of readers and reviewers. New chapters on the measurement of images and the subsequent interpretation of the data were added in the second edition, and a section on surface images was added in the third edition. New chapters in this edition discuss in some depth the

stereological interpretation of measurements on sections through three-dimensional structures, and the various logical approaches to feature classification. The sections on the ever-advancing hardware for image capture and printing have been expanded and information added on the newest technologies. More examples have been added in every chapter, and the reference list was expanded and brought up to date.

As in past editions, I resisted suggestions to put "more of the math" into the book. Excellent texts are available on image processing, compression, mathematical morphology, etc. that provide as much rigor and as many derivations as may be needed. Many of them are referenced here, but the thrust of this book remains teaching by example. Few people learn the principles of image processing from the equations. Just as we use images to "do science," so most of us use images to learn about many things, including imaging itself. The hope is that, by seeing what various operations do to representative images, you will discover how and why to use them. Then, if you need to look up the mathematical foundations, they will be easier to understand.

The reader is encouraged to use this book in concert with a real source of images and a computer-based system, and to freely experiment with different methods to determine which are most appropriate for his or her particular needs. Selection of image processing tools to explore images when you do not know the contents beforehand is a much more difficult task than using tools to make it easier for another viewer or a measurement program to see the same things you have discovered. It places greater demand on computing speed and the interactive nature of the interface, but it particularly requires that you become a very analytical observer of images. If you can learn to see what the computer sees, you will become a better viewer and obtain the best possible images, suitable for further processing and analysis.

To facilitate this hands-on learning process, I collaborated with my son, Chris Russ, to produce a CD-ROM that can be used as a companion to this book. The Image Processing Tool Kit contains more than 300 images, many of them the examples from the book, plus nearly 200 Photoshop-compatible plug-ins that implement many of the algorithms discussed here. These can be used with Adobe Photoshop® or any of the numerous programs (some of them free), which implement the Photoshop plug-in interface, on either Macintosh or Windows computers. Information about the CD-ROM is available on-line at http://ReindeerGraphics.com/

Acknowledgments

All the image processing and the creation of the resulting figures included in this book were performed on an Apple Macintosh® computer. Many of the images were acquired directly from various microscopes and other sources using color or monochrome video cameras and digitized directly into the computer. Others were digitized using a digital camera (most with a Polaroid DMC), and some were obtained using a 36-bit color scanner (Agfa), often from images supplied by many co-workers and researchers. These are acknowledged wherever the origin of an image could be determined. A few examples, taken from the literature, are individually referenced.

The book was delivered to the publisher in digital form (on a writable CD), without intermediate hard copy, negatives, or prints of the images, etc. Among other things, this means that the author must bear full responsibility for any errors because no traditional typesetting was involved. (It has also forced me to learn more than I ever hoped to know about some aspects of this technology!) However, going directly from disk file to print also shortens the

time needed in production and helps to keep costs down, while preserving the full quality of the images. Grateful acknowledgment is made of the efforts by the editors at CRC Press to educate me and to accommodate the unusually large number of illustrations in this book (about 2000 figures and a quarter of a million words).

Special thanks are due to Chris Russ (Reindeer Games Inc., Asheville, NC), who has helped to program many of these algorithms and contributed invaluable comments, and to Helen Adams, who has proofread many pages, endured many discussions, and provided the moral support that make writing projects such as this possible.

John C. Russ
Raleigh, NC

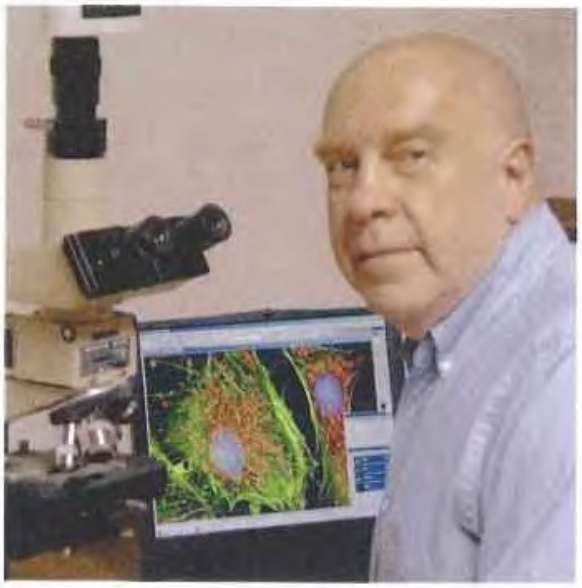


Table of Contents

1 Acquiring Images 1

Human reliance on images for information 1
Video cameras 6
CCD cameras 8
Camera artefacts and limitations 11
Color cameras 13
Camera resolution 15
CMOS cameras 17
Focusing 18
Electronics and bandwidth limitations 18
Pixels 20
Grey-scale resolution 23
Noise 24
High-depth images 26
Color imaging 30
Digital camera limitations 36
Color spaces 40
Color correction 47
Color displays 50
Image types 52
Range imaging 54
Multiple images 59
Stereoscopy 64
Imaging requirements 71

2 Printing and Storage 79

Printing 79

Dots on paper 83
Color printing 88
Printing hardware 93
Film recorders 98
File storage 100
Storage media 101
Magnetic recording 103
Databases for images 104
Browsing and thumbnails 108
Lossless coding 111
Reduced color palettes 117
JPEG compression 118
Wavelet compression 123
Fractal compression 126
Digital movies 128

3 Correcting Imaging Defects 131

Contrast expansion 131
Noisy images 137
Neighborhood averaging 140
Neighborhood ranking 152
Other neighborhood noise-reduction methods 162
Defect removal, maximum entropy, and maximum likelihood 167
Nonuniform illumination 171
Fitting a background function 174
Rank leveling 179
Color shading 181
Non-planar views 188
Computer graphics 189
Geometrical distortion 191
Alignment 193
Interpolation 197
Morphing 202

4 Image Enhancement (Processing in the Spatial Domain) 207

Contrast manipulation 208
Histogram equalization 211
Laplacian 219
Derivatives 229
The Sobel and Kirsch operators 232
Rank operations 248
Texture 257

Fractal analysis 261
Implementation notes 263
Image math 264
Subtracting images 265
Multiplication and division 269

5 Processing Images in Frequency Space 277
Some necessary mathematical preliminaries 277
What frequency space is all about 277
The Fourier transform 278
Fourier transforms of real functions 281
Frequencies and orientations 286
Measuring images in the frequency domain 286
Orientation and spacing 286
Preferred orientation 290
Texture and fractals 295
Filtering images 297
Isolating periodic noise 297
Masks and filters 303
Selection of periodic information 309
Convolution and correlation 314
Fundamentals of convolution 314
Imaging system characteristics 317
Noise and Wiener deconvolution 322
Motion blur 325
Template matching and correlation 328
Autocorrelation 331
Conclusion 332

6 Segmentation and Thresholding 333
Thresholding 333
Multiband images 336
Two-dimensional thresholds 338
Multiband thresholding 341
Thresholding from texture 346
Multiple thresholding criteria 349
Textural orientation 352
Accuracy and reproducibility 357
Including position information 359
Selective histograms 366
Boundary lines 367
Contours 370

Image representation 374
Other segmentation methods 377
The general classification problem 379

7 Processing Binary Images 383
Boolean operations 383
Combining Boolean operations 387
Masks 389
From pixels to features 391
Boolean logic with features 397
Selecting features by location 402
Double thresholding 405
Erosion and dilation 409
Opening and closing 410
Isotropy 413
Measurements using erosion and dilation 417
Extension to grey-scale images 419
Morphology neighborhood parameters 420
Examples of use 421
The custer 425
Euclidean distance map 425
Watershed segmentation 429
Ultimate eroded points 434
Other EDM-based measurements 434
Skeletonization 435
Boundary lines and thickening 437
Combining skeleton and EDM 442

8 Global Image Measurements 445
Global measurements and stereology 445
Surface area 450
ASTM Grain Size 453
Multiple types of surfaces 456
Length 457
Sampling strategies 459
Determining number 461
Curvature, connectivity, and the disector 462
Anisotropy and gradients 466
Size distributions 469
Classical stereology (unfolding) 471

9 Feature-Specific Measurements 475
Brightness measurements 475

Determining location 481
Orientation 484
Neighbor relationships 486
Alignment 491
Counting features 496
Special counting procedures 499
Feature size 502
Circles and ellipses 505
Caliper dimensions 507
Perimeter 509
Describing shape 512
Fractal dimension 515
Harmonic analysis 519
Topology 522
Three-dimensional measurements 523

10 Feature Recognition and Classification 527

Template matching and cross-correlation 527
Parametric description 529
Decision points 534
Multidimensional classification 536
Learning systems 541
kNN and cluster analysis 545
Expert systems 549
Neural nets 551
Syntactical models 553

11 3D Image Acquisition 555

Volume imaging versus sections 555
Basics of reconstruction 558
Algebraic reconstruction methods 563
Maximum entropy 567
Defects in reconstructed images 568
Beam hardening 573
Imaging geometries 578
Three-dimensional tomography 580
High-resolution tomography 587

12 3D Image Visualization 591

Sources of 3D data 591
Serial sections 592
Optical sectioning 596

Sequential removal 598
Stereo measurement 600
3D data sets 604
Slicing the data set 606
Arbitrary section planes 611
The use of color 615
Volumetric display 615
Stereo viewing 618
Special display hardware 623
Ray tracing 624
Reflection 628
Surfaces 633
Multiply connected surfaces 639
Image processing in 3D 643
Measurements on 3D images 647
Conclusion 649

13 Imaging Surfaces 651
Producing surfaces 651
Devices that image surfaces by physical contact 653
Noncontacting measurements 657
Microscopy of surfaces 659
Surface composition imaging 662
Processing of range images 664
Processing of composition maps 668
Data presentation and visualization 669
Rendering and visualization 674
Analysis of surface data 677
Profile measurements 683
The Birmingham measurement suite 687
New approaches — topographic analysis and fractal dimensions 693

References 699

Index 717

1

Acquiring Images

Human reliance on images for information

Humans are primarily visual creatures. Not all animals depend on their eyes, as we do, for 99% or more of the information received about their surroundings. Bats use high-frequency sound, cats have poor vision but a rich sense of smell, snakes locate prey by their heat emission, and fish have organs that sense (and in some cases generate) electrical fields. Even birds, who are highly visual, do not have our eye configuration. With a few exceptions (e.g., owls), their eyes are on opposite sides of their heads, providing nearly 360-degree coverage but little in the way of stereopsis, and they have four or five different color receptors (we have three, loosely described as red, green, and blue). It is difficult to imagine what the world "looks like" to such animals. Even the word "imagine" contains within it our bias toward images, as does much of our language. People with vision defects wear glasses because of their dependence on this sense. We tolerate considerable hearing loss before resorting to a hearing aid and, practically speaking, no prosthetic devices are available for the other senses of touch, taste, and smell.

This bias in everyday life extends to how we pursue more technical goals as well. Scientific instruments commonly produce images to communicate their results to the operator, rather than generating audible tones or emitting a smell. Space missions to other planets and equally arduous explorations of the ocean depths always include cameras as major components, and we judge the success of those missions by the quality of the images returned. This suggests a few of the ways in which we have extended the range of our natural vision. Simple optical devices such as microscopes and telescopes allow us to see things that are vastly smaller or larger than we could otherwise. Beyond the visible portion of the electromagnetic spectrum (a narrow range of wavelengths between about 400 and 700 nanometers), we now have sensors capable of detecting infrared and ultraviolet light, x-rays, radio waves, and, perhaps soon, even gravity waves. **Figure 1** shows an example of an image presenting radio telescope data in the form of an image in which grey scale brightness represents radio intensity. Such devices and presentations are used to further extend our imaging capability.

Signals other than electromagnetic radiation can be used to produce images as well. Novel new types of microscopes that use atomic-scale probes to "feel" the specimen surface present their data as images. Acoustic waves at low frequency produce sonar images, while at gigahertz frequencies

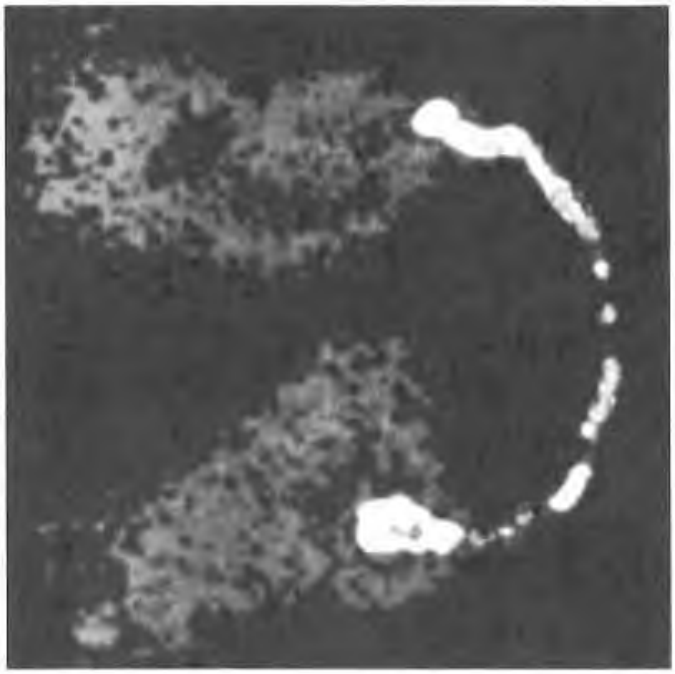


Figure 1. *Radio astronomy produces images such as this view of NGC 1265. These are often displayed with false colors to emphasize subtle variations in signal brightness.*

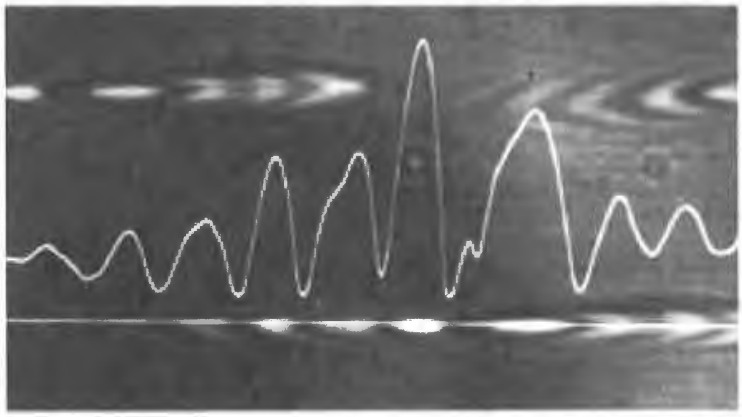


Figure 2. *Scanning acoustic microscope image (with superimposed signal profile along one scan line) of a polished cross section through a composite. The central white feature is a fiber intersecting the surface at an angle. The arcs on either side are interference patterns, which can be used to measure the fiber angle.*

the acoustic microscope produces images with resolution similar to that of the light microscope, but with image contrast that is produced by local variations in the attenuation and refraction of sound waves rather than light. **Figure 2** shows an acoustic microscope image of a composite material, and **Figure 3** shows a sonogram of a baby in the womb.

Some images, such as holograms or electron diffraction patterns, are recorded in terms of brightness as a function of position, but are unfamiliar to the observer. **Figures 4** and **5** show electron diffraction patterns from a transmission electron microscope, in which the atomic structure of the samples is revealed (but only to those who know how to interpret the image). Other kinds of data including weather maps with specialized symbols, graphs of business profit and expenses, and charts with axes representing time, family income, cholesterol level, or even more obscure parameters, have become part of our daily lives. **Figure 6** shows a few examples. The latest developments in computer interfaces and displays make extensive use of graphics, again to take advantage of the large bandwidth of the human visual pathway.



Figure 3. *Surface reconstruction of sonogram imaging, showing a 26-week-old fetus in the womb.*

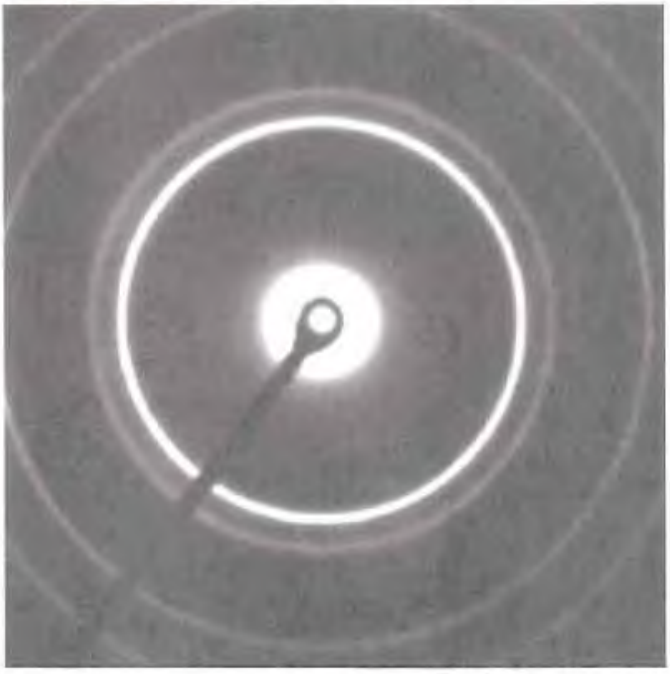


Figure 4. *An electron diffraction pattern from a thin foil of gold. The ring diameters correspond to diffraction angles, which identify the spacings of planes of atoms in the crystal structure.*

Some important differences exist between human vision, the kind of information it yields from images, and the ways in which it appears to do so, and imaging devices that use computers for technical purposes. Human vision is primarily qualitative and comparative, rather than quantitative. We judge the relative size and shape of objects by mentally rotating them to the same orientation, overlapping them in our minds, and performing a direct comparison. This has been shown by tests in which the time required to recognize features as being the same or different is proportional to the degree of misorientation or intervening distance. **Figure 7** shows an example.



Figure 5. *A convergent beam electron diffraction (CBED) pattern from an oxide microcrystal, which can be indexed and measured to provide high accuracy values for the atomic unit cell dimensions.*

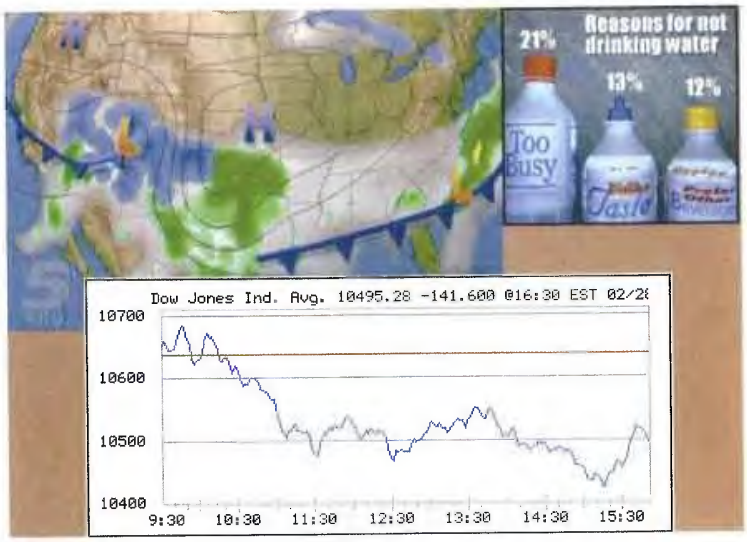


Figure 6. *Typical graphics used to communicate news information include one-dimensional plots such as stock market reports, two-dimensional presentations such as weather maps, and simplified charts using images.*

Humans are especially poor at judging color or brightness of features within images unless they can be exactly compared by making them adjacent. Gradual changes in brightness with position or over time are generally ignored as representing variations in illumination, for which the human visual system compensates automatically. This means that only abrupt changes in brightness are easily seen. Research into human vision indicates that these discontinuities, which usually correspond to physical boundaries or other important structures in the scene being viewed, are extracted from the raw image falling on the retina and sent up to the higher-level processing centers in the cortex.

These characteristics of the visual system are responsible for a variety of visual illusions (**Figure 8**). Some of these illusions enable researchers to study the visual system itself, and others suggest ways that computer-based systems can emulate some of the very efficient (if not always exact) ways that our eyes extract information. Gestalt theory says that this is done by dealing with the

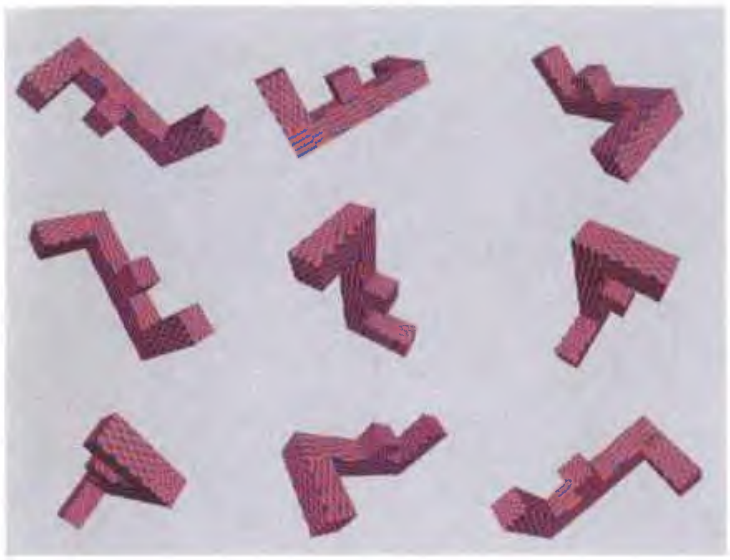


Figure 7. *Several views of a complex three dimensional figure. Which of the representations is/are identical and which are mirror images? The time needed to decide is proportional to the misalignment, indicating that we literally "turn the objects over" in our minds to compare them.*

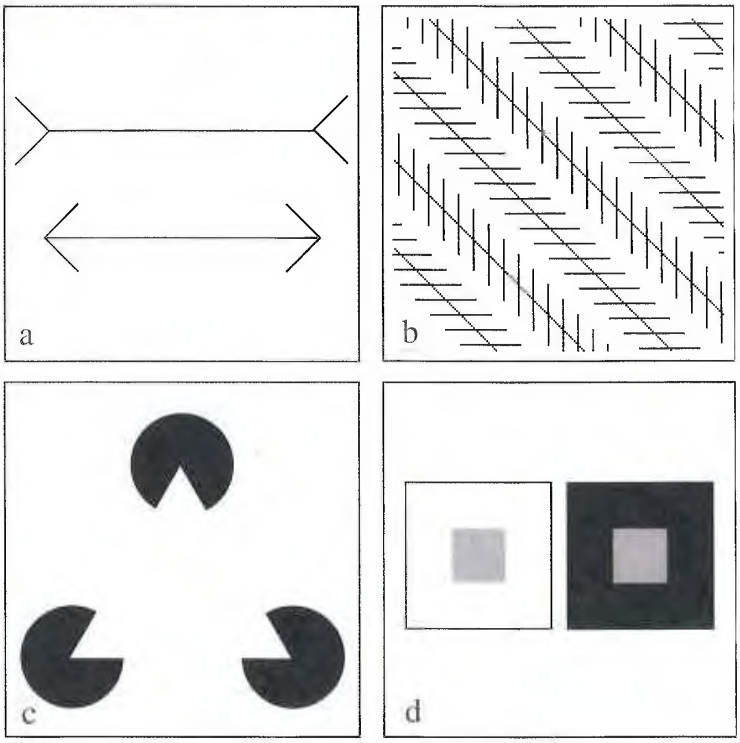


Figure 8. A few common illusions:
*(**a**) the two horizontal lines are identical in length, but appear different because of grouping with the diagonal lines;*
*(**b**) the diagonal lines are parallel, but the crossing lines cause them to appear the diverge due to grouping and inhibition;*
*(**c**) the illusory triangle may appear brighter than the surrounding paper, and is due to grouping and completion;*
*(**d**) the two inner squares have the same brightness, but the surrounding frames cause us to judge them as different due to brightness inhibition.*

image as a whole, rather than by breaking it down to constituent parts. The idea that grouping of parts within images is done automatically by our vision system, and is central to our understanding of scenes, has been confirmed by many experiments and offers important insights into human physiology and psychology. One of the things it explains is why camouflage works (**Figure 9**).

Our purpose in this book is not to study the human visual pathway, but a brief overview can help us to understand how we see things so that we become better observers. The interested reader may want to explore Frisby, 1980; Marr, 1982; Rock, 1984; and Hubel, 1988. Computer-based image processing and analysis use algorithms based on human vision methods when possible, but also employ other methods that seem not to have direct counterparts in human vision.



Figure 9. *An example of camouflage. Patches of color on this rattlesnake break up the overall shape and make it difficult to recognize.*

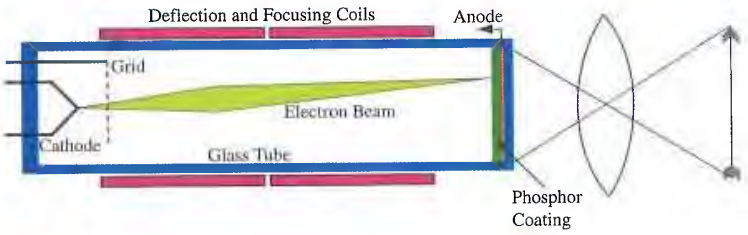


Figure 10. *Functional diagram of a vidicon tube. Light striking the phosphor coating changes its local resistance and, thus, the current that flows as the electron beam scans in a raster pattern.*

Video cameras

When the first edition of this book was published, in 1990, the most common way of acquiring images for computer processing was with a video camera. Mounted onto a microscope, or using appropriate optics to view an experiment, the camera sent an analog signal to a separate "frame grabber" or analog-to-digital converter (ADC) interface board in the computer, which then stored numeric values in memory.

The basic form of the original type of video camera is the vidicon, illustrated in **Figure 10**. It functions by scanning a focused beam of electrons across a phosphor coating applied to the inside of an evacuated glass tube. The light enters the camera through the front glass surface (and a thin metallic anode layer) and creates free electrons in the phosphor. These vary the local conductivity of the layer, so the amount of current that flows to the anode varies as the beam is scanned, according to the local light intensity. This analog (continuously varying) electrical signal is amplified and, as shown in **Figure 11** conforms to standards of voltage and timing (the standards and timing are slightly different in Europe than the U.S., but the basic principles remain the same).

Digitizing the voltage is accomplished by sampling it and generating a comparison voltage. The child's game of "guess a number" illustrates that it takes only eight guesses to arrive at a value that defines the voltage to one part in 256 (the most widely used type of ADC). The first guess is 128, or half the voltage range. If this is, for example, too large, the second guess subtracts 64. Each successive approximation adds or subtracts a value half as large as the previous. In eight steps, the

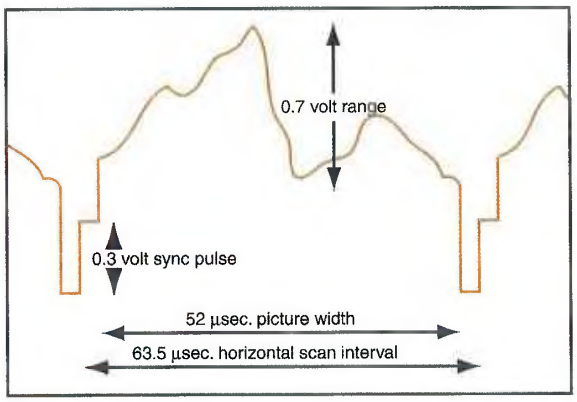


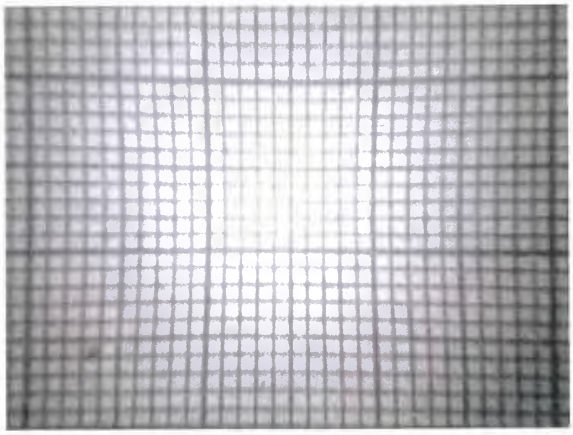
Figure 11. *Standard RS-170 video signal shows the brightness variation along one scan line (ranging between 0 volts = black and 0.7 volts = white).*

final (smallest) adjustment is made. The result is a number that is conveniently stored in the 8-bit memory of most modern computers.

The tube-type camera has several advantages and quite a few drawbacks. Scanning the beam with electromagnetic or electrostatic fields can produce a distorted scan (pincushion or barrel distortion, or more complicated situations) and is subject to further degradation by stray fields from wiring or instrumentation. **Figure 12** shows an example of pincushion distortion, as well as vignetting and loss of focus. Maintaining focus in the corners of the image takes special circuitry, and the corners may also be darkened by the additional thickness of glass through which the light must pass (vignetting). The sealed vacuum systems tend to deteriorate with time, and the "getter" used to adsorb gas molecules may flake and fall onto the phosphor if the camera is used in a vertical orientation. The response of the camera (voltage vs. brightness) approximates the logarithmic response of film and the human eye, but this varies for bright and dark scenes. Recovery from bright scenes and bright spots is slow, and blooming can occur in which bright light produces spots that spread laterally in the coating and appear larger than the features really are.

The tube-type camera does have some advantages. The spatial resolution is very high, limited only by the grain size of the phosphor and the size of the focused beam spot. Also, the phosphor has a spectral response that can be made quite similar to that of the human eye, which sees color from red (about 0.7 µm wavelength) to blue (about 0.4 µm). Adaptations of the basic camera design with intermediate cathode layers or special coatings for intensification are capable of acquiring images in very dim light (e.g., night scenes, fluorescence microscopy).

Figure 12. *Example of an image showing pincushion distortion, as well as loss of focus and vignetting in the edges and corners.*



CCD cameras

The tube-type camera has now been largely supplanted by the solid-state chip camera, the simplest form of which is the CCD (charge-coupled device). The camera chip contains an array of diodes that function as light buckets. Light that enters the semiconductor raises electrons from the valence to the conduction band, so the number of electrons is a direct measure of the light intensity. The diodes are formed by photolithography, so they have a perfectly regular pattern with no image distortion, or sensitivity to the presence of stray fields. The devices are also inexpensive and rugged, compared to tube cameras. CCDs were first invented and patented at Bell Labs (in 1969), have been strongly involved in the NASA-JPL space program because they provide a reliable way to acquire images from satellites and space probes, and now are even displacing film in consumer and professional still cameras.

The basic operation of a CCD is illustrated in **Figure 13**. Each bucket represents one "pixel" in the camera (this word has a lot of different meanings in different contexts, as will be explained below, so it must be used with some care). With anywhere from a few hundred thousand to several million detectors on the chip, it is impractical to run wires directly to each one in order to read out the signal. Instead, the electrons that accumulate in each bucket due to incident photons are transferred, one line at a time, to a readout row. On a clock signal, each column of pixels shifts the charge by one location. This places the contents of the buckets into the readout row, and that row is then shifted, one pixel at a time but much more rapidly, to dump the electrons into an amplifier, where it can be sent out as an analog signal from a video camera, or measured immediately to produce a numeric output from a digital camera.

The simplest way of shifting the electrons is shown in **Figure 14**. Every set of three electrodes on the surface of the device constitutes one pixel. By applying a voltage to two of the electrodes, a field is set up in the semiconductor that acts like a bucket. Electrons are trapped in the central region by the high fields on either side. Note that this does not reduce the area sensitive to incoming photons, because electrons generated in the high field regions will quickly migrate to the low field bucket where they are held. By changing the voltage applied to the regions in six steps or phases, as shown in the figure, the electrons are shifted by one pixel. First, one field region is lowered and the electrons spread into the larger volume. Then the field on the other side is raised, and the electrons have been shifted by one-third of the pixel height. Repeating the process acts like a conveyor belt, and is the reason for the name "charge-coupled device."

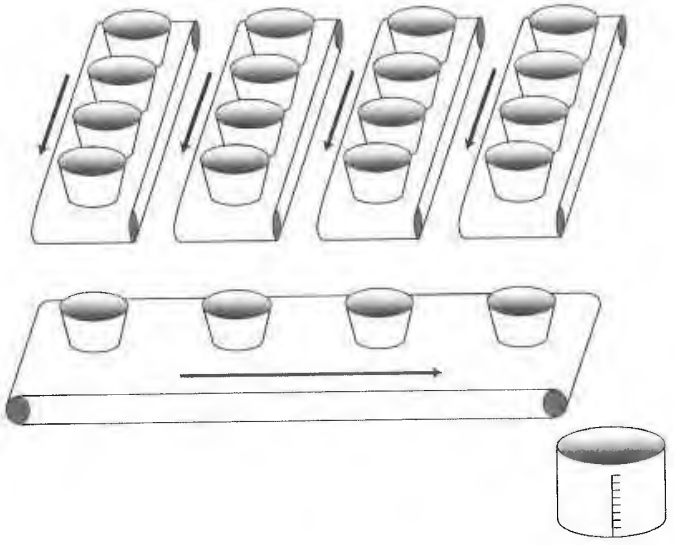


Figure 13. *The basic principle of CCD operation, illustrated as a set of buckets and conveyors (after Janesick, 2001).*

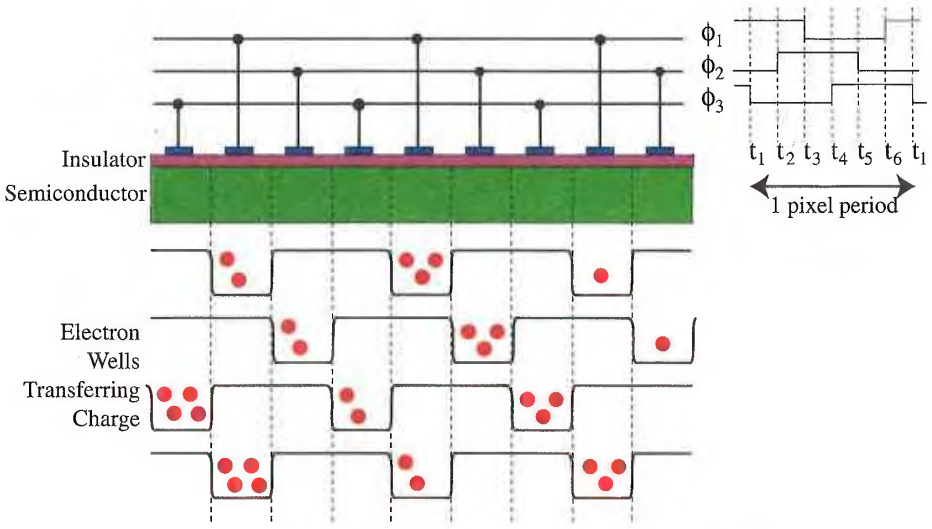
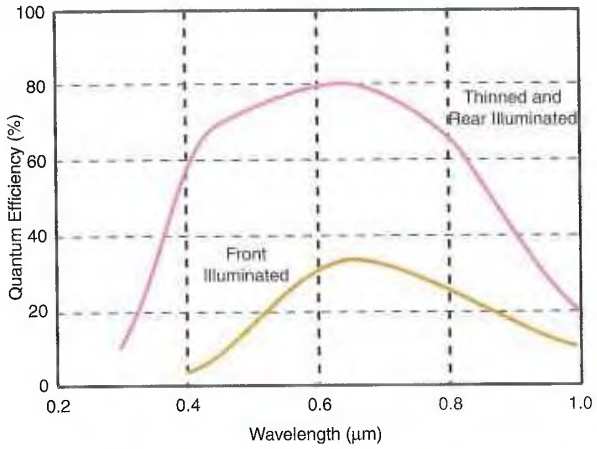


Figure 14. *Varying voltages on a set of three electrodes shifts electrons from one pixel to another.*

One significant problem with the chip camera is its spectral response. Even if the chip is reversed and thinned so that light enters from the side opposite to the electrodes, very little blue light penetrates into the semiconductor to produce electrons. On the other hand, infrared light penetrates easily and these cameras have red and infrared (IR) sensitivity that far exceeds that of human vision, usually requiring the installation of a blocking filter to exclude it (because the IR light is not focused to the same plane as the visible light and would produce blurred or fogged images). **Figure 15** shows this spectral response, which can be further tailored and extended by using materials other than silicon. The chip can reach quite high total efficiency when antireflective coatings are applied, limited primarily by the "fill factor" — the area fraction of the chip that contains active devices between the narrow ditches that maintain electrical separation. Also, the chip camera has an output that is linearly proportional to the incident light intensity, convenient for some measurement purposes but very different from human vision and photographic film, which are logarithmic.

Human vision notices brightness differences of about 2%, that is, a constant ratio of change rather than a constant increment. Film is characterized by a response to light exposure which (after chemical development) produces a density vs. exposure curve such as that shown in **Figure 16**. The low end of this curve represents the fog level of the film, the density that is present even without exposure. At the high end, the film saturates to a maximum optical density, for instance based on the maximum physical density of silver particles. In between, it has a linear relationship whose slope represents the contrast of the film. A steep slope corresponds to a high-contrast film that exhibits a large change in optical density with a small change in exposure. Conversely, a low-contrast film has a broader latitude to record a scene with a greater range of brightnesses. The slope of the curve is usually called "gamma" and some chip cameras, particularly those used for consumer video camcorders, may include circuitry that changes their output from linear to logarithmic so that the image contrast is more familiar to viewers.

When film is exposed directly to electrons, as in the transmission electron micrograph, rather than photons (visible light or x-rays), the response curve is linear rather than logarithmic. Many photons are needed to completely expose a single halide particle for development, but only a single electron. Consequently, electron image films and plates are often very high in density (values greater than 4) which creates difficulties for most scanners.



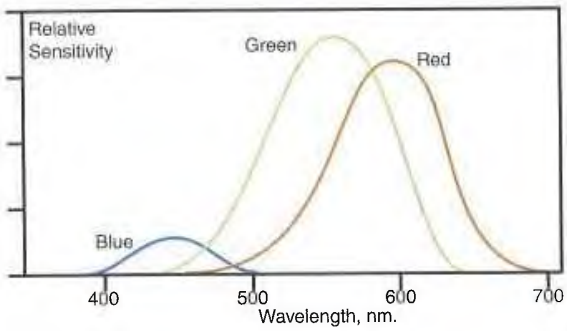


Figure 15. *Spectral response of a silicon based chip*
*(**a**), compared with the color sensors in the human eye*
*(**b**), which are commonly identified as red, green, and blue sensitive but actually cover a range of long, medium, and short wavelengths.*

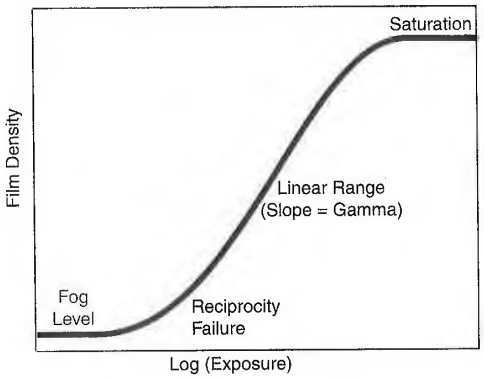
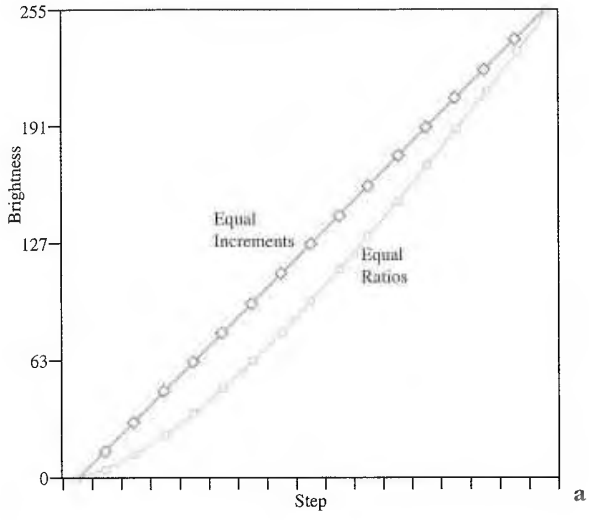
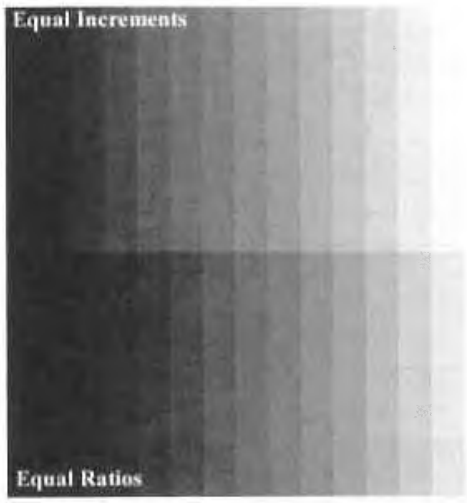


Figure 16. *Response of photographic film. The central portion of the curve shows a linear increase in density (defined as the base-ten logarithm of the fraction of incident light that is transmitted) with the logarithm of exposure. High film contrast ("hard" contrast) corresponds to a steep curve, while "soft" (low contrast, less steep curve) films have a greater dynamic range.*

The trend in camera chips has been to make them smaller and to increase the number of pixels or diodes present. Some scientific chips, such as those used in the Hubble telescope, occupy an entire wafer. But for consumer devices, making them one-third, one-quarter, or even two-tenths of an inch in overall (diagonal) dimension places many devices on a single wafer and allows greater economic yield. Putting more pixels into this reduced chip area (for more spatial resolution, as discussed below) makes the individual detectors small, but the ditches between then have to remain about the same size to prevent electrons from diffusing laterally. The result is to reduce the total efficiency markedly. Some devices place small lenses over the diodes to capture light that would otherwise fall into the ditches, but these add cost and also are not so uniform as the diodes themselves (which are typically within 1% across the entire chip).



a



b

Figure 17. Comparison of visibility of grey level steps from linear (equal steps) and logarithmic (equal ratios) detectors:
*(**a**) plots of intensity;*
*(**b**) display of the values from figure a.*

The other effect of making the detectors small is to reduce their capacity for electrons, called the well capacity. A typical 15 µm pixel in a scientific grade CCD has a capacity of about 500,000 electrons, which with low readout noise (which can be achieved in special situations) of a few electrons gives a dynamic range greater than photographic film. Even larger well capacity and dynamic range can be achieved by combining more detectors in each pixel (binning) by using more steps in the phase shifting. Conversely, reducing the area of the detector also reduces the well size and, with it, the dynamic range. Increasing the noise, for instance by reading out the signal at video rates (each horizontal line in 52 µs for U.S. video), dramatically reduces the dynamic range and a typical consumer grade video camera has no more than about 64 distinguishable brightness levels (studio cameras meet the broadcast video specification of 100 levels). With the chip camera, they are linear with brightness, so they produce even fewer viewable grey levels as shown in **Figure 17**. This performance is much inferior to film, which can distinguish thousands of brightness levels.

Camera artefacts and limitations

Video cameras that use chips have several problems, which contribute to the specific types of defects present in the images which must be dealt with by subsequent processing. One is the fact that video signals are interlaced (**Figure 18**). This clever trick to minimize visual flicker in broadcast television images was accomplished with tube cameras simply by scanning the electron beam in the same interlace pattern as the display television set. With a chip camera, it requires that the array be read out twice for every 30th of a second frame, once to collect the even numbered lines and again for the odd numbered lines. In fact, many cameras combine two lines to get better sensitivity, averaging lines 1 and 2, 3 and 4, 5 and 6, and so on, in one interlace field, and then 2 and 3, 4 and 5, 6 and 7, etc., in the other. This reduces vertical resolution but for casual viewing purposes is not noticeable. Motion can cause the even and odd fields of a full frame to be offset from each other, producing a significant degradation of the image as shown in the figure. A similar effect occurs with stationary images if the horizontal retrace signal is imprecise or difficult for the electronics to lock onto; this is a particular problem with signals played back from consumer video recorders.

During the transfer and readout process, unless the camera is shuttered either mechanically or electrically, photons continue to produce electrons in the chip. This produces a large background signal that further degrades dynamic range and may produce blurring. Electronic shuttering is usually done one line at a time so that moving images are distorted. Some designs avoid shuttering problems by doubling the number of pixels, with half of them opaque to incoming light. A single transfer shifts the electrons from the active detectors to the hidden ones, from which they can be read out. Of course, this reduces the active area (fill factor) of devices on the chip, costing 50% in sensitivity.

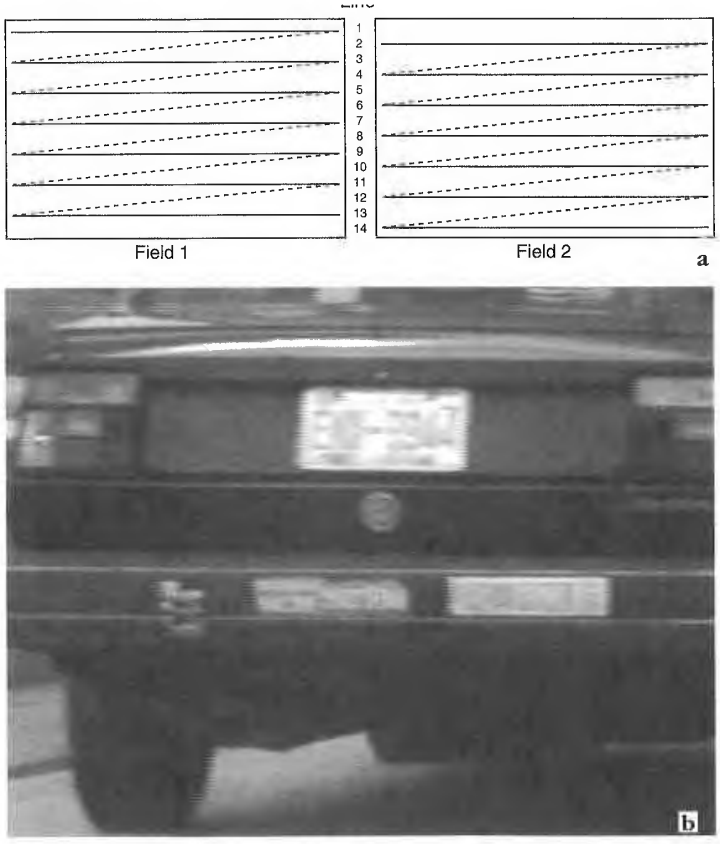


Figure 18. *Interlace scan covers even numbered lines in a 1/60th of a second field, and even numbered lines in a second field (**a**). When motion is present, this produces an offset in the complete image (**b**).*

The high speed of horizontal line readout can produce horizontal blurring of the signal, again reducing image resolution. This is partially due to inadequate time for the electrons to diffuse along with the shifting fields and the time needed to recover electrons from traps (impurities in the silicon lattice), and partially to the inadequate frequency response of the amplifier, which is a tradeoff with amplifier noise. Even though the individual electron transfers are very efficient, better than 99.99% in most cases, the result of being passed through many such transfers before being collected and amplified increases the noise. This varies from one side of the chip to the other, and from the top to the bottom, and can be visually detected in images if there is not a lot of other detail or motion to obscure it.

Many transfers of electrons from one detector to another occur during readout of a chip, and this accounts for some of the noise in the signal. Purely statistical variations in the production and collection of charge is a relatively smaller effect. The conversion of the tiny charge to a voltage and its subsequent amplification is the greatest source of noise in most systems. Readout and amplifier noise can be reduced by slowing the transfer process so that fewer electrons are lost in the shifting process and the amplifier time constant can integrate out more of the noise, producing a cleaner signal. Cooling the chip to about –40°C also reduces the noise from these sources and from dark current, or thermal electrons. Slow readout and cooling are used only in non-video applications, of course. Janesick (2001) discusses the various sources of noise and their control in scientific CCDs of the type used in astronomical imaging (where they have almost entirely replaced film) and in space probes.

Color cameras

Color cameras can be designed in three principle ways, as shown in **Figures 19**, **20** and **21**. For stationary images (which includes many scientific applications such as microscopy, but excludes "real-time" applications such as video), a single detector array can be used to acquire three sequential exposures through red, green and blue filters, respectively, which are then combined for viewing (**Figure 19**). The advantages of this scheme include low cost and the ability to use different exposure times for the different color bands, which can compensate for the poorer sensitivity of the CCD chip to blue light.

Many high-end consumer and most professional and scientific grade video cameras use three sensors (**Figure 20**). A prism array splits the incoming light into red, green, and blue components

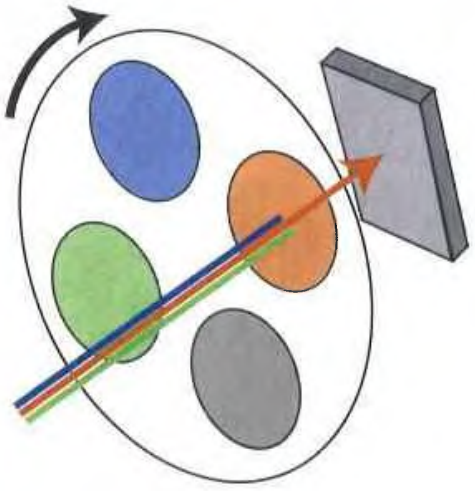
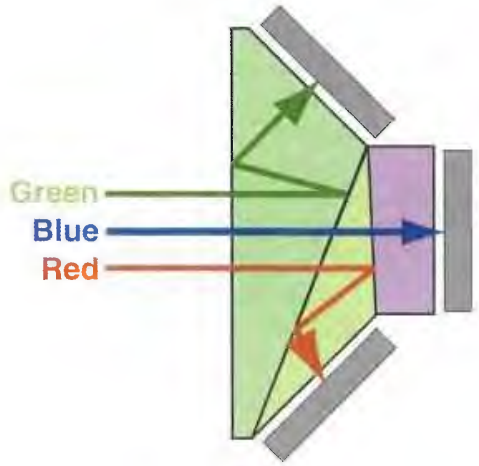
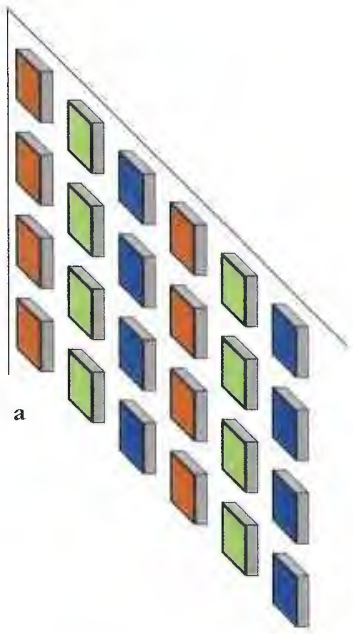


Figure 19. *Schematic diagram of a color wheel camera. The fourth filter position is empty, allowing the camera to be used as a monochrome detector with greater sensitivity for dim images (e.g., fluorescence microscopy).*

Figure 20. *Schematic diagram of a three-chip color camera.*





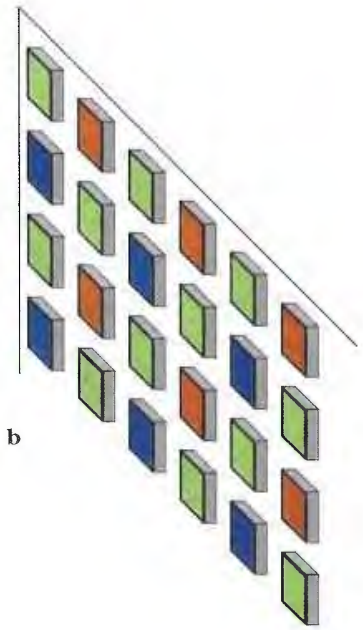


Figure 21. *Stripe (**a**) and Bayer (**b**) filter patterns used in single chip cameras.*

which are recorded by three different sensors whose outputs are combined electronically to produce a standard video image. This approach is more costly, because three chips are needed, but for video applications they need not be of particularly high resolution (fewer pixels). The optics and hardware to keep everything in alignment add some cost, and the depth of the prism optics makes it impractical to use short focal length (wide angle) lenses.

Video images are often digitized into a 640 × 480 array of stored pixels, but this is not the actual resolution of the image. The broadcast bandwidth limits the high frequencies and eliminates any rapid variations in brightness and color. A video image has no more than 330 actual elements of resolution in the horizontal direction for the brightness (luminance) signal, and about half that for the color (chrominance) information. Color information is intentionally reduced in resolution because human vision is not very sensitive to blurring of color beyond boundary lines.

Of course, video signals can be further degraded by poor equipment. Recording video on consumer-grade tape machines can reduce the resolution by another 50% or more, particularly if the record head is dirty or the tape has been used many times before. Video images are just not very high resolution, although some forms of HDTV (high definition television) may improve things in the future. Consequently, video technology is usually a poor choice for scientific imaging unless there is some special need to capture "real-time" images (i.e., 25–30 frames per second) to record changes or motion. Digital cameras have largely replaced them now, as they produce much higher resolution images and at lower cost.

Most digital cameras use a single pixel array, often of very high pixel (detector) count, with a color filter that allows red, green and blue light to reach specific detectors. Different patterns may be used (**Figure 21**), with the Bayer pattern being very common. Notice that it assigns twice as many detectors for green as for red or blue, which mimics to some extent the human eye's greater sensitivity to green. The problem with the single-chip camera, of course, is that the image resolution in each color channel is reduced. The red intensity at some locations must be interpolated from nearby sensors, for example. It is also necessary to design the filters to give the same brightness sensitivity in each channel. If this is not done well, a herringbone pattern will appear in images of a uniform grey test card and color fringes will appear along edges in the picture.

Interpolation techniques for Bayer pattern color filters reduce the image resolution as compared to the number of individual detectors in the camera (which is generally the specification advertised by the manufacturer). The quality of the interpolation, judged by its ability to preserve sharp boundaries in brightness while minimizing the introduction of color artefacts, varies inversely with the computational requirements. A comparison of several patented methods can be found in (Ramanesh, 2000).

Camera resolution

The signal coming from the chip is an analog voltage, of course, even if the digitization takes place in the camera itself, so the interpolation is done in the amplifier stage, by the use of appropriate time constants. This means that the voltage cannot vary rapidly enough to correspond to brightness differences for every pixel. In most cases, because of this interpolation, the actual image resolution with a single chip camera and filter arrangement will be one-half to two-thirds the value that might be expected by the advertised number of pixels in the camera. And some cameras record images with many more stored pixels than the chip resolution would warrant in any case. Such empty magnification contributes no additional information in the image.

Comparing cameras based on actual resolution rather than the stated number of recorded pixels can be quite difficult. It is important to consider the multiple meanings of the word "pixel." In some contexts, it refers to the number of light detectors in the camera (without regard to any color filtering, and sometimes including ones around the edges that do not contribute to the actual image but are

used to measure dark current). In other contexts, it describes the number of recorded brightness or color values stored in the computer, although these may represent empty magnification. In other situations, it is used to describe the displayed points of light on the computer monitor, even if the image is shown in a compressed or enlarged size. It makes much more sense to separate these various meanings and to talk about resolution elements when considering real image resolution. This refers to the number of discrete points across the image that can be distinguished from each other, and is sometimes described in terms of the number of line pairs that can be resolved. This is half the number of resolution elements, because one element is needed for the line and one for the space between lines, and it depends on the amount of brightness contrast between the lines and the spaces and, of course, the amount of noise (random variations) present in the image.

The situation is even more complicated with some digital cameras that shift the detector array to capture multiple samples of the image. The most common method is to use a piezo device to offset the array by half the pixel spacing in the horizontal and vertical directions, capturing four images that can be combined to nearly double the resolution of the image as data are acquired from the gaps between the original pixel positions. For an array with colored filters, additional shifts can produce color images with resolution approaching that corresponding to the pixel spacing; and some studio cameras displace the entire sensor array to different regions of the film plane to collect tiles that are subsequently assembled into an image several times as large as the detector array. Of course, the multiple exposures required with these methods means that more time is required to acquire the image.

With either the single-chip or three-chip camera, the blue channel is typically the noisiest due to the low chip sensitivity to blue light and the consequent need for greater amplification. In many cases, processing software that reduces image noise using one of the averaging or median filters discussed in Chapter 3 can be applied separately to each color channel, using different parameters according to the actual noise content, to best improve image appearance.

Because of the longer exposure times, which collect more electrons and so reduce noise due to statistics and amplification, and because of the much slower readout of the data from the chip, which may take several seconds instead of 1/30th of a second, digital cameras using the same chip technology as a video camera can produce much better image quality. Digital still cameras read out the data in one single pass (progressive scan), not with an interlace. By cooling the chip and amplifier circuitry to reduce dark currents, integration (long exposures up to tens of seconds or, for some astronomical applications, many minutes) can be used to advantage because of the high dynamic range of some chip designs. In addition, the ability to use a physical rather than electronic shutter simplifies chip circuitry and increases fill factor. The number of pixels in video cameras need not be any greater than the resolution of the video signal, which, as noted previously, is quite poor. In a digital still camera, very high pixel counts can give rise to extremely high resolution, which is beginning to rival film in some cases.

An interesting crossover is also occurring between high-end consumer and professional scientific-grade cameras. In addition to dedicated cameras for attachment to microscopes or other separate optics, manufacturers are producing consumer cameras with enough resolution (3–6 million pixels at this writing) that it is becoming practical to use them in technical applications, and simple optical attachments are making it easy to connect them to microscopes. **Figure 22** shows two of the cameras in use in the author's laboratory. The large, dedicated camera cost more than six times the small one. The small one is also portable with its own storage, and useful for many other applications. At the same time, professional digital cameras with extremely high resolution detector arrays, interchangeable lenses, etc., are providing capabilities that compete strongly with traditional 35mm and larger film cameras. Every manufacturer of cameras has recognized the shift away from film and toward digital recording, and an incredibly wide variety of cameras is now available with new developments appearing frequently.



Figure 22. *Two digital cameras mounted on the author's microscope. The top camera (Polaroid DMC Ie) provides greater bit depth and compensation for illumination temperature. The one mounted on the eyepiece (Nikon 990) has greater spatial resolution, and can be used without a tethered computer. Both cameras store images as uncompressed TIFF files.*

CMOS cameras

A competitor to the CCD design for chips now appearing on the horizon, CMOS (complementary metal oxide semiconductor) uses a different fabrication approach that promises several potential advantages. The most important is cost, because the manufacturing process is similar to that used for other common chips such as computer memory, whereas CCDs require separate fabrication lines. It is also possible to incorporate additional circuitry on the CMOS chip, such as the amplifiers and readout and control electronics, and the digitizer, leading to the possibility of a single-chip camera. This may lead to less expensive systems, but with more costly development procedures. The devices will also be more compact and possibly more rugged. CMOS devices also require lower voltages and have lower power requirements, which can be important for portable devices.

Another potential advantage is windowing, the possibility to read out a portion of the array rather than the entire array, because row–column addressing (just like memory chips) can be used to access any pixel. This can provide rapid updating for focusing, measurement of regions of interest, etc.

On the other hand, CMOS devices have much higher noise than CCDs. Partly, this is due to the fact that the CCD has less on-chip circuitry and uses a common output amplifier for less variability. CMOS has higher dark currents, and thus more noise. Also, there is more capacitance in the output wiring, which degrades amplifier noise characteristics. CMOS uses amplifiers for each pixel, producing a unique nonuniformity for each chip called fixed pattern noise (FPN). FPN can in

principle be removed by recording the pattern and post processing with software to remove it, but this further reduces the useful dynamic range. The amplifiers also take up more area on the chip, further reducing fill factor, and with it well capacity and dynamic range. Each pixel requires at least two or three transistors to control the detector, plus the amplifier, and fill factors are often less than 30%. The sensitivity and dynamic range of the CCD is much greater, due to the larger size of the detector, lower noise, etc.

Although CMOS detector production facilities are similar in principle to those used for computer memory and other devices, in fact the quest for lower noise has led to the use of special fabrication processes and techniques to reduce trapping and other defects, which largely eliminates this commonality. Defect control for acceptable yields currently limits CMOS arrays to 1–2 million pixels, whereas CCDs with up to 16 million pixels are currently available.

CMOS is expected to dominate in the low-end consumer market for camcorders and snapshot cameras, toys, and other specialty applications such as surveillance video cameras, but not to challenge CCDs for scientific and technical applications requiring high fidelity, resolution, and dynamic range for some time to come.

Focusing

Regardless of what type of camera is employed to acquire images, it is important to focus the optics correctly to capture the fine details in the image. Usually, the human eye is used to performing this task. In some situations, such as automated microscopy of pathology slides or surveillance tracking of vehicles, automatic focusing is required. This brings computer processing into the initial step of image capture. Sometimes, in the interests of speed, the processing is performed in dedicated hardware circuits attached to the camera; but, in any case, the algorithms are the same as might be applied in the computer. In many cases the focusing is accomplished in software by stepping the optics through a range of settings and choosing the one that gives the "best" picture.

Several different approaches to automatic focus are used. Cameras used for macroscopic scenes may employ methods that use some distance-measuring technology (e.g., using high frequency sound or infrared light to determine the distance to the subject so that the lens position can be adjusted). In microscopy applications, this is impractical and the variation in the image itself with focus adjustment must be used. Various algorithms are used to detect the quality of image sharpness, and all are successful for the majority of images in which there is good contrast and fine detail present. Each approach selects some implementation of a high-pass filter output that can be realized in various ways, using either hardware or software, but must take into account the effect of high frequency noise in the image and the optical transfer function of the optics (Green et al., 1985; Firestone et al., 1991; Boddeke et al., 1994).

Electronics and bandwidth limitations

Video cameras of either the solid-state or tube type produce analog voltage signals corresponding to the brightness at different points in the image. In the standard RS-170 signal convention, the voltage varies over a 0.7-volt range from minimum to maximum brightness, as shown above in Figure 11. The scan is nominally 525 lines per full frame, with two interlaced 1/60th-second fields combining to make an entire image. Only about 480 of the scan lines are actually usable, with the remainder lost during vertical retrace. In a typical broadcast television picture, more of these lines are lost due to overscanning, leaving about 400 lines in the actual viewed area. The time duration of each scan line is 62.5 µs, part of which is used for horizontal retrace. This leaves 52 µs for the image data, which must be subdivided into the horizontal spacing of discernible pixels. For PAL (European) television, these values are slightly different, based on a 1/25th-second frame time and more scan lines, but the resulting resolution is similar.

Broadcast television stations are given only a 4-MHz bandwidth for their signals, which must carry color and sound information as well as the brightness signal we have so far been discussing. This narrow bandwidth limits the number of separate voltage values that can be distinguished along each scan line to a maximum of 330, as mentioned previously, and this value is reduced if the signal is degraded by the electronics or by recording using standard videotape recorders. Consumer-quality videotape recorders reduce the effective resolution substantially; in "freeze frame" playback, they display only one of the two interlaced fields, so that only about 200 lines are resolved vertically. Using such equipment as part of an image analysis system makes choices of cameras or digitizer cards on the basis of resolution quite irrelevant.

Even the best system can be degraded in performance by such simple things as cables, connectors, or incorrect termination impedance. Another practical caution in the use of standard cameras is to avoid automatic gain or brightness compensation circuits. These can change the image contrast or linearity in response to bright or dark regions that do not even lie within the digitized portion of the image, and increase the gain and noise for a dim signal.

Figure 23 shows a micrograph with its brightness histogram. This is an important tool for image analysis, which plots the number of pixels as a function of their brightness values. The histogram as shown is well spread out over the available 256 brightness levels, with peaks corresponding to each of the phases in the metal sample. If a bright light falls on a portion of the detector in the solid-state camera that is not part of the image area of interest (e.g. due to internal reflections in the optics), automatic gain circuits in the camera may alter the brightness–voltage relationship so that the image changes. This same effect occurs when a white or dark mask is used to surround images placed under a camera on a copy stand. The relationship between structure and brightness is changed, making subsequent analysis more difficult.

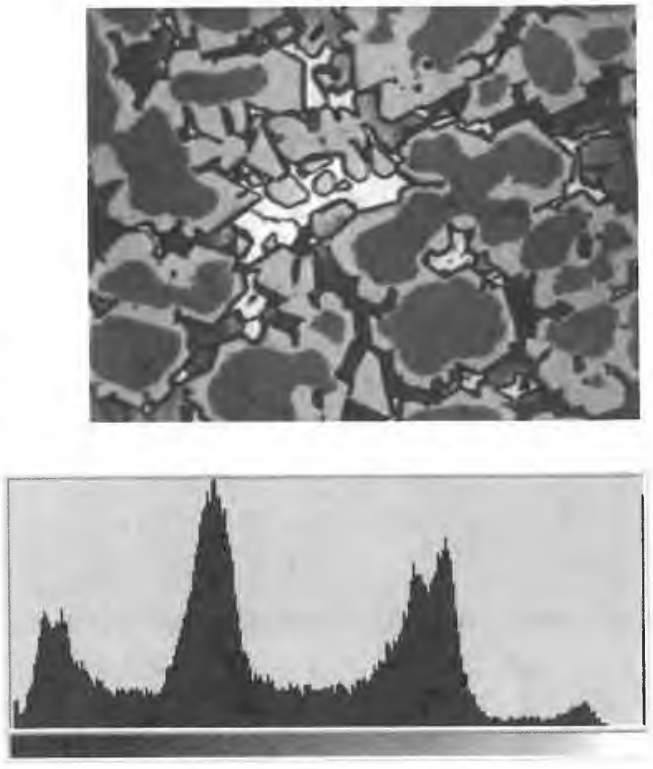


Figure 23. *A grey-scale image digitized from a metallographic microscope and its brightness histogram, which plots the number of pixels with each possible brightness value.*

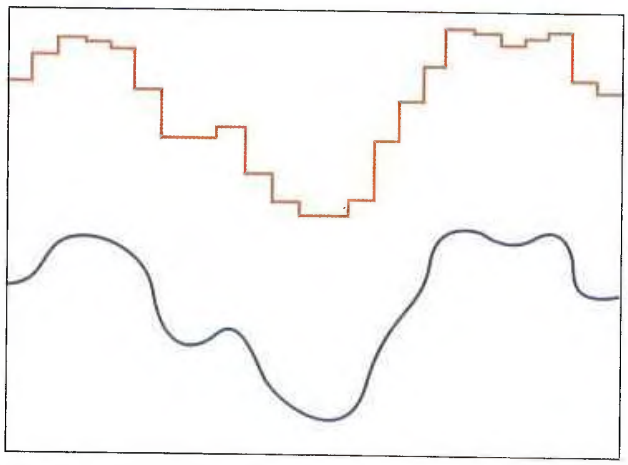


Figure 24. *Digitization of an analog voltage signal such as one line in a video image (blue) produces a series of numbers that represent a series of steps (red) equal in time and rounded to integral multiples of the smallest measurable height increment.*

Issues involving color correction and calibration will be dealt with in the following paragraph, but obtaining absolute color information from video cameras is nearly impossible considering the broad range of wavelengths passed through each filter, the variation in illumination color (e.g., with slight voltage changes on an incandescent bulb) and the way the color information is encoded.

The analog voltage signal is usually digitized with an 8-bit "flash" ADC (analog-to-digital converter). This is a chip using successive approximation techniques to rapidly sample and measure the voltage in less than 100 ns, producing a number value from 0 to 255 that represents the brightness. This number is immediately stored in memory and another reading made, so that a series of brightness values is obtained along each scan line. **Figure 24** illustrates the digitization of a signal into equal steps both in time and value. Additional circuitry is needed to trigger each series of readings on the retrace events so that positions along successive lines are consistent. Digitizing several hundred points along each scan line, repeating the process for each line, and storing the values into memory while adjusting for the interlace of alternate fields produces a digitized image for further processing or analysis.

Pixels

It is most desirable to have the spacing of the pixel values be the same in the horizontal and vertical directions (i.e., square pixels), as this simplifies many processing and measurement operations. Accomplishing this goal requires a well-adjusted clock to control the acquisition. Since the standard video image is not square, but has a width-to-height ratio of 4:3, the digitized image may represent only a portion of the entire field of view. Digitizing boards, also known as frame grabbers, were first designed to record 512 × 512 arrays of values, because the power-of-two dimension simplified design and memory addressing. Most of the current generation of boards acquire a 640 wide by 480 high array, which matches the image proportions and the size of standard VGA display monitors while keeping the pixels square. Because of the variation in clocks between cameras and digitizers, it is common to find distortions of a few percent in pixel squareness. This can be measured and compensated for after acquisition by resampling the pixels in the image as discussed in Chapter 3.

Of course, digitizing 640 values along a scan line that is limited by electronic bandwidth and contains only 300+ meaningfully different values produces an image with unsharp or fuzzy edges and "empty" magnification. Cameras that are capable of resolving more than 600 points along each scan line can sometimes be connected directly to the digitizing electronics to reduce this loss of horizontal resolution. Digital camera designs bypass the analog transmission altogether, sending

digital values to the computer, but these are generally slower than standard video systems as discussed previously.

Pixels have a finite area, so those that straddle a boundary effectively average the brightness levels of two regions and have an intermediate brightness that depends on how the pixels lie with respect to the boundary. This means that a high lateral pixel resolution and a large number of distinguishable grey levels are needed to accurately locate boundaries. **Figure 25** shows several examples of an image with varying numbers of pixels across its width, and **Figure 26** shows the same image with varying numbers of grey levels.



***Figure 25**. Four representations of the same image, with variation in the number of pixels used: (**a**) 256 × 256; (**b**) 128 × 128; (**c**) 64 × 64, (**d**)32 × 32. In all cases, a full 256 grey values are retained. Each step in coarsening of the image is accomplished by averaging the brightness of the region covered by the larger pixels.*

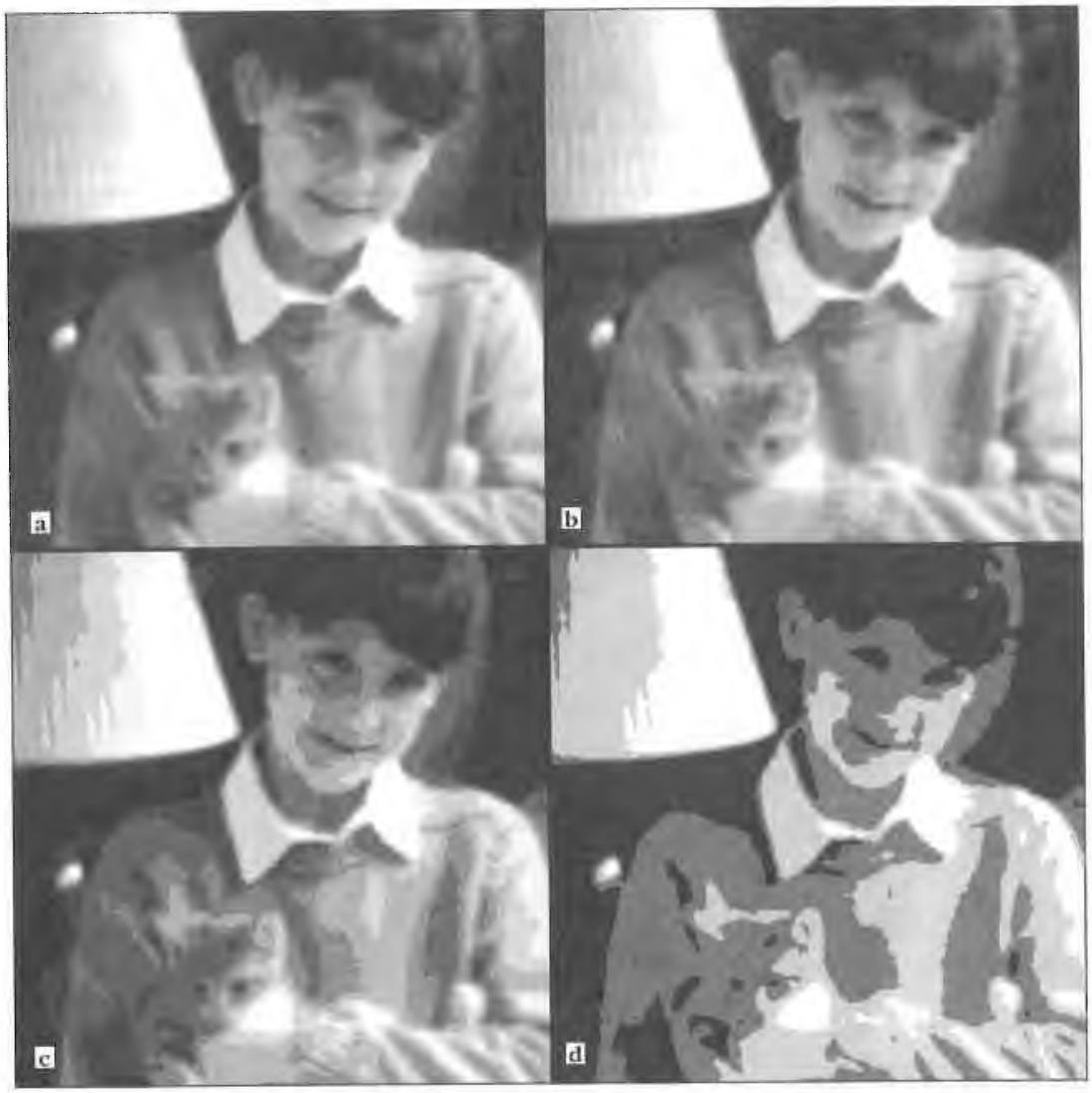


Figure 26. Four representations of the same image, with variation in the number of grey levels used: (a) *32;* ***(b)*** *16;* ***(c)*** *8;* ***(d)*** *4. In all cases, a full 256 × 256 array of pixels are retained. Each step in the coarsening of the image is accomplished by rounding the brightness of the original pixel value.*

For the most common types of image acquisition devices, such as video cameras, the pixels represent an averaging of the signal across a finite area of the scene or specimen; however, this is not so in other situations. At low magnification, for example, the scanning electron microscope beam samples a volume of the specimen much smaller than the dimension of a pixel in the image, as does the probe tip in a scanned probe microscope. Range imaging of the moon from the Clementine orbiter read the elevation of points about 10 cm in diameter using a laser range finder, but the points were spaced apart by 100 meters or more.

In these cases, the interpretation of the relationship between adjacent pixels is slightly different. Instead of averaging across boundaries, the pixels sample points that are discrete and well separated. Cases of intermediate or gradually varying values from pixel to pixel are rare, and the

problem instead becomes how to locate a boundary between two sampled points on either side. If there are many points along both sides of the boundary, and the boundary can be assumed to have some geometric shape (such as a straight line), fitting methods can be used to locate it to a fraction of the pixel spacing. These methods are discussed further in Chapter 9 on image measurements.

Grey-scale resolution

In addition to defining the number of sampled points along each scan line, and thus the resolution of the image, the design of the ADC also controls the precision of each measurement. Inexpensive commercial flash analog-to-digital converters usually measure each voltage reading to produce an 8-bit number from 0 to 255. This range may not be used entirely for an actual image, which may not vary from full black to white. Also, the quality of most cameras and other associated electronics rarely produces voltages that are free enough from electronic noise to justify full 8-bit digitization anyway. A typical "good" camera specification of 49-dB signal-to-noise ratio implies that only 7 bits of real information are available, and the eighth bit is random noise; but 8 bits corresponds nicely to the most common organization of computer memory into bytes, so that one byte of storage can hold the brightness value from one pixel in the image.

Some cameras and most scanners produce more than 256 distinguishable brightness values, and for these it is common to store the data in two bytes or 16 bits, giving a possible range of 65536:1, which exceeds the capability of any current imaging device (but not some other sources of data that may be displayed as images, such as surface elevation measured with a scanned probe as discussed in Chapter 13). For a camera with a 10- or 12-bit output, the values are simply shifted over to the most significant bits and the low order bits are either zero or random values. For display and printing purposes 8 bits is enough, but the additional depth can be very important for processing and measurement, as discussed in subsequent chapters. In some systems the histogram of values is still expressed as 0–255 for compatibility with the more common 8-bit range, but instead of being restricted to integers, the brightness values can take on real values.

Figure 27 shows an image that appears to be a uniform grey. When the contrast range is expanded we can see the faint lettering present on the back of this photographic print. Also evident is a series of vertical lines which are due to the digitizing circuitry, in this case electronic noise from the high-frequency clock used to control the time base for the digitization. Nonlinearities in the ADC, electronic noise from the camera itself, and degradation in the amplifier circuitry combine to make the lowest two bits of most standard video images useless, so that only about 64 grey levels are actually distinguishable in the data. As noted previously, higher performance cameras and circuits exist, but do not generally offer "real-time" video speed (30 frames per second).

When this stored image is subsequently displayed from memory, the numbers are used in a digital-to-analog converter to produce voltages that control the brightness of a display, often a cathode ray tube (CRT). This process is comparatively noise-free and high resolution, since computer display technology has been developed to a high level for other purposes. A monochrome (black/grey/white) image displayed in this way, with 640 × 480 pixels, each of which can be set to one of 256 brightness levels (or colors using pseudo-color techniques to be discussed below), can be used with many desktop computers.

The human eye cannot distinguish all 256 different levels of brightness in this type of display, nor can they be successfully recorded or printed using inexpensive technology, such as ink-jet or laser printers discussed in Chapter 2. About 20–30 grey levels can be visually distinguished on a CRT or photographic print, suggesting that the performance of the digitizers in this regard is more than adequate, at least for those applications where the performance of the eye was enough to begin with.

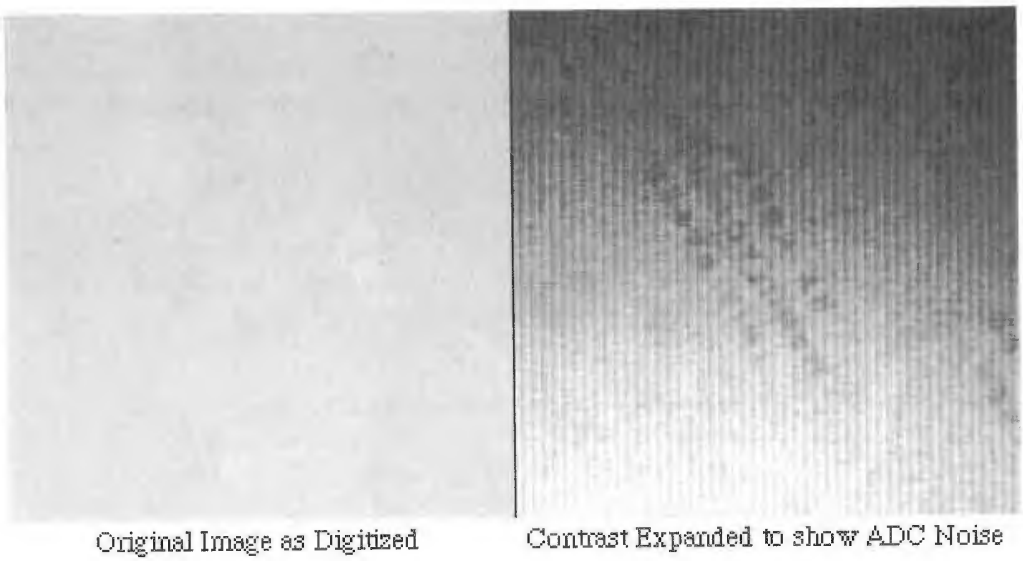


Figure 27. *Digitized camera image of the back of a photographic print, showing the periodic noise present in the lowest few bits of the data from the electronics (especially the clock) in the analog to digital converter.*

Images acquired in very dim light, or some other imaging modalities such as x-ray mapping in the scanning electron microscope (SEM), impose another limitation of the grey-scale depth of the image. When the number of photons (or other particles) collected for each image pixel is low, statistical fluctuations become important. **Figure 28** shows a fluorescence microscope image in which a single video frame illustrates extreme statistical noise, which would prevent distinguishing or measuring the structures present.

Noise

Images in which the pixel values vary even within regions that are ideally uniform in the original scene can arise either because of limited counting statistics for the photons or other signals, losses introduced in the shifting of electrons within the chip, or due to electronic noise in the amplifiers or cabling. In either case, the variation is generally referred to as noise, and the ratio of the contrast which is actually due to structural differences in the image to the noise level is the signal-to-noise ratio. When this is low, the features present may be invisible to the observer. **Figure 29** shows an example in which several features of different size and shape are superimposed on a noisy background with different signal-to-noise ratios. The ability to discern the presence of the features is generally proportional to their area, and independent of shape.

In the figure, a smoothing operation is performed on the image with the poorest signal-to-noise ratio, which somewhat improves the visibility of the features. The methods available for smoothing noisy images by image processing are discussed in the chapters on spatial and frequency domain methods; however, the best approach to noisy images, when it is available, is simply to collect more signal and improve the statistics.

Adding together more video frames is shown in **Figure 28**. The improvement in quality is proportional to the square root of the number of frames. Because each frame is digitized to 8 bits, adding together up to 256 frames as shown requires a total image storage capability that is 16 bits (2 bytes) deep. Acquiring frames and adding together the pixels at video rates generally requires

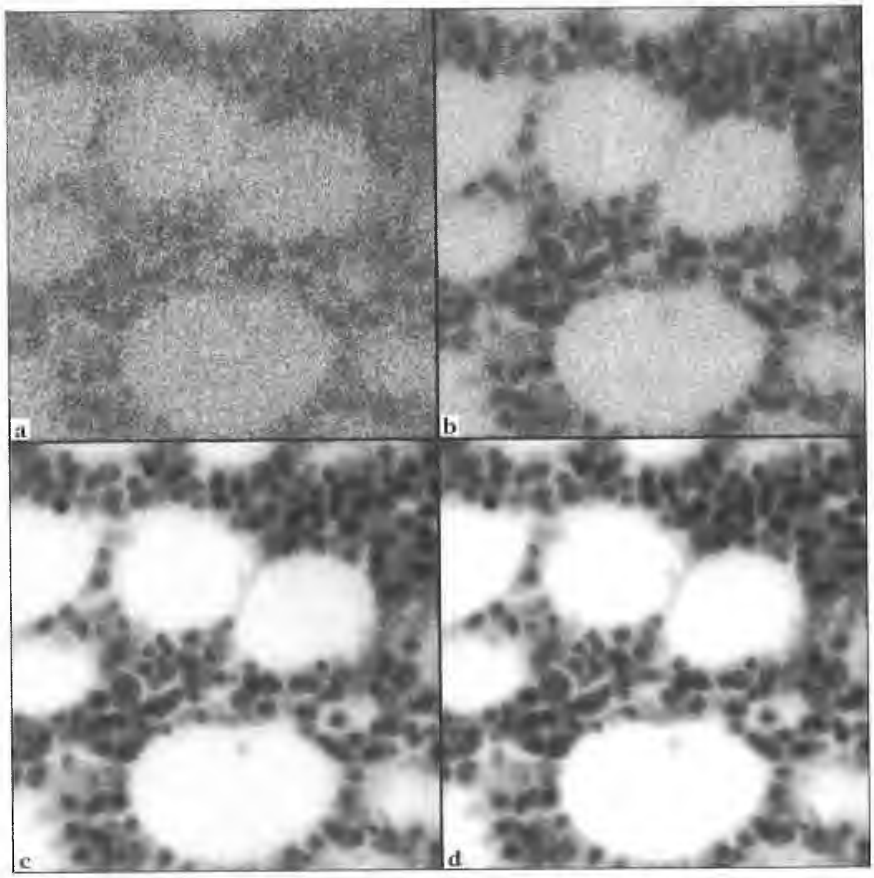


***Figure 28. Averaging of a noisy (low photon intensity) image (light microscope image of bone marrow)**: one frame (**a**); addition of 4 (**b**), 16 (**c**) and 256 (**d**) frames.*

specialized hardware, and performing the operation in a general-purpose computer limits the practical acquisition to only a few of the video frames per second. This limitation discards a large percentage of the photons that reach the detector. It is more efficient to use a camera capable of integrating the signal directly for the appropriate length of time, and then to read the final image to the computer, which also reduces the noise due to readout and digitization. Digital cameras are available with this capability, and with cooled chips can reduce electronic noise during long acquisitions of many minutes. Most uncooled camera chips will begin to show unacceptable pixel noise due to dark current with integration times of more than a few seconds.

Acquiring images from a video camera is sometimes referred to as "real-time" imaging, but of course this term should properly be reserved for any imaging rate that is adequate to reveal temporal changes in a particular application. For some situations, time-lapse photography may only require one frame to be taken at periods of many minutes. For others, very short exposures and high rates are needed. Special cameras that do not use video frame rates or bandwidths can achieve rates up to ten times that of a standard video camera for full frames, and even higher for small image dimensions. These cameras typically use a single line of detectors and optical deflection (e.g., a rotating mirror or prism) to cover the image.

For many applications, the repetition rate does not need to be that high. Either stroboscopic imaging or simply a fast shutter speed may be enough to stop the important motion to provide a sharp

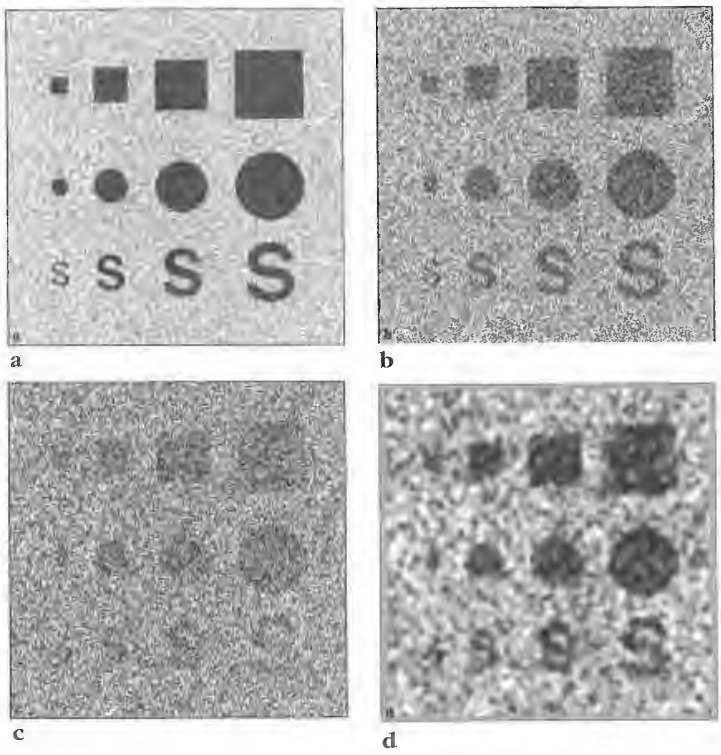


Figure 29. *Features on a noisy background:*
*(**a**) signal to noise ratio 1:1;*
*(**b**) signal to noise ratio 1:3;*
*(**c**) signal to noise ratio 1:7;*
*(**d**) image **c** after spatial smoothing.*

image. Electronic shutters can be used to control solid state imaging devices, instead of a mechanical shutter. Exposure times under 1/1000th of a second can easily be achieved, but of course this short exposure requires plenty of light.

High depth images

Other devices that produce data sets that are often treated as images for viewing and measurement produce data with a much greater range than a camera. For instance, a scanned stylus instrument that measures the elevation of points on a surface may have a vertical resolution of a few nanometers with a maximum vertical travel of hundreds of micrometers, for a range-to-resolution value of 10^5. This would require storing data in a format that preserved the full resolution values, and such instruments typically use 4 bytes per pixel.

In some cases with cameras having a large brightness range, the entire 12- or 14-bit depth of each pixel is stored; however, because this depth exceeds the capabilities of most CRTs to display, or of the user to see, reduction may be appropriate. If the actual brightness range of the image does not cover the entire possible range, scaling (either manual or automatic) to select just the range actually used can significantly reduce storage requirements. In other cases, especially when performing densitometry, a nonlinear conversion table is used. For densitometry, the desired density value varies as the logarithm of the brightness; this is discussed in detail in Chapter 9. A range of 256 brightness steps is not adequate to cover a typical range from 0 to greater than 3 in optical density with useful precision, because at the dark end of the range, 1 part in 256 represents a step of more than 0.1 in optical density. Using a digitization with 12 bits (1 part in 4096) solves this problem, but it is efficient to convert the resulting value with a logarithmic lookup table to store an 8-bit value (occupying a single computer byte) that is the optical density.

Lookup tables (LUTs) may be implemented either in hardware or software. They simply use the original value as an index into a stored or precalculated table, which then provides the derived

value. This process is fast enough that acquisition is not affected. The context for LUTs discussed here is for image acquisition, converting a 10-, 12-, or 14-bit digitized value with a nonlinear table to an 8-bit value that can be stored. LUTs are also used for displaying stored images, particularly to substitute colors for grey scale values to create pseudo-color displays. This topic is discussed later in this chapter.

Many images do not have a brightness range that covers the full dynamic range of the digitizer. The result is an image whose histogram covers only a portion of the available values for storage or for display. **Figure 30** shows a histogram of such an image. The flat (empty) regions of the plot indicate brightness values at both the light and dark ends that are not used by any of the pixels in the image. Expanding the brightness scale by spreading the histogram out to the full available range, as shown in the figure, may improve the visibility of features and the perceived contrast in local structures. The same number of brightness values are missing from the image, as shown by the gaps in the histogram, but now they are spread uniformly throughout the range. Other ways to stretch the histogram nonlinearly are discussed in Chapter 3.

Because the contrast range of many astronomical images is too great for photographic printing, special darkroom techniques have been developed. "Unsharp masking" (**Figure 31**) increases the ability to show local contrast by suppressing the overall brightness range of the image. The suppression is done by first printing a "mask" image, slightly out of focus, onto another negative. This negative is developed and then placed on the original to make the final print. This superposition reduces the exposure in the bright areas so that the detail can be shown. The same method can also be used in digital image processing, either by subtracting a smoothed version of the image or by using a Laplacian operator (both are discussed in Chapter 3 on spatial domain processing). When the entire depth of a 12- or 14-bit image is stored, such processing may be needed in order to display the image for viewing on a CRT.

Some perspective on camera performance levels is needed. Although a standard video camera has about 300,000 sensors, and a high performance digital camera may have a few million, the human eye has about 1.5×10^8. Furthermore, these are clustered particularly tightly in the fovea, the central area where we concentrate our attention. It is true that only a few dozen brightness levels can be distinguished in a single field, but the eye adjusts automatically to overall brightness levels covering nine orders of magnitude to select the optimal range (although color sensitivity is lost in the darkest part of this range).

Some cameras use a single-line CCD instead of a two-dimensional array. This gives high resolution but requires physically scanning the line across the film plane, much like a flat-bed scanner. Most of these systems store the image digitally, converting the signal to a full-color image (for instance with 8 to 12 bits each of red, green, and blue data).

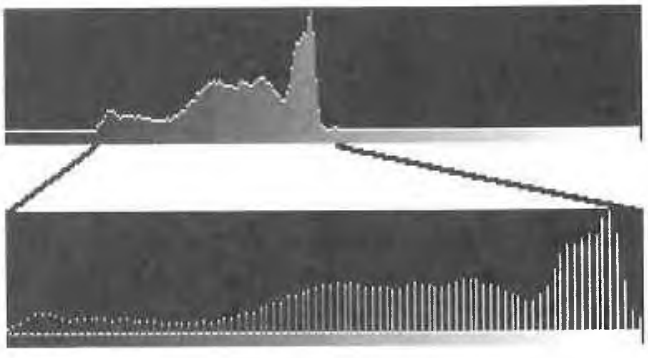
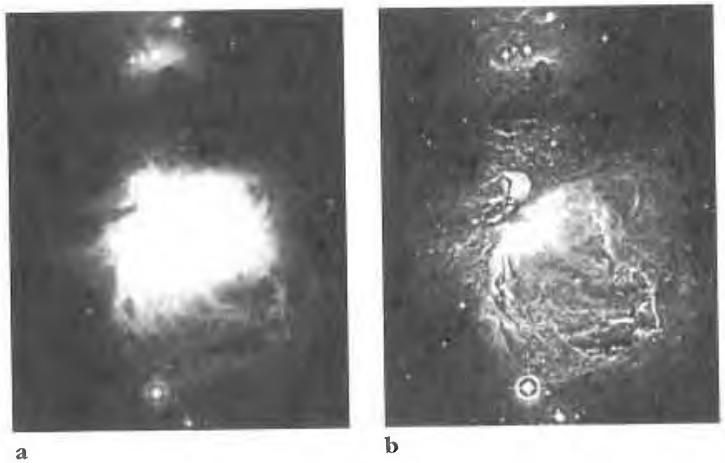


Figure 30. *Linear expansion of a histogram to cover the full range of storage and/or display.*



a b

Figure 31. Unsharp masking: *(a) a telescope image originally recorded on film of the Orion nebula; (b) the same image using unsharp masking. An out-of-focus photographic print is made onto negative material, which is then placed on the original to reduce the exposure in bright areas when the final print is made (adapted from a 1920s photographic darkroom manual). This reduces the overall contrast so that local variations become visible.*

There is a major difference between the interlace scan used in conventional television and a non-interlaced or "progressive" scan. The latter gives better quality because there are no line-to-line alignment or shift problems. Most high definition television (HDTV) proposals use progressive scan. The format requires a higher rate of repetition of frames to fool the human eye into seeing continuous motion without flicker, but it has many other advantages. These include simpler logic to read data from the camera (which may be incorporated directly on the chip), more opportunity for data compression because of redundancies between successive lines, and simpler display or storage devices. Most scientific imaging systems such as digital cameras, direct-scan microscopes (i.e., the scanning electron microscope (SEM), scanning tunneling microscope (STM), the atomic force microscope (AFM), etc.), flat-bed scanners, film or slide digitizers, and similar devices, use progressive rather than interlaced scan.

HDTV proposals include many more differences from conventional television than the use of progressive scan. The pixel density is much higher, with a wider aspect ratio of 16:9 (instead of the 4:3 used in NTSC television) and the pixels are square. A typical HDTV proposal presents 1920 × 1080 pixel images at the rate of 30 full scans per second, for a total data rate exceeding 1 gigabit per second, which is several hundred times as much data as current broadcast television. One consequence of this high data rate is the interest in data compression techniques, discussed in the next chapter, and the investigation of digital transmission techniques, perhaps using cable or optical fiber instead of broadcast channels. Whatever the details of the outcome in terms of consumer television, the development of HDTV hardware is likely to produce spin-off effects for computer imaging, such as high pixel density cameras with progressive scan output, high bandwidth recording devices, and superior CRT displays. For example, color cameras being designed for HDTV applications output digital rather than analog information by performing the analog-to-digital conversion within the camera, with at least 10 bits each for red, green, and blue.

It is also interesting to compare camera technology to other kinds of image acquisition devices. The scanning electron (SEM) or scanning tunneling microscope (STM) typically use from a few hundred to about 1000 scan lines. Those that digitize the signals use 8 or sometimes 12 bits, and so are similar in image resolution and size to many camera systems. **Figure 32** shows schematically the function of an SEM. The focused beam of electrons is scanned over the sample surface in a raster pattern while various signals generated by the electrons are detected. These include secondary and backscattered electrons, characteristic x-rays, visible light, and electronic effects in the sample.

Other point-scanning microscopes, such as the STM, the confocal scanning light microscope (CSLM), and even contact profilometers, produce very different signals and information. All provide

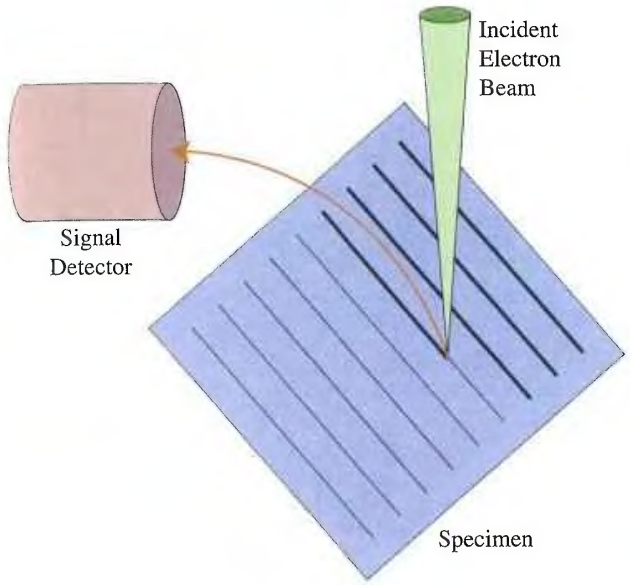


Figure 32. *The scanning electron microscope (SEM) focuses a fine beam of electrons on the specimen, producing various signals which may be used for imaging as the beam is scanned in a raster pattern.*

a time-varying signal that is related to spatial locations on the sample by knowing the scanning speed and parameters, which allows storing the data as an image.

Significantly larger arrays of pixels are available from flat-bed scanners. These devices use a linear solid-state detector array, and can typically scan areas at least 8 inches by 10 inches, and sometimes up to four times that size. Although primarily intended for the desktop publishing market, they are readily adapted to scan electrophoresis gels used for protein separation, or photographic prints or negatives. A high-quality negative can record more than a thousand distinguishable brightness levels and several thousand points per inch (both values are much better than prints). Scanners are also used for digitizing photographic negatives such as medical x-rays and 35-mm slides. Commercial scanners are used in publishing and to convert films to digital values for storage (for instance in Kodak's Photo-CD format). At the consumer level, scanners with 300 to as many as 1200 pixels per inch are common for large-area reflective originals, and 2500 to 3500 pixels per inch for 35-mm slide film. Most digitize full-color red, green, and blue (RGB) images.

These devices are quite inexpensive, but rather slow, taking tens of seconds to digitize the scan area. Another characteristic problem they exhibit is pattern noise in the sensors: If all of the detectors along the row are not identical in performance, a "striping" effect will appear in the image as the sensor is scanned across the picture. If the scan motion is not perfectly smooth, it can produce striping in the other direction. This latter effect is particularly troublesome with hand-held scanners that rely on the user to move them and sense the motion with a contact roller that can skip, but these devices should not be used for inputting images for measurement anyway (they are marketed for reading text), because of the tendency to skip or twist in motion.

By far the greatest difficulty with such scanners arises from the illumination. There is often a drop-off in intensity near the edges of the scanned field because of the short length of the light source. Even more troublesome is the warm-up time for the light source to become stable. In scanners that digitize color images in a single pass by turning colored lights on and off, stability and consistent color is especially hard to obtain. On the other hand, making three separate passes for red, green, and blue often causes registration difficulties. Even for monochrome scanning, it may take several repeated scans before the light source and other electronics become stable enough for consistent measurements.

Color imaging

Most real-world images are not monochrome, of course, but full color. The light microscope produces color images, and many tissue specimen preparation techniques make use of color to identify structure or localize chemical activity in novel ways. Even for inorganic materials, the use of polarized light or surface oxidation produces color images to delineate structure. The SEM would seem to be a strictly monochromatic imaging tool, but color may be introduced based on x-ray energy or backscattered electron energy. **Figure 33** shows individual grey scale images from the x-ray signals from a mineral sample imaged in the SEM. The x-ray images show the variation in composition of the sample for each of nine elements (Al, Ca, Fe, K, Mg, Na, O, Si, Ti). More individual images are available than just the three red, green, and blue display channels, with no obvious "correct" choice for assigning colors to elements. **Figure 34** shows a few possibilities, but it is important to keep in mind that no single color image can show all of the elements at once. Assigning several of these arbitrarily (e.g., to the red, green, and blue planes of a display), may aid the user in judging the alignment of regions and areas containing two or more elements. Colored x-ray maps are now fairly common with the SEM, as are similar concentration maps based on measured intensities from ion microprobes, but other uses of color such as energy loss in the transmission electron microscope (TEM) are still experimental.

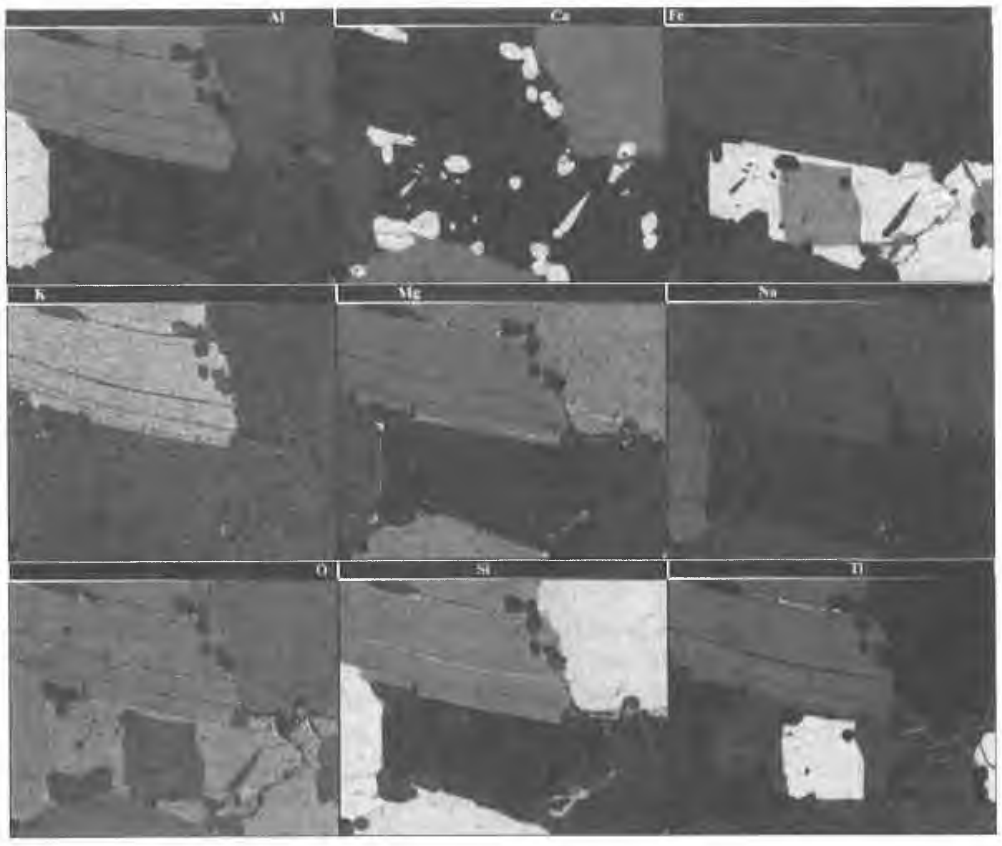


Figure 33. *Scanning electron microscope images of a polished sample of mica. The individual X-ray images or "dot maps" show the localization of the corresponding elements within various minerals in the sample.*

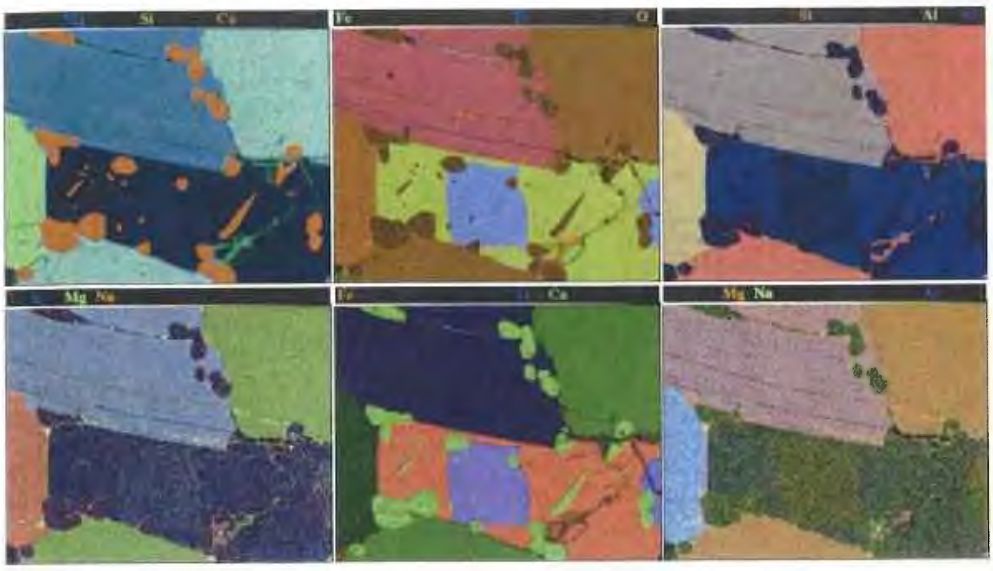


Figure 34. *Composite color images made by assigning various images from* ***Figure 33*** *to the red, green, or blue channels of the display. This assists in the identification and arrangement of the various minerals.*

Similar use of color is potentially useful with other kinds of microscopes, although, in many cases, these possibilities have not been exploited in commercial instruments. This is also true for macroscopic imaging tools. A simple example is the use of color to show altitude in air traffic control displays. This use of color increases the bandwidth for communicating multidimensional information to the user, but the effective use of these methods will require some education of users, and would benefit from some standardization.

The use of color to encode richly multidimensional information must be distinguished from the very common use of false-color or pseudo-color to substitute colors for brightness in a monochrome image. Pseudo-color is used because of the limitation mentioned before in our visual ability to distinguish subtle differences in brightness. Although we can only distinguish about 20–30 shades of grey in a monochrome image, we can distinguish hundreds of different colors. Also, it is may aid communication to describe a particular feature of interest as "the dark reddish-blue one" instead of "the medium grey one."

The use of color scales as a substitute for brightness values allows us to show and see small changes locally, and identify the same brightness values globally in an image. This should be a great benefit, since these are among the goals for imaging discussed below. Pseudo-color has been used particularly for many of the images returned from space probes. It would be interesting to know how many people think that the rings around Saturn really are brightly colored, or that Halley's Comet really is surrounded by a rainbow-colored halo. The danger in the use of pseudo-color is that it can obscure the real contents of an image. The colors force us to concentrate on the details of the image, and to lose the gestalt information. Examples of image processing in this book will use pseudo-color selectively to illustrate some of the processing effects and the changes in pixel values that are produced, but often pseudo-color distracts the human eye from seeing the real contents of the enhanced image.

Pseudo-color displays as used in this context simply substitute a color from a stored or precalculated table for each discrete stored brightness value. As shown in **Figure 35**, this should be distinguished from some other uses of color displays to identify structures or indicate feature

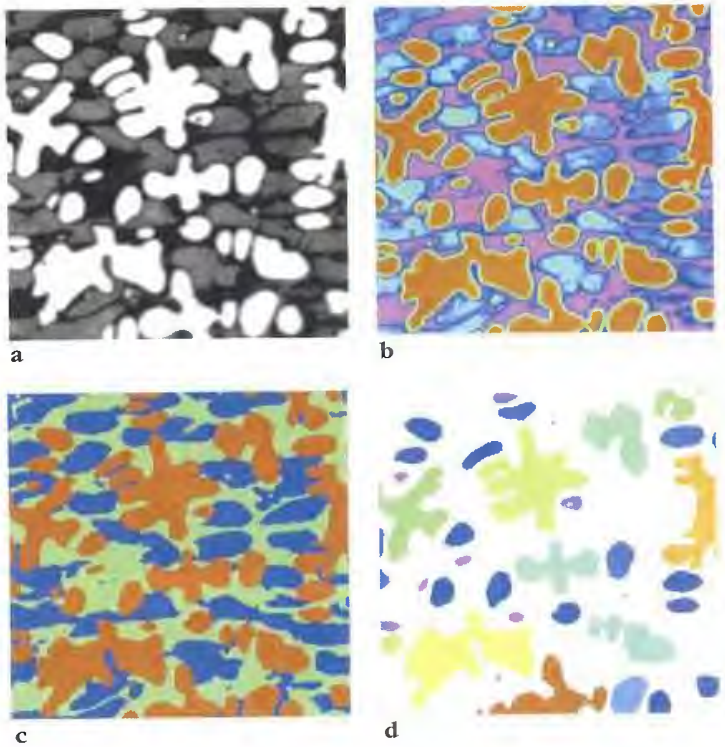


Figure 35. Different uses for pseudo-color displays: *(**a**) portion of a grey-scale microscope image of a polished metallographic specimen with three phases having different average brightnesses; (**b**) image **a** with pseudo-color palette or LUT which replaces grey values with colors (note the misleading colors along boundaries between light and dark phases); (**c**) image **a** with colors assigned to phases (this requires segmentation of the image by thresholding and other logic to assign each pixel to a phase based on grey-scale brightness and neighboring pixel classification); (**d**) lightest features from image **a** with colors assigned based on feature size (this requires the steps to create image **c**, plus collection of all touching pixels into features and the measurement of the features.*

properties. These also rely on the use of color to communicate rich information to the viewer, but require considerable processing and measurement of the image before this information becomes available.

Color can be used to encode elevation of surfaces (see Chapter 13). In scientific visualization it is used for velocity, density, temperature, composition, and many less obvious properties. These uses generally have little to do with the properties of the image and simply take advantage of the human ability to distinguish more colors than grey-scale values.

Most computer-based imaging systems make it easy to substitute various lookup tables (LUTs) of colors for the brightness values in a stored image. These work in the same way as input lookup tables, described earlier. The stored grey scale value is used to select a set of red, green, and blue brightnesses in the LUT that control the voltages sent to the display tube. Many systems also provide utilities for creating tables of these colors, but few guidelines exist to assist in constructing useful ones. All we can do here is advise caution. One approach is to systematically and gradually vary color along a path through color space. Examples (**Figure 36**) are a rainbow spectrum of colors or a progression from brown through red and yellow to white, the so-called heat scale. This gradual variation can help to organize the different parts of the scene. Another approach is to rapidly shift colors, for instance by varying the hue sinusoidally. This enhances gradients and makes it easy to see local variations, but may completely hide the overall contents of some images (**Figure 37**).

Some image sources may use color to encode a variety of different kinds of information, such as the intensity and polarization of radio waves in astronomy; however, by far the most common type of color image is that produced by recording the intensity at three different wavelengths of visible light. Video deserves consideration as a suitable medium for this type of image, since standard

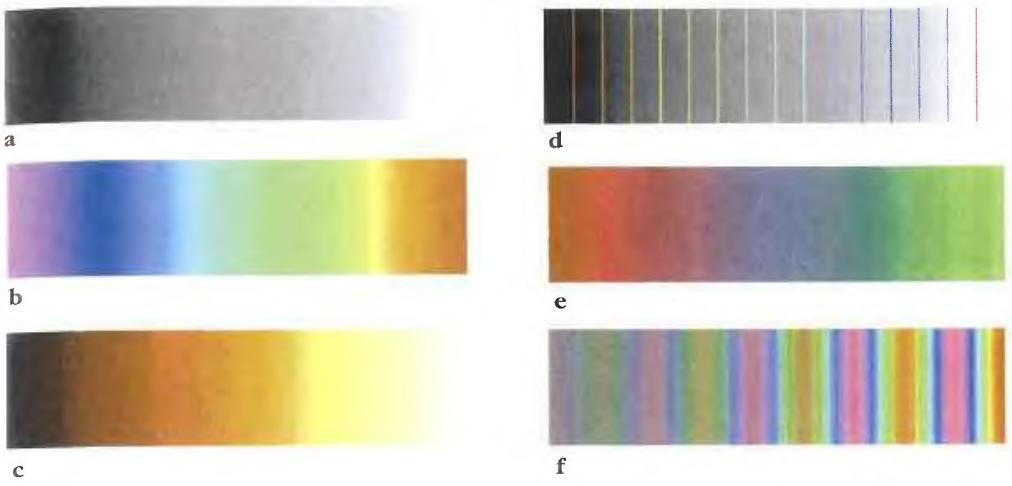


Figure 36. Six examples of display look-up tables (LUTs). *(**a**) monochrome (grey-scale); (**b**) spectrum or rainbow (variation of hue, with maximum saturation and constant intensity); (**c**) heat scale; (**d**) monochrome with contour lines (rainbow colors substituted every 16th value); (**e**) tri-color blend of three primary colors; f) sinusoidal variation of hue with linear variation of saturation and intensity.*

broadcast television uses color in a way that is visually satisfactory. The National Television Systems Committee (NTSC) color encoding scheme used in the U.S. was developed as a compatible add-on to existing monochrome television broadcasts. It adds the color information within the same, already narrow bandwidth limitation. The result is that the color has even less lateral resolution than the brightness information.

Color subsampling reduces the amount of data in an image by representing color data with lower resolution than luminance (brightness) data. This is done in the YUV color space (Y = luminance or brightness, U and V = chrominance or color) described later in this chapter. Uncompressed YUV color is represented as 4:4:4. The common subsampling options are 4:2:2, 4:2:0, 4:1:1, and YUV-9:

4:2:2. — full-bandwidth sampling of Y and 2:1 horizontal sampling of U and V. This is the sampling scheme most commonly used in professional and broadcast video and in tape formats such as D-1 and Digital Betacam. It looks good, but the data compression ratio is only 1.5:1.

4:2:0. — full-bandwidth sampling of Y and 2:1 sampling of U and V in both the horizontal and vertical dimensions. That is, for every four luminance samples, there are two chrominance samples every other line. This yields a 2:1 reduction in data. 4:2:0 is the color space used in MPEG compression.

4:1:1. — full-bandwidth sampling of Y and 4:1 horizontal sampling of U and V. This is the color space of the digital video (DV) formats. It uses U and V samples four pixels wide and one pixel tall, so color bleeding is much worse in the horizontal than in the vertical direction.

YUV-9. — this is the color format used in most of the video compression on the internet. For every 16 luminance Y samples in a 4 × 4-pixel block, there is only one U and one V sample, producing smaller files with correspondingly lower color fidelity. YUV-9 subsampling often results in noticeable color artifacts around the edges of brightly colored objects, especially red.

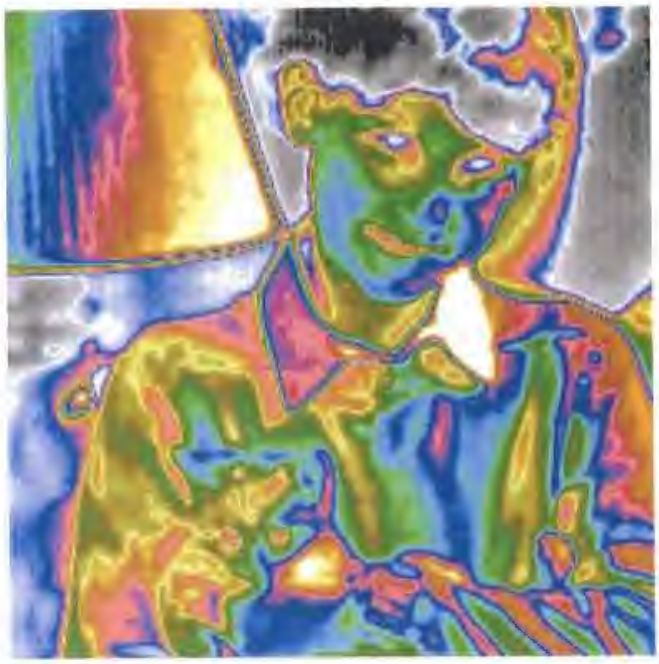


Figure 37. *The same image used in* ***Figures 25*** *and* ***26*** *with a pseudocolor display LUT. The gestalt contents of the image are obscured.*

Such limitations are acceptable for television pictures because the viewer tolerates colors that are less sharply bounded and uses the edges of features defined by the brightness component of the image where they do not exactly correspond. The same tolerance has been used effectively by painters, and may be familiar to parents whose young children have not yet learned to color "inside the lines." **Figure 38** shows an example in which the bleeding of color across boundaries or variations within regions is not confusing to the eye.

The poor spatial sharpness of NTSC color is matched by its poor consistency in representing the actual color values (a common joke is that NTSC means "Never The Same Color" instead of "National Television Systems Committee"). Videotape recordings of color images are even less



Figure 38. *A child's painting of a clown. Notice that the colors are unrealistic, but their relative intensities are correct and the painted areas are not exactly bounded by the lines. The dark lines nevertheless give the dimensions and shape to the features, and we are not confused by the colors which extend beyond their regions.*

useful for analysis than monochrome ones, but the limitations imposed by the broadcast channel do not necessarily mean that the cameras and other components may not be useful. An improvement in the sharpness of the color information in these images is afforded by Super-VHS or S-video recording equipment, also called component video, in which the brightness or luminance and color or chrominance information are transmitted and recorded separately.

Some color cameras intended for technical purposes bring out the red, green, and blue signals separately so that they can be individually digitized. Recording the image in computer memory then simply involves treating each signal as a monochrome one, converting it to a set of numbers, and storing it in memory. If the signals have first been combined, the encoding scheme used is likely to be YIQ or YUV (defined in the "color spaces" section), which are closely related to the NTSC broadcasting scheme. Much better fidelity in the image can be preserved by not mixing together the color and brightness information. Instead of the composite signal carried on a single wire, some cameras and recorders separate the chrominance (color) and luminance (brightness) signals onto separate wires. This so-called component, Y-C or S-video format is used for high-end consumer camcorders (Hi-8 and S-VHS formats). Many computer interfaces accept this format, which gives significant improvement in the quality of digitized images, as shown in **Figure 39**.

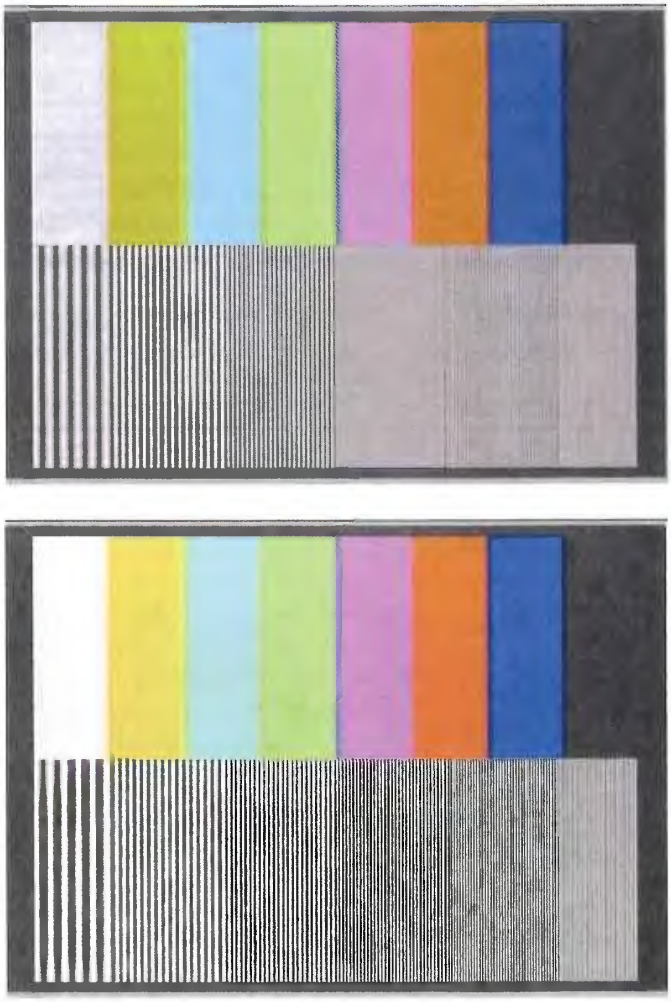


Figure 39. *Comparison of* ***(a)*** *composite and* ***(b)*** *component video signals from a signal generator, digitized using the same interface board. Note the differences in resolution of the black and white stripes and the boundaries between different colors.*

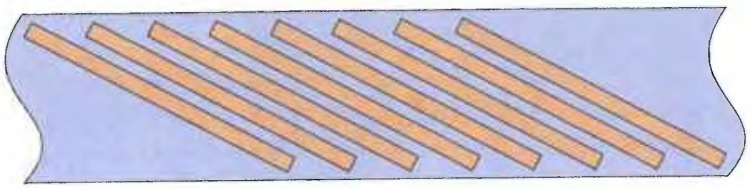


Figure 40. *In digital video, each video frame is recorded digitally in ten tracks on metal-coated magnetic tape (12 tracks for the larger PAL format). Each track contains video, audio, and timing information.*

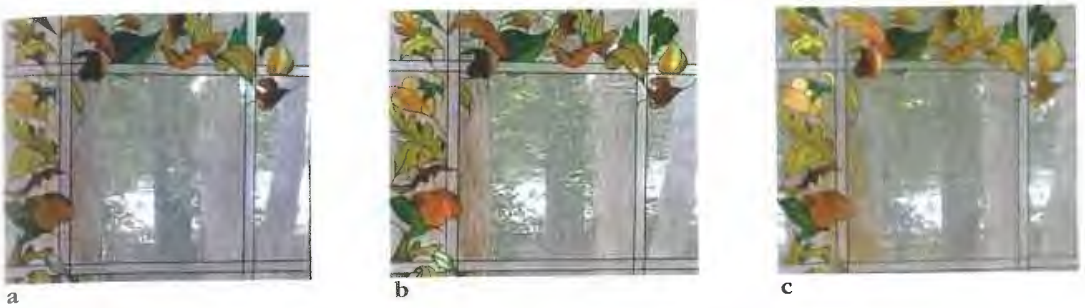
Another development in video that is important is digital video (DV) recording. Similar to analog video tape recorders, digital video writes each scan line onto the tape at an angle using a moving head that rotates as the tape moves past it (**Figure 40**); however, the signal is encoded as a series of digital bits that offers several advantages. Just as CD technology replaced analog audio tapes, the digital video signal is not subject to loss of fidelity as images are transmitted or copies are made. More important, the high frequency information that is discarded in analog recording because of the 4 MHz upper frequency limit imposed by conventional broadcast TV is preserved in DV. Digital video records up to 13.5 MHz for the luminance (brightness) information, and up to one-fourth that for the chrominance (color) information. This produces much sharper edge delineation at least for the brightness data in the image, and greatly improves the usefulness of the resulting images. The digital video signal is fully preserved on the tape recording, unlike conventional analog recorders which impose a further bandwidth reduction on the data, and are susceptible to quality loss due to dirty heads, worn tape, or making successive copies from an original.

The result is that DV images have about 500 × 500 pixel resolution and nearly 8 bits of contrast, and can be read into the computer without a separate digitizer board since they are already digital in format. The IEEE 1394 standard protocol for digital video (also known as firewire) establishes a standard serial interface convention that is available on consumer-priced cameras and decks, and is being supported by many computer manufacturers as well. Inexpensive interface boards are available, and the capability is built into the basic circuitry of some personal computers, even laptops. Most of these digital cameras can be controlled by the computer to select individual frames for transfer. They can also be used to record single frames annotated with date and time, turning the digital video tape cartridge into a tiny but high-fidelity storage format for hundreds of single images.

Digital camera limitations

The current development of the digital still-frame cameras for the consumer market involves practically every traditional maker of film cameras and photographic supplies. Many of these are unsuitable for technical applications because of limitations in the optics (fixed focus lenses with geometric distortions), and limited resolution (although even a low-end 320 × 240 pixel camera chip is as good as conventional analog video recording). Some cameras interpolate between the pixels on the chip to create images with empty magnification. In most cases this produces artefacts in the image that are serious impediments to quantitative measurement.

The use of image compression creates the most important problem. In an effort to pack many images into the smallest possible memory, JPEG and other forms of image compression are used. As discussed in Chapter 2, these are lossy techniques that discard information from the image. The discarded data is selected to minimally impact the human interpretation and recognition of familiar images — snapshots of the kids and the summer vacation — but the effect on quantitative image analysis can be severe. Edges are broken up and shifted, color and density values are altered, and



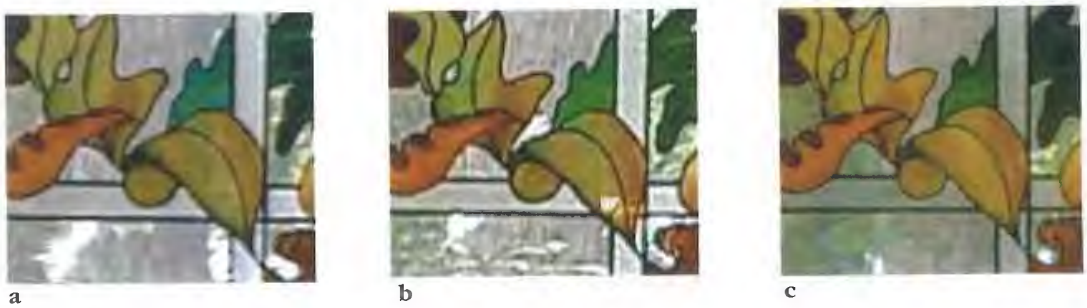
a b c

Figure 41. Images of the same scene (see text) recorded using: *(**a**) a 3-chip video camera and analog to digital converter; (**b**) an inexpensive consumer digital camera; and (**c**) a research-grade digital camera.*

fine details can be eliminated or moved. The important advantage of the higher-end consumer and professional cameras is that they offer storage or transfer to the computer without any lossy compression, and this is much preferred for technical applications.

Figure 41 compares images of the same scene using three different cameras. The subject is a difficult view through a stained glass window, with a dark interior and bright daytime exterior. This produces a very high contrast range, regions that are saturated in color and others with low saturation, subtle differences and fine detail. The first image is video from a 3-chip color video camera acquired using a high-quality analog to digital converter producing a 640 × 480 pixel image and averaging eight frames. The second image was acquired with a consumer digital camera, also as a 640 × 480 pixel image, using the highest quality compression setting (least compression). The third image was acquired with a research-grade digital camera as a 1600 × 1200 pixel image.

The original colors are represented with greater fidelity in the third image. In addition to color shifts, the first two images have brightness values that are clipped to eliminate the brightest and darkest values, and the video image does not produce 256 distinct values of red, green, and blue so that the image has missing values within the range. The consumer digital camera apparently applies a hi-pass or sharpening filter that exaggerates contrast at edges producing dark and light borders. Other artefacts are present in the first two images as well: scan line noise in the video image and square blocks resulting from the JPEG compression in the consumer digital camera image. Enlarging the image to see fine details makes apparent the far higher resolution of the research grade digital image. In **Figure 42** a region is expanded to show fine details in each of the images. The research grade digital camera image renders these with greater fidelity in both the high-brightness and high-saturation regions.



a b c

Figure 42. *Enlarged regions of the images in **Figure 41,** showing the greater resolution and fidelity in the high-end digital camera image.*



Figure 43. *Macro images using the same three cameras as in* ***Figures 41*** *and* ***42****. Note the variations in resolution and color rendition.*

The high-resolution digital camera offers the same advantages for recording of images in microscope and other technical applications. **Figure 43** shows three images obtained with the same three cameras using a low power microscope to view an antique cross-stitch fabric specimen. Identification of individual characteristics (and occasional errors) in the formation of the stitches and tension on the threads is of interest to cultural anthropologists to identify individual handiwork. This requires a low enough magnification to see the overall pattern, while identification of the threads and dyes used requires a high enough resolution to see individual fibers, and good color fidelity to represent the dyes. **Figure 44** shows an enlargement of one small region of the three images. The differences in resolution and color fidelity between these images is another demonstration of the importance of using a high-quality camera.

A few words of caution may be useful. The slower readout from digital cameras makes focusing more difficult: the camera produces a preview image that updates a few times a second. This is fine for selecting fields of view, but for convenient focusing it is important to adjust the optics so that the camera is truly parfocal with the eyepiece image. Some digital cameras also put out a live video signal that can be connected to a monitor for previews, albeit with only video resolution. Some digital cameras do not have a useful preview image but instead rely on a viewfinder that is of little practical use. Also the area of the image sensing chip is smaller than film but larger than many video cameras so that standard transfer optics may not provide complete coverage; clipping or vignetting can result.

On the other hand, it is possible in some instances to use very low-end cameras to acquire useful images. For instance, digital cameras intended for video conferencing are available that deliver 64



Figure 44. *Enlargement of a portion of the images in* ***Figure 43****, showing the appearance of individual fibers.*

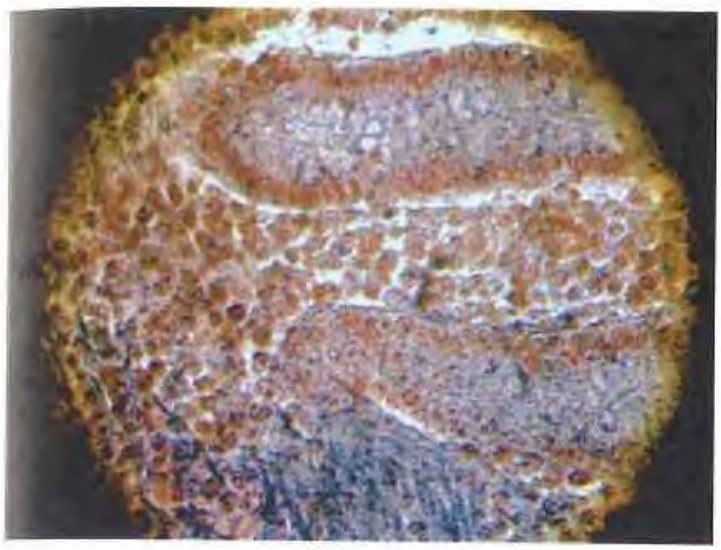


Figure 45. *View of a biological thin section using a video conferencing camera placed on the eyepiece of a bench microscope. Despite significant vignetting, the main details of the slide are visible.*

Figure 46. *Using the same camera as in* ***Figure 45*** *with a loupe to record an image for forensic examination.*



grey levels or 32 levels of red, green, and blue with 320 × 240 pixel resolution (about as good as many examples of digitized video) at about 10 frames per second via a simple serial or parallel interface to the computer. No digitizer or separate power supply is required, and the digitization is built into the camera chip requiring no hardware to be added to the computer. Such cameras cost well under $100 and can be quite useful for purposes such as capturing an image for videoconferencing.

The illustration in **Figure 45** was obtained by holding one such camera directly onto the eyepiece of a microscope with duct tape. The automatic gain control in the software was turned off because the black area around the image confused the automatic adjustment. **Figure 46** shows another example of the use of the same camera. Placing it on an 8X loupe allows direct capture of images when (as in this forensic case) it was essential to collect images and nothing else except the camera and a laptop computer was available. The biggest flaws with this setup were the difficulty in controlling the specimen illumination and the distortion caused by the very wide angle view from the optics.

Color spaces

Conversion from RGB (the brightness of the individual red, green, and blue signals at defined wavelengths, as captured by the camera and stored in the computer) to YIQ/YUV and to the other color encoding schemes is straightforward and loses no information except for possible round-off errors. Y, the "luminance" signal, is just the brightness of a panchromatic monochrome image that would be displayed by a black-and-white television receiver. It combines the red, green, and blue signals in proportion to the human eye's sensitivity to them. The I and Q (or U and V) components of the color signal are chosen for compatibility with the hardware used in broadcasting; the I signal is essentially red minus cyan, while Q is magenta minus green. The relationship between YIQ and RGB is shown in **Table 1**. An inverse conversion from the encoded YIQ signal to RGB simply requires inverting the matrix of values.

RGB (and the complementary CMY subtractive primary colors used for printing) and YIQ are both hardware-oriented schemes. RGB comes from the way camera sensors and display phosphors work, while YIQ or YUV stem from broadcast considerations. **Figure 47** shows the "space" defined by RGB signals: it is a Cartesian cubic space because the red, green, and blue signals are independent and can be added to produce any color within the cube. Other encoding schemes are available, which are more useful for image processing and are more closely related to human perception.

Table 1. Interconversion of RGB and YIQ color scales

Y = 0.299 R + 0.587 G + 0.114 B	R = 1.000 Y + 0.956 I + 0.621 Q
I = 0.596 R − 0.274 G − 0.322 B	G = 1.000 Y − 0.272 I − 0.647 Q
Q = 0.211 R − 0.523 G + 0.312 B	B = 1.000 Y − 1.106 I + 1.703 Q

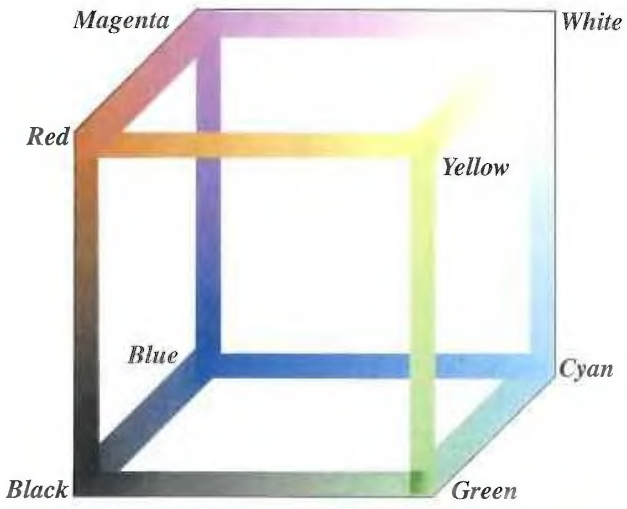


Figure 47. *RGB Color Space, showing the additive progression from Black to White. Combining Red and Green produces Yellow, Green plus Blue produces Cyan, and Blue plus Red produces Magenta. Greys lie along the cube diagonal with equal proportions of Red, Green, and Blue. Cyan, Yellow, and Magenta are subtractive primaries used in printing, which, if subtracted from White, leave Red, Blue, and Green, respectively.*

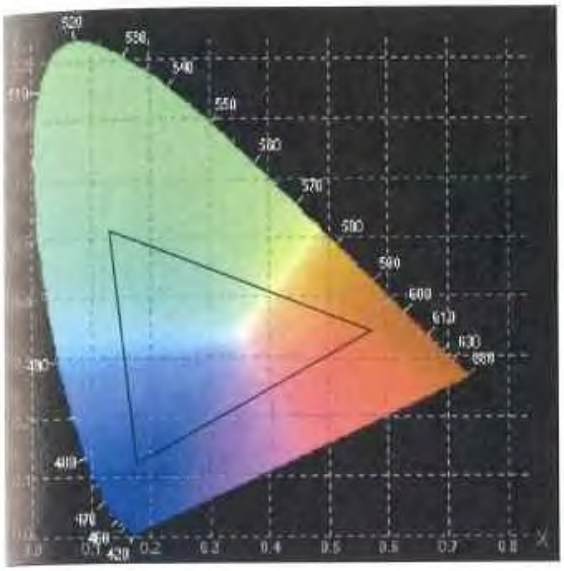


Figure 48. *The CIE chromaticity diagram. The colors are fully saturated along the edge. Numbers give the wavelength of light in nanometers. The inscribed triangle shows the colors that typical color CRTs can produce by mixing of red, green and blue light from phosphors.*

The oldest of these is the CIE (Commission Internationale de L'Éclairage) chromaticity diagram. This is a two-dimensional plot defining color, shown in **Figure 48**. The third axis is the luminance, which corresponds to the panchromatic brightness which, like the Y value in YUV, would produce a monochrome (grey-scale) image. The other two primaries, called x and y are always positive (unlike the U and V values) and combine to define any color that we can see.

Instruments for color measurement utilize the CIE primaries, which define the dominant wavelength and purity of any color. Mixing any two colors corresponds to selecting a new point in the diagram along a straight line between the two original colors. This means that a triangle on the CIE diagram with its corners at the red, green, and blue locations of emission phosphors used in a cathode ray tube (CRT) defines all of the colors that the tube can display. Some colors cannot be created by mixing these three phosphor colors, shown by the fact that they lie outside the triangle. The range of possible colors for any display or other output device is the gamut; hardcopy printers generally have a much smaller gamut than display tubes as discussed in Chapter 2. The edge of the bounded region in the diagram corresponds to pure colors, and is marked with the wavelength in nanometers.

Complementary colors are shown in the CIE diagram by drawing a line through the central point, which corresponds to white light. Thus, a line from green passes through white to magenta. One of the drawbacks of the CIE diagram is that it does not indicate the variation in color that can be discerned by eye. Sometimes this is shown by plotting a series of ellipses on the diagram. These are much larger in the green area, where small changes are poorly perceived, than elsewhere.

The CIE diagram provides a tool for color definition, but corresponds neither to the operation of hardware nor directly to human vision. An approach that does is embodied in the HSV (hue, saturation, and value), HSI (hue, saturation, and intensity) and HLS (hue, lightness, and saturation) systems. These are closely related to each other and to the artist's concept of tint, shade, and tone. In this system, hue is the color as described by wavelength, for instance the distinction between red and yellow. Saturation is the amount of the color that is present, for instance the distinction between red and pink. The third axis (called lightness, intensity, or value) is the amount of light, the distinction between a dark red and light red, or between dark grey and light grey.

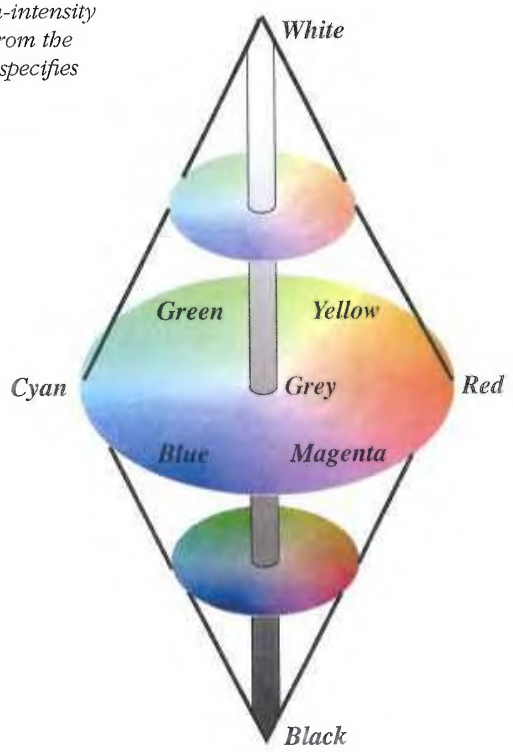


Figure 49. *Bi-conic representation of hue-saturation-intensity space. Greys lie along the central axis. Distance from the axis gives the Saturation, while direction (angle) specifies the hue.*

The space in which these three values is plotted can be shown as a circular or hexagonal cone or double cone, or sometimes as a cylinder. It is useful to imagine the space as a double cone, in which the axis of the cone is the grey scale progression from black to white, distance from the central axis is the saturation, and the direction is the hue. **Figure 49** shows this concept schematically.

This space has many advantages for image processing. For instance, if the algorithms discussed in Chapter 3 such as spatial smoothing or median filtering are used to reduce noise in an image, applying them to the RGB signals separately will cause color shifts in the result, but applying them to the HSI components will not. Also, the use of hue (in particular) for distinguishing features in the process called segmentation (Chapter 6) often corresponds to human perception and ignores shading effects. On the other hand, because the HSI components do not correspond to the way that most hardware works (either for acquisition or display), it requires computation to convert RGB-encoded images to HSI and back.

The HSI spaces are useful for image processing because they separate the color information in ways that correspond to the human visual system's response, and also because the axes correspond to many physical characteristics of specimens. One example of this is the staining of biological tissue. To a useful approximation, hue represents the stain color, saturation represents the amount of stain, and intensity represents the specimen density; however, these spaces are awkward ones mathematically: not only does the hue value cycle through the angles from 0 to 360 degrees and then wrap around, but the conical spaces mean that increasing the intensity or luminance can alter the saturation. A geometrically simpler space that is close enough to the HSI approach for must applications and easier to deal with mathematically is the spherical L·a·b model. L, as usual, is the grey scale axis, or luminance, while **a** and **b** are two orthogonal axes that together define the color and saturation (**Figure 50**). The **a** axis runs from red (**+a**) to green (**–a**) and the **b** axis from yellow (**+b**) to blue (**–b**). Notice that the hues do not have the same angular distribution in this

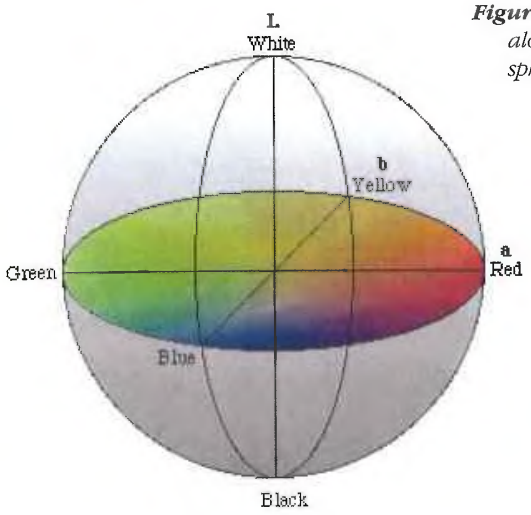


Figure 50. *The L•a•b color space. Grey values lie along a vertical north–south line through the sphere.*

space as in the usual color wheel. These axes offer a practical compromise between the simplicity of the RGB space that corresponds to hardware and the more physiologically based spaces such as CIE and HSI which are used in many color management systems, spectrophotometers and colorimeters.

Conversion between RGB space and hue-saturation-intensity coordinates can be performed in several ways, depending on the shape of the HSI space that is used. The most common choices are a sphere, cylinder, or double cone. In all cases, the intensity axis is aligned along the body diagonal of the cube, but none of the HSI space geometries exactly fits the shape of the RGB cube. This means in order to represent colors in both space the saturation values must be distorted somewhat in the conversion process. This can be seen in the example of **Figure 70** (below) in which a high-resolution panchromatic satellite image is used to replace the intensity channel in an image constructed by merging three lower-resolution R, G, B images.

A spherical color model similar to L·a·b can used as an intermediate step in converting from RGB to HSI. This is simpler than a true HSI space because it uses orthogonal axes instead of representing color as an angle. In this spherical space, the luminance (intensity) value is simply the average of R, G, and B, and the a and b values are coordinates along two orthogonal directions. A typical conversion is

$$\begin{bmatrix} L \\ a \\ b \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ -\sqrt{2}/6 & -\sqrt{2}/6 & 2\sqrt{2}/6 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \end{bmatrix} \bullet \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

From these values, the hue and saturation can be calculated as

$$H = \tan^{-1}\left(\frac{b}{a}\right)$$
$$S = \sqrt{a^2 + b^2} \tag{2}$$

Alternatively, the conversion can be performed for a conical space:

$$
\begin{aligned}
I &= \frac{R+G+B}{3} \\
S &= 1 - \frac{3 \cdot \min(R,G,B)}{R+G+B} \\
H &= \left\{ \begin{array}{ll} \cos^{-1}(z) & if(G \geq R) \\ 2\pi - \cos^{-1}(a) & if(G \leq R) \end{array} \right\} \\
z &= \frac{(2B-G-R)/2}{\sqrt{(B-G)^2 + (B-R)/(G-R)}}
\end{aligned}
\tag{3}
$$

Ledley et al. (1990) showed that the difference between these two RGB-HSI conversions lies principally in the saturation values. A fuller analysis of how this affects combinations of RGB images such as the satellite imagery in **Figure 70** can be found in Tu et al. (2001), where ways to compensate for the saturation changes are proposed.

Hardware for digitizing color images accepts either direct RGB signals from the camera, or the Y-C component signals, or a composite signal (e.g., NTSC) and uses electronic filters to separate the individual components and extract the red, green, and blue signals. As for the monochrome case discussed above, these signals are then digitized to produce values, usually 8-bit ones ranging from 0 to 255 each for R, G, and B. This takes 3 bytes of storage per pixel, so a 640 × 480 pixel image would require nearly 1 megabyte of storage space in the computer. With most cameras and electronics, the signals do not contain quite this much information, and the lowest 2 or more bits are noise. Consequently, some systems keep only 5 bits each for R, G, and B, which can be fit into two bytes. This reduction is often adequate for internet graphics and some desktop publishing applications, but when packed this way, the color information is hard to get at for any processing or analysis operations.

Color images are typically digitized as 24-bit RGB, meaning that 8 bits or 256 (linear) levels of brightness for red, green, and blue are stored. This is enough to allow display on video or computer screens, or for printing purposes, but, depending on the dynamic range of the data, may not be enough to adequately measure small variations within the image. Since photographic film can capture large dynamic ranges, some scanners intended for transparency or film scanning provide greater range — as much as 12 bits (4096 linear levels) for R, G, and B. These 36-bit images may be reduced to an "optimum" 8 bits when the image is stored. One of the problems with the dynamic range of color or grey scale storage is that the brightness values are linear, whereas the film (and human vision) are fundamentally logarithmic, so that in the dark regions of an image the smallest brightness step that can be stored is quite large and may result in visual artefacts or poor measurement precision for density values.

Further reduction of color images to 256 colors using a lookup table in which the best 256 colors are matched to the contents of the image is commonly used for computer images. This reduces the file size. The lookup table itself requires only 3 × 256 bytes to specify the R, G, B values for the 256 colors, which are written to the display hardware. Only a single byte per pixel is needed to select from this palette of colors. The most common technique for selecting the optimum palette is the Heckbert or median cut algorithm, which subdivides color space based on the actual RGB values of the pixels in the original image. For visual purposes such a reduction often provides a displayed image that is satisfactory, but for image analysis purposes this should be avoided.

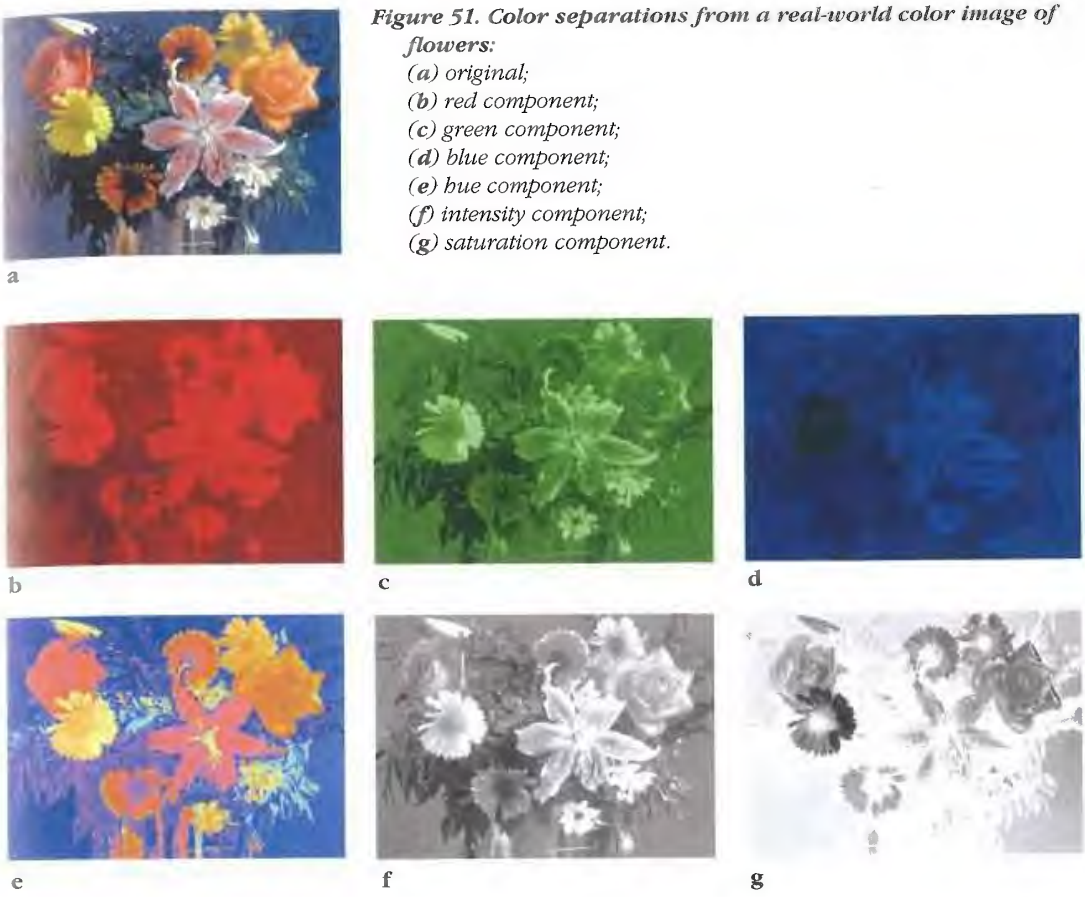


Figure 51. Color separations from a real-world color image of flowers:
*(**a**) original;*
*(**b**) red component;*
*(**c**) green component;*
*(**d**) blue component;*
*(**e**) hue component;*
*(**f**) intensity component;*
*(**g**) saturation component.*

In many situations, although color images are acquired, the "absolute" color information is not useful for image analysis. Instead, it is the relative differences in color from one region to another that can be used to distinguish the structures and features present. In such cases, it may not be necessary to store the entire color image. Various kinds of color separations are available. Calculation of the RGB components (or the complementary CMY values) is commonly used in desktop publishing, but not often useful for image analysis. Separating the image into hue, saturation, and intensity components (also called planes or channels) can also be performed. **Figures 51** and **52** show two examples, one a real-world image of flowers, and the second a microscope image of stained biological tissue. Note that some structures are much more evident in one separation than another (e.g., the pink spots on the white lily petals are not visible in the red image, but are much more evident in the green or saturation images). Also, note that for the stained tissue, the hue image shows where the stains are and the saturation image shows how much of the stain is present. This distinction is also evident in the stained tissue sample shown in **Figure 53**.

It is possible to compute the amount of any particular "color" (hue) in the image. This calculation is equivalent to the physical insertion of a transmission filter in front of a monochrome camera. The filter can be selected to absorb a complementary color (for instance, a blue filter will darken yellow regions by absorbing the yellow light) and transmit light of the same color, so that the resulting image contrast can be based on the color distribution in the image. Photographers have long used yellow filters to darken the blue sky and produce monochrome images with dramatic contrast for clouds, for example. The same color filtering can be used to convert color images to monochrome (grey scale). **Figures 54** and **55** show examples in which the computer is used to

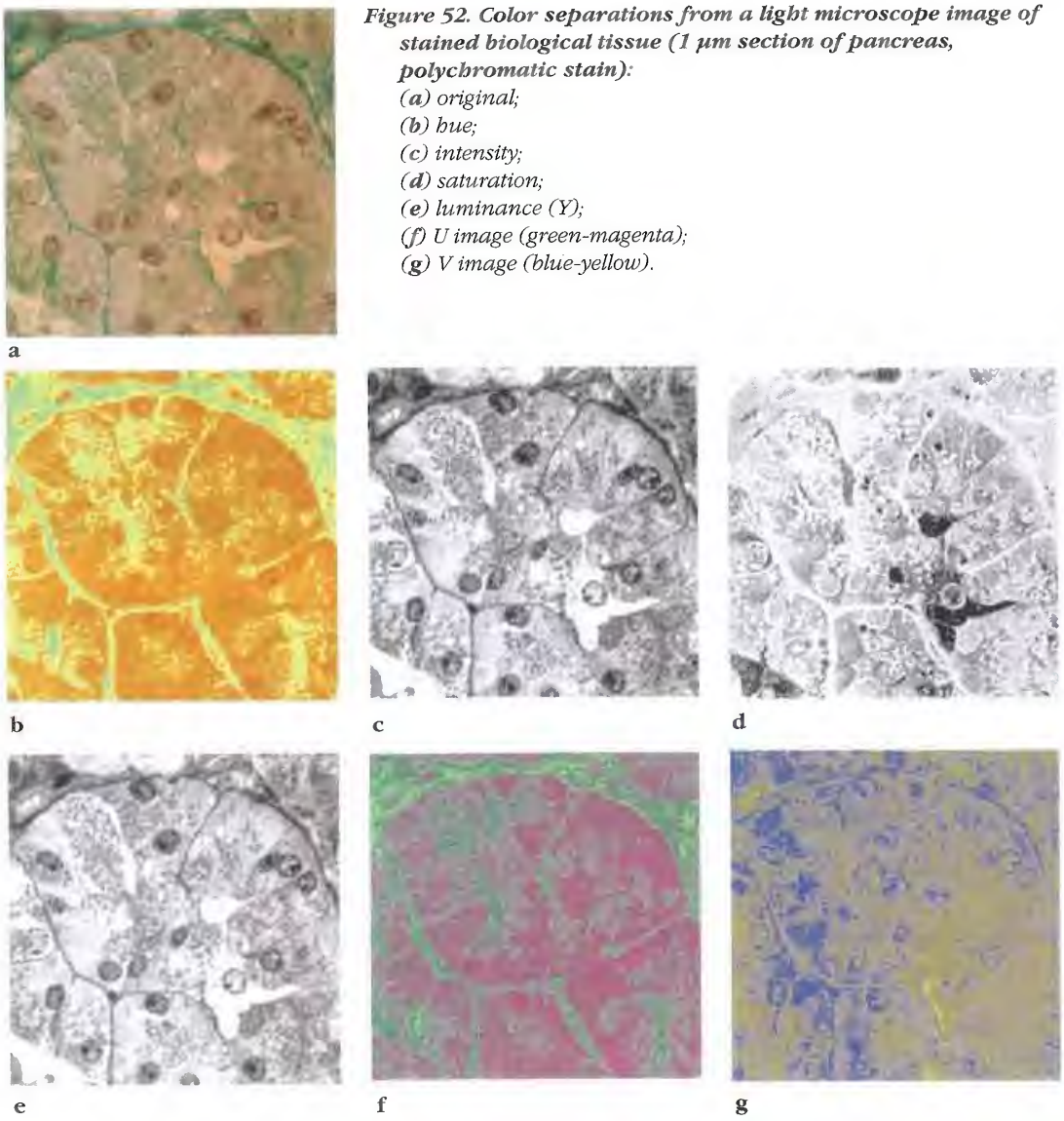


Figure 52. Color separations from a light microscope image of stained biological tissue (1 μm section of pancreas, polychromatic stain):
*(**a**) original;*
*(**b**) hue;*
*(**c**) intensity;*
*(**d**) saturation;*
*(**e**) luminance (Y);*
*(**f**) U image (green-magenta);*
*(**g**) V image (blue-yellow).*

apply the filter. This method offers greater flexibility than physical filters, because the desired wavelength can be specified, the operation can be performed later, using a stored image, and a drawer full of physical filters is not needed.

Reducing a color image to grey scale is useful in many situations and can be accomplished in many ways in addition to the obvious methods of selecting an R, G, B or H, S, I image plane, or applying a selected color filter. If obtaining the maximum grey scale contrast between structures present in the image is desired to facilitate grey scale image thresholding and measurement, then a unique function can be calculated for each image that fits a line through the points representing all of the pixels' color coordinates in color space. This least-square-fit line gives the greatest separation of the various pixel color values and the position of each pixel's coordinates as projected onto the line can be used as a grey scale value that gives the optimum contrast (Russ, 1995e). **Figure 56** shows an example.

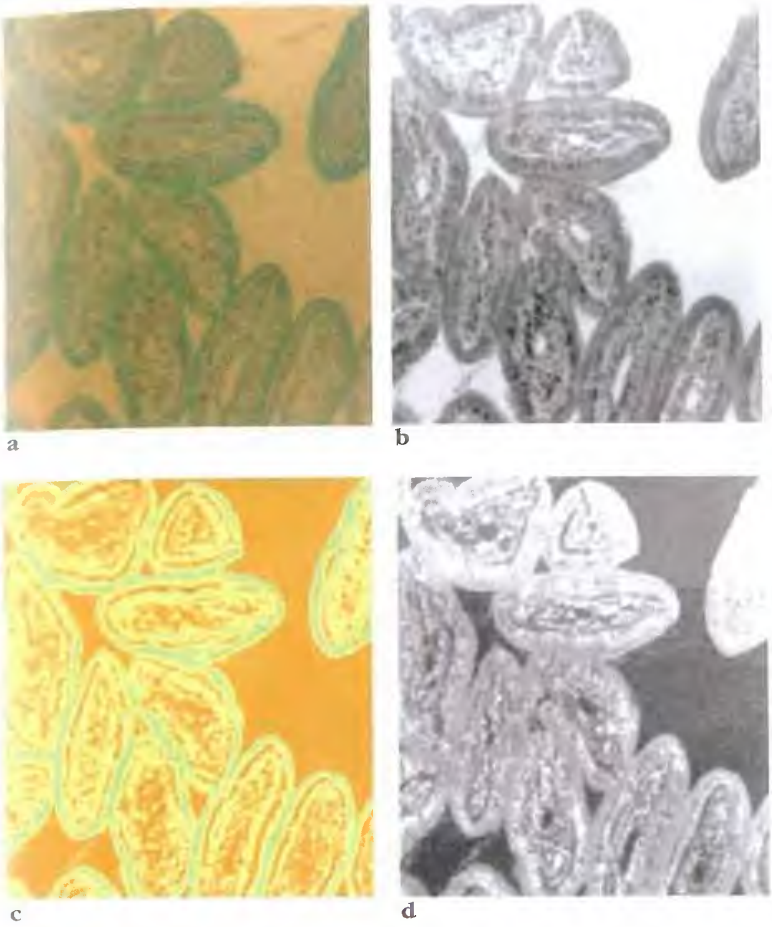


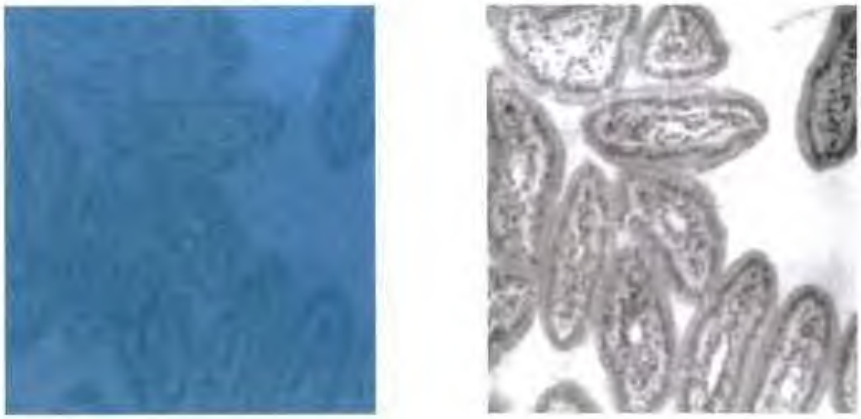
Figure 53. Microtomed tissue section of paraffin embedded mouse intestine, Feulgen stained and counter stained with fast green, Bouin's fixative (specimen courtesy of Dr. Barbara Grimes, North Carolina State University): *(**a**) original color image; (**b**) intensity component of image a; (**c**) hue component of image **a**; (**d**) saturation component of image **a**.*

Color correction

When images are acquired under different lighting conditions, the color values recorded are affected. Human vision is tolerant of considerable variation in lighting, apparently using the periphery of the viewing field to normalize the color interpretation. Some cameras, especially video cameras, have an automatic white point correction that allows recording an image from a grey card with no color, and using that to adjust subsequent colors. This same correction can be applied in software, by adjusting the relative amount of red, green and blue to set a region that is known to be without any color to pure grey. The method is easy, but does not work well in images that have some regions with highly saturated color and others with little color.

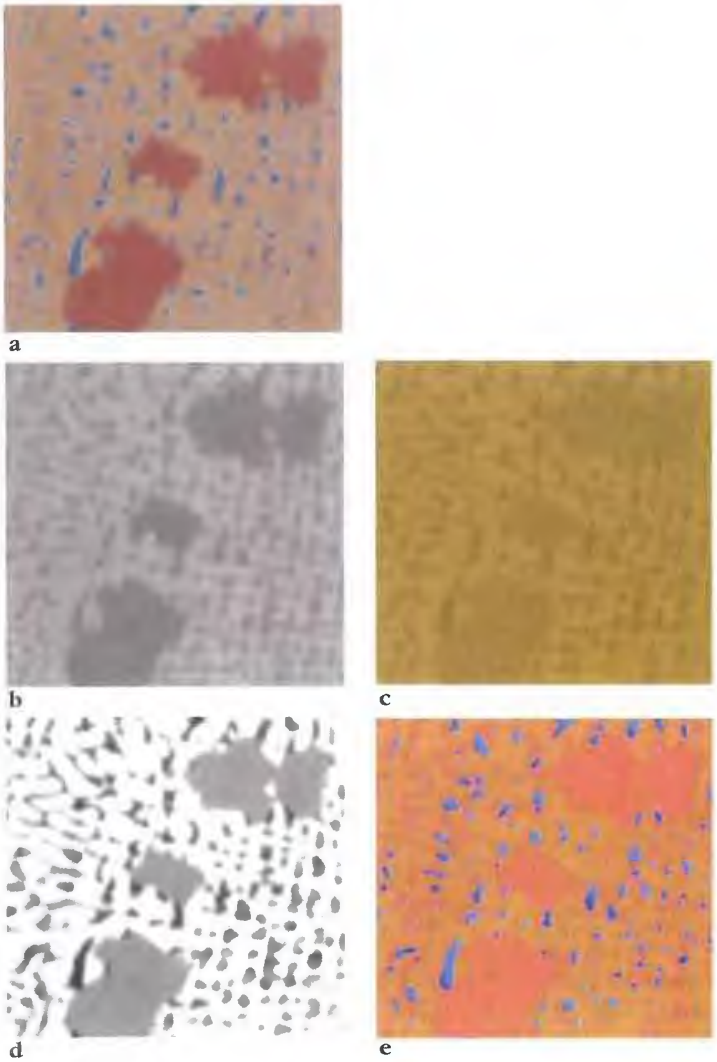
Another approach to color balance is to find a region of known color, or which is to be matched to a region in another image, and to rotate the hues of all colors in the image until they match. This often works well as an adjustment for varying illumination, but only for colors with similar hue values to the point that is matched. **Figure 57** shows an example. It is unrealistic to expect the effect of illumination to shift different colors by the same amount, and to affect only hue and not intensity or saturation.

The most complete approach to color adjustment is tristimulus correction. This requires measuring a test image with areas of pure red, green and blue color, such as the calibration card shown in **Figure 58**. In practical situations, such a card can be included in the scene being digitized. The camera



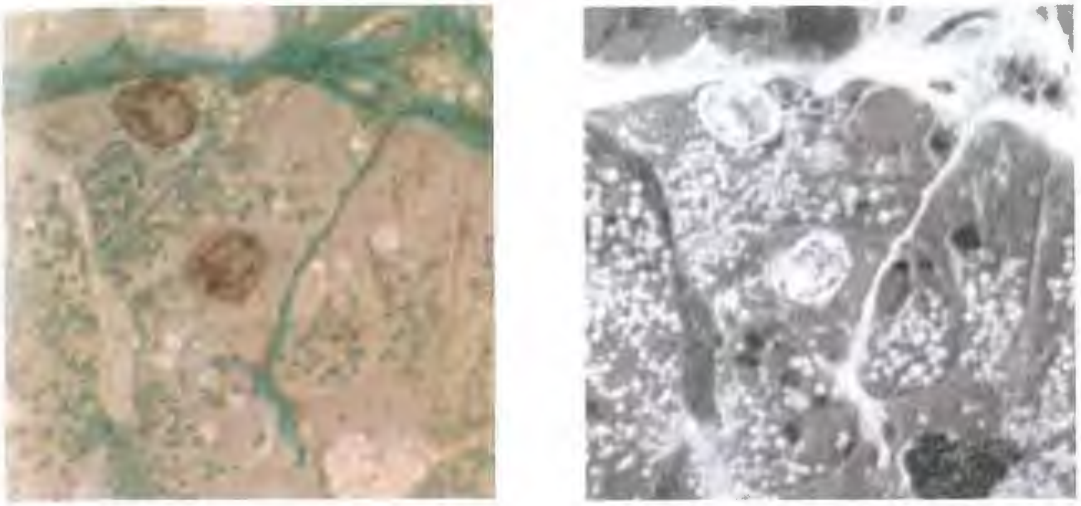
a b

Figure 54. Filtering of the image in Figure 53: *(**a**) application of a 480 nm filter to the original color image; (**b**) monochrome intensity from image c.*



a b c d e

Figure 55. Cast aluminum alloy containing 7.4% Ca, 0.8%Si, 0.2%Ti, showing two intermetallic phases (Al4Ca in blue, CaSi2 in reddish violet) *(original image from H-E. Bühler, H. P. Hougardy (1980)* Atlas of Interference Layer Metallography, *Deutsche Gesellschaft für Metallkunde, Oberursel):*
*(**a**) original color image;*
*(**b**) the panchromatic intensity component of image **a** showing inability to distinguish two of the phases;*
*(**c**) application of a 590 nm filter;*
*(**d**) monochrome intensity from image **c**;*
*(**e**) the hue component of image **a**.*



a b

*Figure 56. Color image of stained biological tissue (**a**) and the optimal grey scale image produced by fitting a line through the pixel color coordinates as described in the text (**b**).*

*Figure 57. **Color matching by shifting hue values:***
*(**a**) image of a flower in shade, with an area marked for hue measurement;*
*(**b**) another similar flower imaged in direct sunlight, with an area marked for matching;*
*(**c**) image a with the hue rotated by 12° to match the hue in the two test areas.*



a



b



c

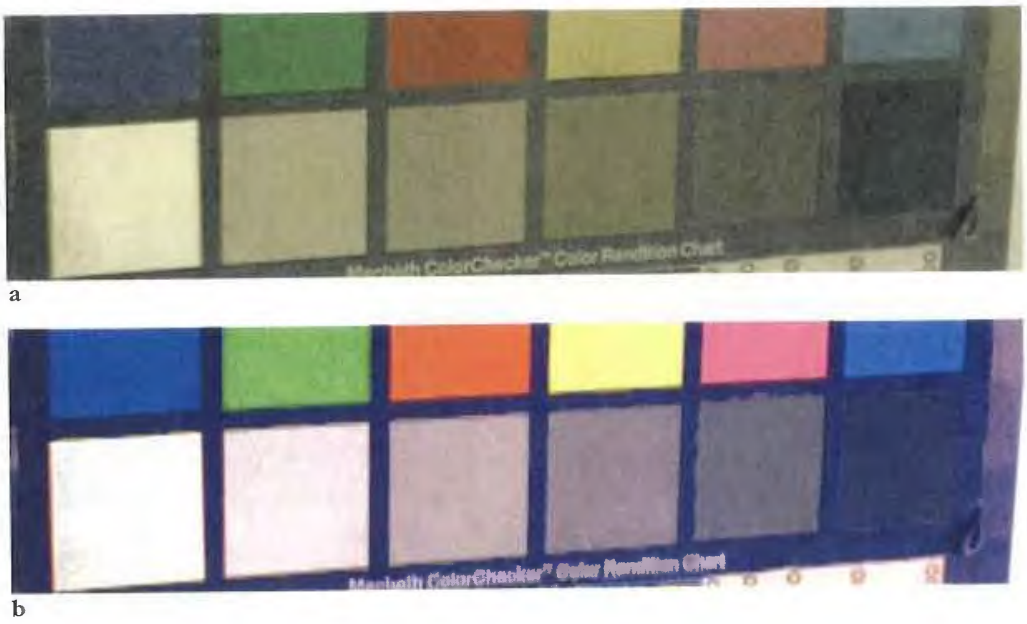


Figure 58. A portion of a Macbeth color checker chart (GretagMacbeth Corp., New Windsor, NY): (a) original image; (b) after tristimulus correction using the values in Table 2c.

(or scanner) will record some intensity for red, green, and blue in each of the areas. This produces a matrix of values as shown in **Table 2**, which can be inverted to generate the tristimulus correction matrix. Calculating new red, green, and blue values by multiplying of this matrix times the measured R, G, B values produces an image in which the calibration areas are adjusted to pure red, green, and blue, and other colors in the image are also corrected, as shown.

Color displays

Of course, color is also important as a vehicle for communicating to the computer user. Most computers use color monitors that have much higher resolution than a television set but operate on essentially the same principle. Smaller phosphor dots, a higher-frequency scan and higher-bandwidth amplifiers, and a single progressive scan (rather than interlace) produce much greater sharpness and color purity.

Besides color video monitors, other kinds of displays may be used with desktop computers. For example, many notebook computers use a liquid crystal display (LCD). The passive type of LCD display has much poorer saturation and is also slower than the somewhat more costly active-matrix type that uses a separate transistor to control each pixel (or actually, each of the RGB cells that together make up a pixel in a color display). Even the best active-matrix color LCDs, however, are inferior to most CRTs because of their lower brightness and narrower viewing angle. They also have an inferior brightness and contrast range for each color, which reduces the number of distinct colors that can be displayed.

These same display devices may be connected to a computer and placed on an overhead projector to show images on a screen. Although very convenient, and acceptable for presentations in a small room, they are inferior to dedicated projection devices. These may either use a very bright (and color corrected) light source with small LCD panels and appropriate optics, or three separate

Table 2. Calculation of tristimulus color correction for Figure 58

Matrix a — measured average R, G, B intensities in red, green, and blue target areas of original image

	Measured intensity:		
Area	**Red**	**Green**	**Blue**
Red	90.223	45.865	35.556
Green	59.933	86.941	40.296
Blue	38.088	36.635	43.028

Matrix b — normalized intensity matrix (values from **a** divided by 255)

0.353826	0.179863	0.139435
0.235031	0.340945	0.158024
0.149365	0.143667	0.168737

Matrix c — inverse of matrix **b**

4.94557	-1.46508	-2.71469
-2.27990	5.52036	-3.28586
-2.43661	-3.40327	11.12704

CRTs with red, green, and blue filters. In the latter case, the resolution is potentially higher because the individual CRTs have continuous phosphor coatings; although, careful alignment of the optics is needed to keep the three images in registration; readjustment may be needed every time the equipment is moved or even as it heats up. Getting enough brightness for viewing large images in rooms with imperfect light control is also a challenge.

A new class of displays uses the digital light modulation principle developed by Texas Instruments. An array of tiny mirrors produced by photolithography on silicon wafers (the same technology used to create electronic devices) is used to reflect light from the illumination source through appropriate optics to a viewing screen. The mirrors can be flipped from the "on" to the "off" position in nanoseconds. Moving each mirror back and forth rapidly to control the fraction of the time that it is in the "on" position controls the brightness of each pixel. A rotating filter wheel allows the array to sequentially project the red, green, and blue channels, which the eye perceives as a color display.

Other kinds of flat-panel computer displays, including electroluminescence and plasma (gas discharge), are fundamentally monochrome. Arrays of red, green, and blue LEDs can in principle be arranged to make a flat-panel display, but the difficulty of generating blue light with these devices and the prohibitive cost of such devices has so far prevented their common use. Arrays of colored light bulbs are used to show images in some sports stadia.

It takes combinations of three color phosphors (RGB) to produce the range of colors displayed on the CRT. The brightness of each phosphor is controlled by modulating the intensity of the electron beam in the CRT that strikes each phosphor. Using a separate electron gun for each

color and arranging the colored dots as triads is the most common method for achieving this control. To prevent stray electrons from striking the adjacent phosphor dot, a shadow mask of metal with holes for each triad of dots can be used. Each of the three electron beams passes through the same hole in the shadow mask and strikes the corresponding dot. The shadow mask increases the sharpness and contrast of the image, but reduces the total intensity of light that can be generated by the CRT.

A simpler design that increases the brightness applies the phosphor colors to the CRT in vertical stripes. It uses either a slotted pattern in the shadow mask or no mask at all. The simplicity of the Sony Trinitron design makes a tube with lower cost, no curvature of the glass in the vertical direction, high display brightness, and fewer alignment problems; however, the vertical extent of the phosphor and of the electron beam tends to blur edges in the vertical direction on the screen. Although this design has become fairly common for home television, most high-performance computer CRTs use triads of phosphor dots because of the greater sharpness it affords the image, particularly for lines and edges.

Image types

In traditional images with which we are visually experienced, the brightness of a point is a function of the brightness and location of the light source combined with the orientation and nature of the surface being viewed. These "surface" or "real-world" images are actually rather difficult to interpret using computer algorithms, because of their three-dimensional nature, and the fact that some surfaces may obscure others. Even for relatively flat scenes in which precedence is not a problem and the light source is well controlled, the combination of effects of surface orientation and the color, texture, and other variables make it difficult to quantitatively interpret these parameters independently. Only in the case of a carefully prepared flat and polished surface (as in the typical metallographic microscope) can interpretation of contrast as delineating phases, inclusions, grains, or other structures be successful. For additional background on image formation, and the role of lighting and optics, see (Jahne, 1997).

A second class of images that commonly arises in microscopy shows the intensity of light (or other radiation) that has come through the sample (for additional background on light microscopes and their use for imaging, see texts such as Smith, 1990; Bracegirdle and Bradbury, 1995; Bradbury and Bracegirdle, 1998). These transmission images start with a uniform light source, usually of known intensity and color. The absorption of the light at each point is a measure of the density of the specimen along that path. For some kinds of transmission images, such as those formed with electrons and x-rays, diffraction effects due to the coherent scattering of the radiation by atomic or molecular structures in the sample may also be present. These often complicate analysis, because diffraction is strongly dependent on the exact orientation of the crystalline lattice or other periodic microstructure.

To illustrate the complications that factors other than simple density can cause, **Figure 59** shows a transmission electron microscope (TEM) image of a thin cobalt foil. The evident structure is the magnetic domains in this ferromagnetic material. In each striped domain, the electron spins on the atoms have spontaneously aligned. There is no change in the atomic structure, sample density, or thickness, although the image certainly can fool the eye into thinking such variations may be present.

Likewise, **Figure 60** shows an image of a colloidal gold particle on an amorphous carbon film viewed in a high-resolution TEM. The so-called atomic resolution shows a pattern of dark spots on the substrate that appear more-or-less random, while within the gold particle they are regularly arranged. The arrangement is a result of the crystalline structure of the particle, and the spots are related to the atom positions. The spots are not simply the atoms, however, because the

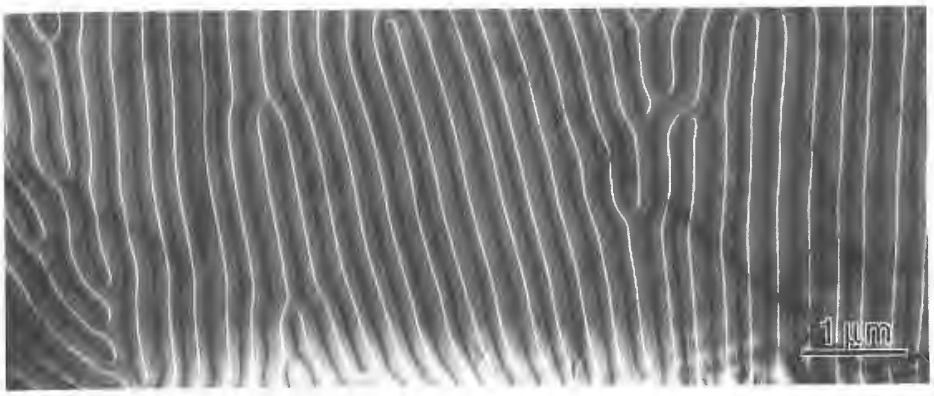
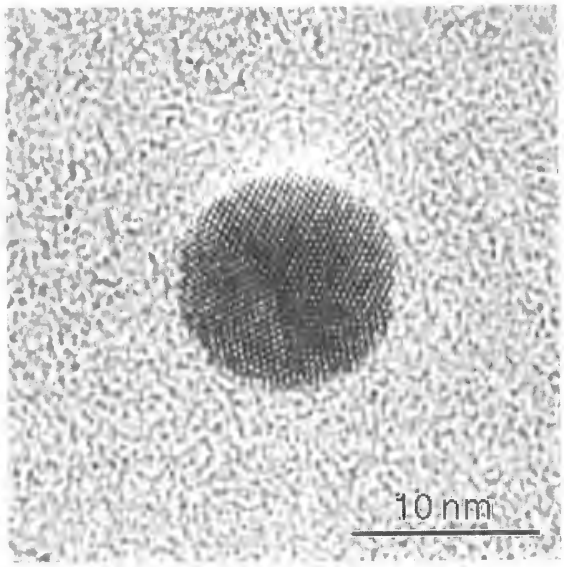


Figure 59. *TEM image of a thin metal foil of cobalt. The striped pattern reveals ferromagnetic domains, in which the electron spins of the atoms are aligned in one of two possible directions (image courtesy Hitachi Scientific Instruments).*

relationship between the structure and the image is very complex, and depends strongly on the microscope parameters and on the amount of defocus in the lens. Calculating the expected image contrast from a predicted structure is possible, and can be done routinely for simple structures. The inverse calculation (structure from image) is more interesting, but can only be accomplished by iteration.

In another subclass of transmission images, some colors (or energies) of radiation may be selectively absorbed by the sample according to its chemical composition or the presence of selective stains and dyes. Sometimes, these dyes also emit light of a different color themselves, which can be imaged to localize particular structures. In principle, this is very similar to the so-called "X-ray maps" made with the scanning electron microscope (SEM), in which electrons excite the atoms of the sample to emit their characteristic x-rays. These are imaged using the time-base of the raster scan of the microscope to form a spatial image of the distribution of each selected element, since the wavelengths of the x-rays from each atom are unique. In many of these emission images,

Figure 60. *TEM image of colloidal gold particle on an amorphous carbon substrate, used to show extremely high microscope resolution (image courtesy Hitachi Scientific Instruments).*



density variations, changes in the thickness of the specimen, or the presence of other elements can cause at least minor difficulties in interpreting the pixel brightness in terms of concentration or amount of the selected target element or compound.

A third class of images uses the pixel brightness to record distances. For example, an atomic force microscope image of a surface shows the elevation of each point on the surface, represented as a grey-scale (or pseudo-color) value. Range images are produced by raster-scan microscopes such as the scanning tunneling microscopes (STM) and atomic force microscopes (AFM), or by physical scanning with a profilometer stylus. They are also produced by interferometric light microscopes, and at larger scales by laser ranging, synthetic aperture radar, side-scan sonar, and other techniques. Additional methods that present an image in which pixel brightness values represent range information obtained indirectly by other means include stereoscopy, shape-from-shading, and motion blur.

Range imaging

Most of the measurement tools available for very flat surfaces provide a single-valued elevation reading at each point in an x,y raster or grid. Chapter 13 discusses the processing and measurement of surface range images in detail. This set of data is blind to any undercuts that may be present. Just as radar and sonar have wavelengths in the range of centimeters to meters (and thus are useful for measurements of large objects such as geologic landforms), so a much shorter measuring scale is needed for high precision, or very flat surfaces. Attempts to use SEM, conventional light microscopy, or confocal scanning light microscopy (CSLM) either on the original surfaces or on vertical sections cut through them, have been only partially satisfactory. The lateral resolution of the SEM is very good, but its depth resolution is not. Stereo pair measurements are both difficult to perform and time consuming to convert to an elevation map or range image of the surface, and the resulting depth resolution is still much poorer than the lateral resolution.

Conventional light microscopy has a lateral resolution of better than one micrometer, but the depth of field is neither great enough to view an entire rough surface nor shallow enough to isolate points along one iso-elevation contour line. The CSLM improves the lateral resolution slightly and greatly reduces the depth of field while at the same time rejecting scattered light from out-of-focus locations. The result is an instrument that can image an entire rough surface by moving the sample vertically and keeping only the brightest light value at each location, or can produce a range image by keeping track of the sample's vertical motion when the brightest reflected light value is obtained for each point in the image. It is the latter mode that is most interesting for surface measurement purposes. The resolution is better than one micrometer in all directions.

Figure 61 shows a reconstructed view of the surface of a fracture in a brittle ceramic. It is formed from 26 planes, separated by 1 μm in the z direction, each of which records a pixel only if that location is brighter than any other plane. The perspective view can be rotated to present a very realistic image of the surface, but plotting the same information in the form of a range image (in which each pixel brightness corresponds to the plane in which the brightest reflection was recorded) is more useful for measurement. **Figure 62** shows this presentation of the same surface along with an elevation profile along an arbitrary line across the surface.

This method is interesting for many macroscopically rough samples, such as fractures and some deposited coatings, but it is not adequate for the really flat surfaces that are currently being produced in many applications. The surface irregularities on a typical polished silicon wafer or precision-machined mirror surface are typically of the order of nanometers.

Three principal methods have been applied to such surfaces. Historically, the profilometer provided a tool that would accurately measure vertical elevation with a resolution approaching a nanometer. Although it has been widely used, the profilometer has two serious disadvantages for many

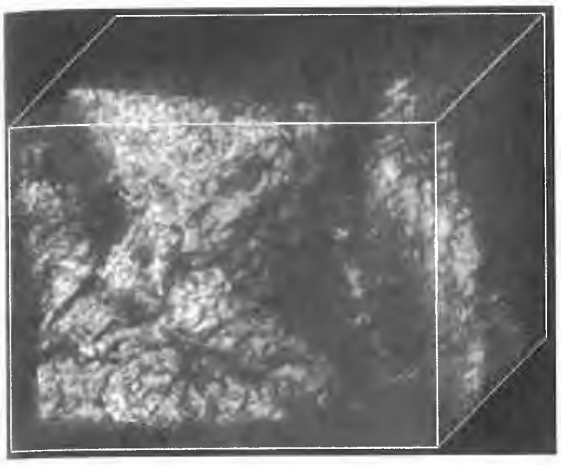


Figure 61. *Reconstructed 3D image of a brittle fracture surface in a ceramic, imaged with a confocal scanning light microscope.*

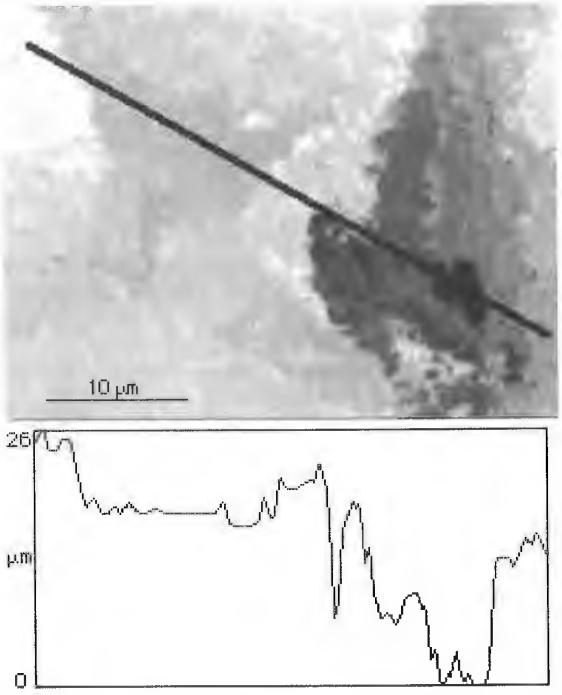


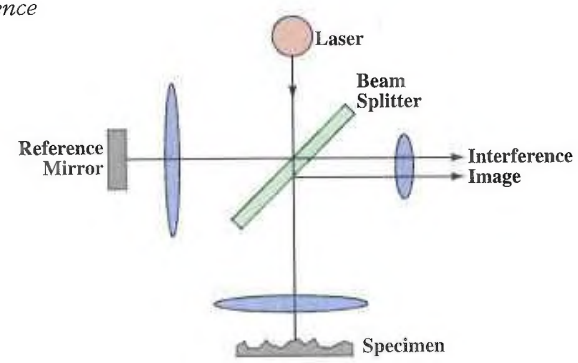
Figure 62. *Range image produced from the data in* ***Figure 61,*** *with an elevation profile along an arbitrary line.*

surface applications. The first is that it determines elevations only along a single profile. Although the analysis of such elevation profiles is straightforward, their relevance to complex surfaces that may have anisotropic properties is questionable. The second limitation is the large tip size, which makes it impossible to follow steep slopes or steps accurately.

This leaves the interferometric light microscope and the AFM as the methods of choice for studying very flat surfaces. Both are somewhat novel instruments. One is a modern implementation of the principles of light interference discovered a century ago, while the other is a technology invented and rapidly commercialized only within the past decade.

The interferometric light microscope (see the review by Robinson et al., 1991) reflects light from the sample surface as one leg in a classic interferometer, which is then combined with light from a reference leg. **Figure 63** shows a schematic diagram. The usual principles of phase-sensitive

Figure 63. Schematic diagram of an interference microscope.



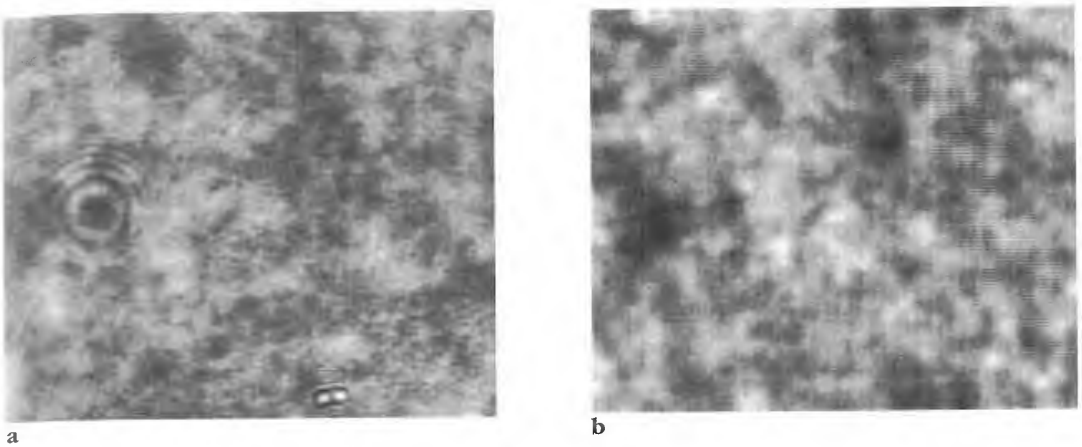
interference occur so that changes in path length (due to different elevations of points on the surface) produce changes in the intensity of the light. This image is then digitized using an appropriate CCD detector array and analog-to-digital conversion, so that a brightness value (and from it a derived elevation value) is recorded for each point on the surface. Although the wavelength of light used is typically about 630 nm, phase differences between the two legs of the interferometer of one-thousandth of the wavelength produce a change in intensity so that the vertical resolution is a few Ångstroms. The lateral resolution, however, is still of the order of one micrometer, limited by the wavelength of the light used and the design of the optics.

The interferometric light microscope suffers if the surface has high slopes or a highly specular finish, since no light will be reflected back to the detector. Such points become drop-outs in the final image. For visual purposes, it is satisfactory to fill in such missing points with a median or smoothing filter (Chapter 3), but, of course, this may bias subsequent measurements.

The vertical resolution of the interferometric microscope is very high, approaching atomic dimensions. Although the lateral resolution is much lower, it should be quite suitable for many purposes. The absolute height difference between widely separated points is not measured precisely, because the overall surface alignment and the shape or "figure" of the part is not known. It is common to deal with the problems of alignment and shape by fitting a function to the data. Determining a best-fit plane or other low-order polynomial function by least squares fitting to all of the elevation data and then subtracting it is called "detrending" the data or "form removal." It is then possible to display the magnitude of deviations of points from this surface; however, the absolute difference between points that are widely separated is affected by the detrending plane, and the ability to distinguish small differences between points that are far apart is reduced.

Figure 64 shows a very flat surface (produced by polishing a silicon wafer) imaged in a commercial interferometric light microscope (Zygo) using both Fizeau and Mirau sets of optics. The field of view is the same in both images, and the total vertical range of elevations is only about 2 nm. The two bright white spots toward the bottom of the image are probably due to dirt somewhere in the optics. These artefacts are much less pronounced with the Mirau optics. In addition, the ringing (oscillation or ripple pattern) around features that can be seen in the Fizeau image is not present with the Mirau optics. These characteristics are usually interpreted as indicating that the Mirau optics are superior for the measurement of very flat surfaces. On the other hand, the Mirau image has less lateral resolution (it appears to be "smoothed"). A pattern of horizontal lines can be discerned that may come from the misalignment of the diffuser plate with the raster scan pattern of the camera.

The AFM is in essence a profilometer that scans a complete raster over the sample surface, but with a very small tip (see the review by Wickramasinghe, 1989). The standard profilometer tip cannot



a b

Figure 64. *Comparison of interferometric range images on a flat polished silicon wafer using Fizeau (**a**) and Mirau (**b**) optics.*

follow very small or very steep-sided irregularities on the surface because of its dimensions, which are of the order of μm. The AFM tip can be much smaller and sharper, although it is still not usually fine enough to handle the abrupt steps (or even undercuts) present in microelectronic circuits and some other surfaces. The tip can be operated in either an attractive or repulsive mode of interaction between the electrons around the atom(s) in the tip and those in the surface. Usually, repulsive mode (in which the tip is pressed against the surface) does a somewhat better job of following small irregularities, but it may also cause deformation of the surface and the displacement of atoms.

Other modalities of interaction are used for these scanned-tip microscopes. The STM was the original, but it can only be used for conductive specimens and is strongly sensitive to surface electronic states, surface contamination, and oxidation. Lateral force, tapping mode, and other modes of operation developed within the last few years offer the ability to characterize many aspects of surface composition and properties, but the straightforward AFM mode is most often used to determine surface geometry, and presents quite enough complexities for understanding.

As indicated in the schematic diagram of **Figure 65**, the usual mode of operation for the AFM is to move the sample in x, y, and z. The x, y scan covers the region of interest, while the z motion brings the tip back to the same null position as judged by the reflection of a laser beam. The necessary motion is recorded in a computer to produce the resulting image. Modifications that move the tip instead of the sample, or use a linear-array light sensor to measure the deflection of the tip rather than wait for the specimen to be moved in z, do not change the basic principle. The motion is usually accomplished with piezoelectric elements whose dimensions can be sensitively adjusted by varying the applied voltage; however, there is a time lag or creep associated with these motions that appears in the modulation transfer function (MTF) of the microscope as a loss of sensitivity at the largest or smallest dimensions (lowest and highest frequencies). Locating the x, y coordinates of the tip interferometrically instead of based on the piezo driver voltages can overcome some of these problems.

Although the AFM has in principle a lateral resolution of a few Angstroms, as well as vertical sensitivity in this range, it is not always possible to realize that performance. For one thing, the adjustment of scan speeds and amplifier time constants for the visually best picture may eliminate some of the fine-scale roughness and thus bias subsequent analysis. Conversely, it may introduce additional noise from electrical or mechanical sources. Special care must also be taken to eliminate vibration.

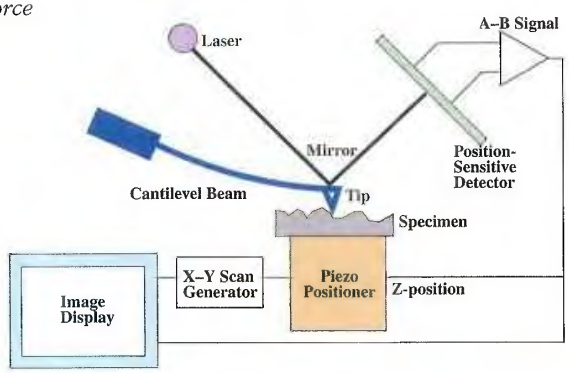


Figure 65. *Schematic diagram of an atomic force microscope (AFM).*

Most of the attention to the performance of the AFM, STM, and related instruments has been concerned with the high-resolution limit (Denley, 1990a; Denley, 1990b; Grigg et al., 1992). This is generally set by the shape of the tip, which is not easy to characterize. Many authors (Reiss et al., 1990; Pancorbo et al., 1991; Aguilar et al., 1992) have shown that it may be possible to deconvolve the tip shape and improve the image sharpness in exactly the same way that it is done for other imaging systems. This type of deconvolution is discussed in Chapters 5 and 13.

Figure 66 shows a sample of polished silicon (traditional roughness values indicate 0.2–0.3 nm magnitude, near the nominal vertical resolution limit for interferometry and AFM) with a hardness indentation. One limitation of the AFM can be seen by generating a rendered or isotropic display of the indentation, which shows that the left side of the indentation appears to be smoother than the right. This difference is an artefact of the scanning, because the tip response dynamics are different when following a surface down (where it may lag behind the actual surface and fail to record deviations) or up (where contact forces it to follow irregularities). In addition, the measured depth of the indentation is much less than the actual depth, because the tip cannot follow the deepest part of the indentation and because the calibration of the AFM in the vertical direction is less precise than in the x,y directions.

Tools for surface characterization are available with sufficient lateral and vertical resolution for application to a variety of surface range measurements. The interferometer is more convenient to use than the AFM, operates over a wide range of magnifications, and accepts large samples. It also introduces less directional anisotropy due to the instrument characteristics. The AFM, on the

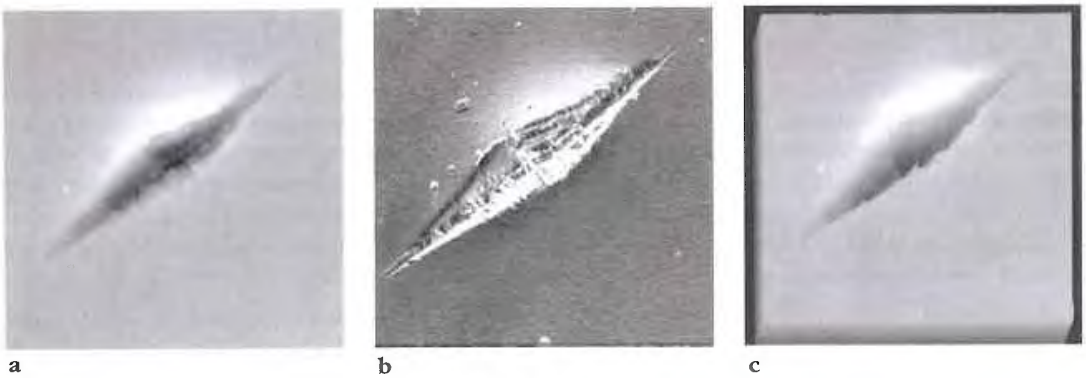


Figure 66. *AFM image of a Knoop hardness indentation in metal, shown as range (**a**), rendered (**b**) and isometric (**c**) presentations.*

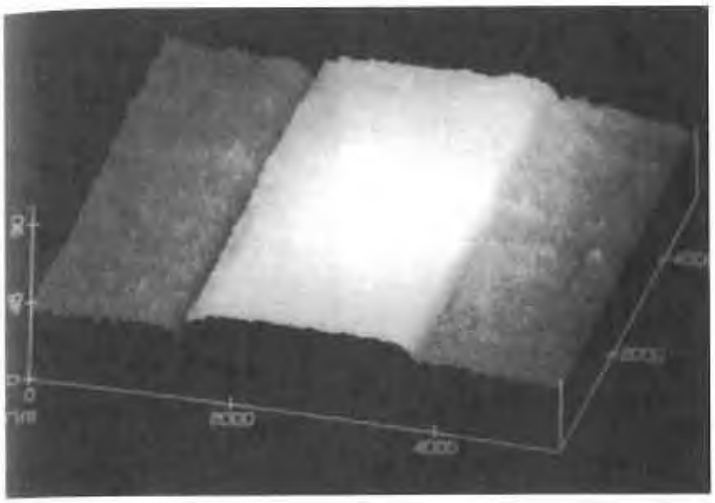


Figure 67. *Scanning tunneling microscope (STM) image. The specimen actually is flat-surfaced silicon, with apparent relief showing altered electron levels in a 2μm wide region with implanted phosphorus. (image courtesy of J. Labrasca, North Carolina State University, and R. Chapman, Microelectronics Center of North Carolina).*

other hand, has higher lateral resolution which may be required for the metrology of very fine features, now becoming commonplace in integrated circuit fabrication and nanotechnology.

It is important to keep in mind that the signal produced by many microscopes is only indirectly related to the surface elevation. In some cases, it may represent quite different characteristics of the sample such as compositional variations or electronic properties. In **Figure 67**, the apparent step in surface elevation provides an example. The surface is actually flat, but the electronic properties of the sample (a junction in a microelectronic device) produce the variation in signal.

Multiple images

For many applications, a single image is not enough. Multiple images may constitute a series of views of the same area, using different wavelengths of light or other signals. Examples include the images produced by satellites, such as the various visible and infrared wavelengths recorded by the Landsat Thematic Mapper (TM), and images from the SEM in which as many as a dozen different elements may be represented by their x-ray intensities. These images may each require processing; for example, x-ray maps are usually very noisy. Then, they are often combined either by using ratios, such as the ratio of different wavelengths used to identify crops in TM images, or Boolean logic, such as locating regions that contain both iron and sulfur in an SEM image of a mineral. **Figure 68** shows an example in which two satellite color photographs of the same region, one covering the usual visual range of wavelengths and one extending into the near infrared, are combined by constructing the ratio of infrared to green intensity as a vegetation index, insensitive to the local inclination of the surface to the sun. Combinations reduce, but only slightly, the amount of data to be stored.

Satellite images are typically acquired in several wavelengths, covering the visible and near infrared bands. Landsat images, for example, can be combined to generate a "true color" image by placing bands 1, 2, and 3 into the blue, green, and red channels of a color image, respectively. Other satellites produce images with higher resolution, although without color sensitivity. **Figure 69** shows three color channels with ground resolution of about 15 meters, and one panchromatic grey scale image with ground resolution of 5 meters. Combining the three low resolution channels to produce a color image, converting this to HSI, and then replacing the intensity channel with the higher resolution image, produces a result shown in **Figure 70** with both color information and high spatial resolution. The fact that the color data are not as sharp as the brightness data does not hinder viewing or interpretation, as discussed previously.

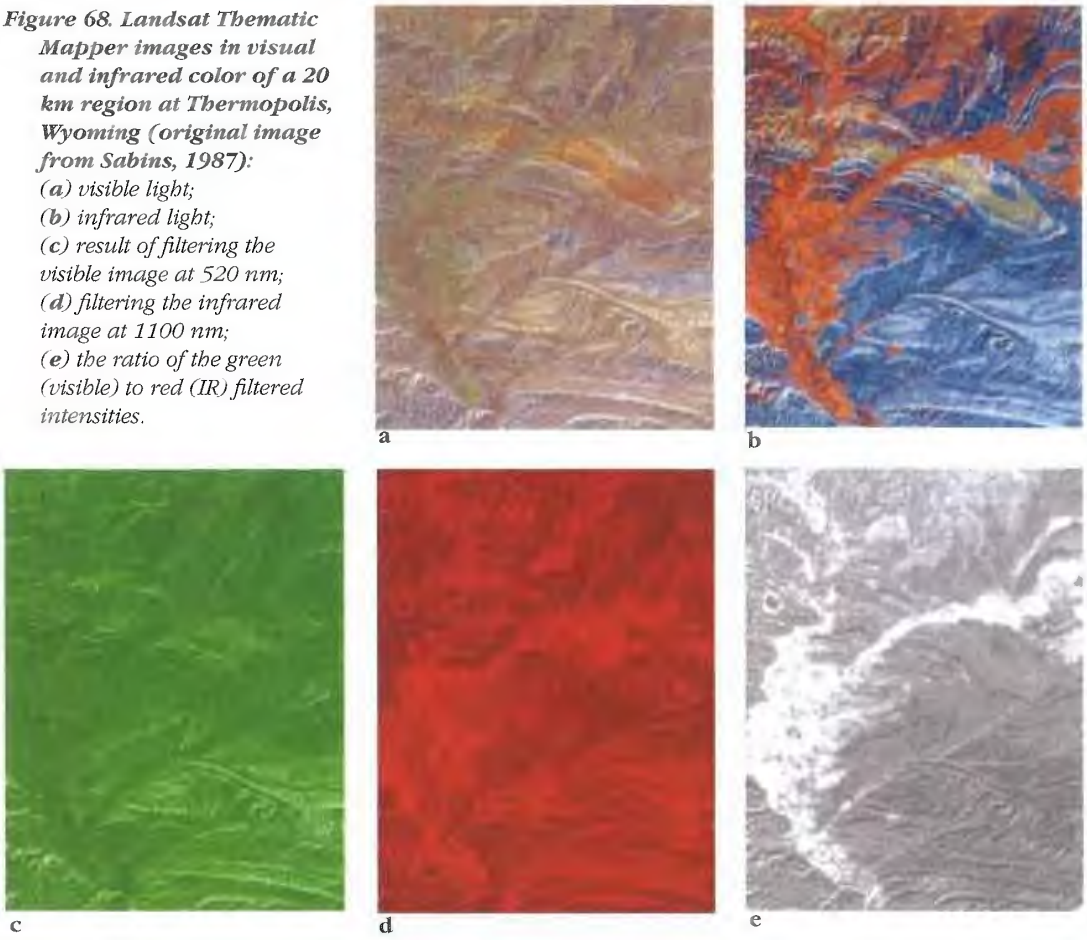


Figure 68. Landsat Thematic Mapper images in visual and infrared color of a 20 km region at Thermopolis, Wyoming (original image from Sabins, 1987):
(a) visible light;
(b) infrared light;
(c) result of filtering the visible image at 520 nm;
(d) filtering the infrared image at 1100 nm;
(e) the ratio of the green (visible) to red (IR) filtered intensities.

Another multiple image situation is a time sequence. This could be a series of satellite images used to track the motion of weather systems or a series of microscope images used to track the motion of cells or beating of cilia. In all of these cases, the need is usually to identify and locate the same features in each of the images, even though there may be some gradual changes in feature appearance from one image to the next. If the images can be reduced to data on the location of only a small number of features, then the storage requirements are greatly reduced.

A technique known as motion flow works at a lower level. With this approach, matching of pixel patterns by a correlation method is used to create a vector field showing the motion between successive images. This method is particularly used in machine vision and robotics work, in which the successive images are very closely spaced in time. Simplifying the vector field can again result in modest amounts of data, and these are usually processed in real time so that storage is not an issue. The principal requirement is that the local texture information needed for matching be present. Thus, the images must be low in random noise and each image should contain simple and distinctive surfaces with consistent illumination. The matching is performed using cross-correlation, in either the spatial or frequency domain (as discussed in Chapter 5).

A set of images can also produce three-dimensional information. They are usually a series of parallel slice images through a solid object (**Figure 71**). Medical imaging methods such as tomography and magnetic resonance images can produce this sort of data. So can some seismic imaging techniques. Even more common are various serial section methods used in microscopy. The classic method for

producing such a series of images is to microtome a series of sections from the original sample, image each separately in the light or electron microscope, and then align the images afterwards.

Optical sectioning, especially with the CSLM, which has a very shallow depth of field and can collect images from deep in partially transparent specimens, eliminates the problems of alignment. **Figure 72** shows several focal section planes from a CSLM. Some imaging methods, such as the SIMS (secondary ion mass spectrometer), produce a series of images in depth by physically eroding the

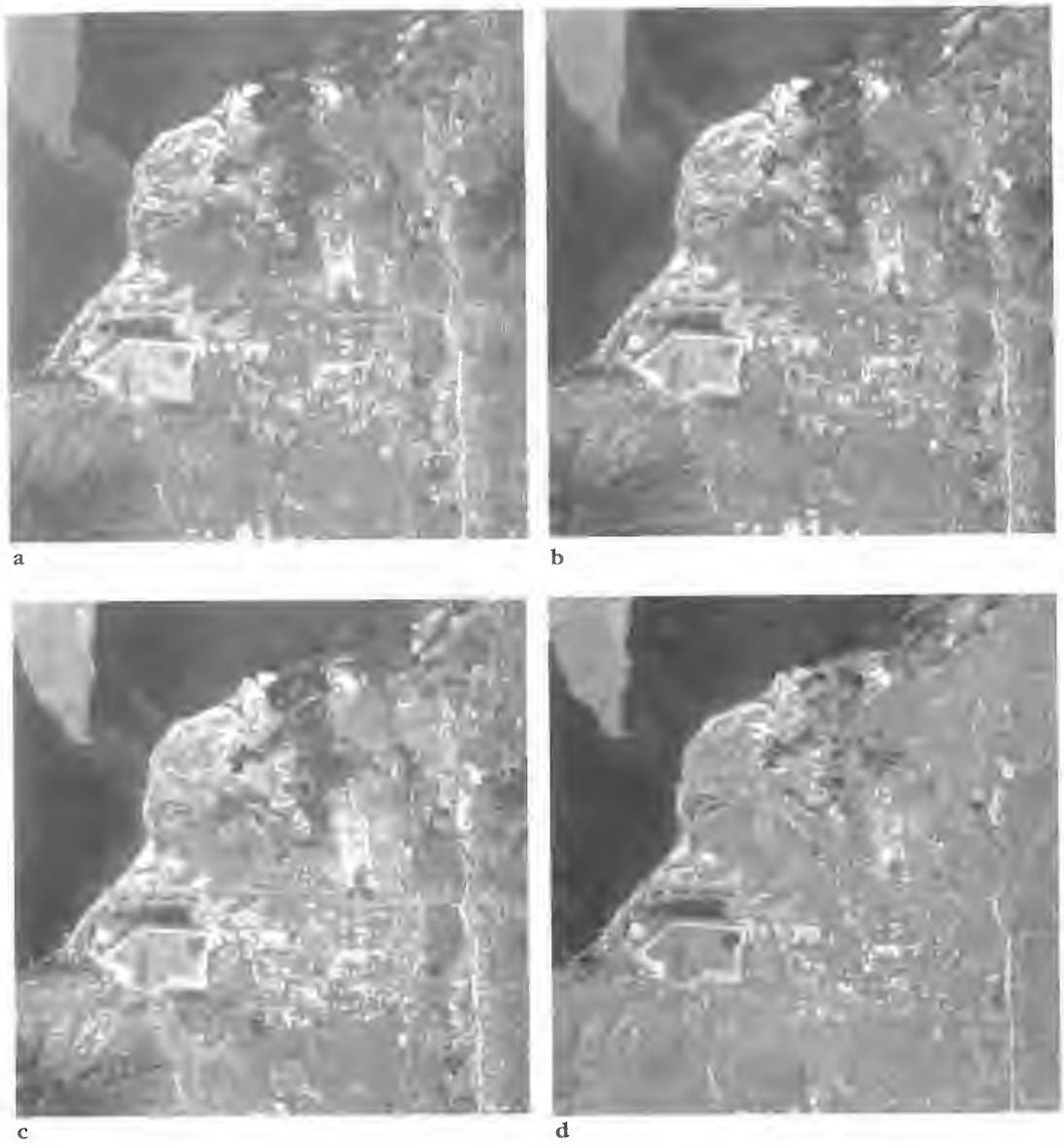


Figure 69. Satellite images of Salt Lake City *(courtesy Y. Siddiqul, I-Cubed Corp., Ft. Collins, CO):*
*(**a**) Band 1 (blue) 15-m resolution;*
*(**b**) Band 2 (green) 15-m resolution;*
*(**c**) Band 3 (red) 15-m resolution;*
*(**d**) Panchromatic, 5-m resolution.*

Figure 70. Combination of planes :
*(**a**) RGB composite using 20-m resolution images, **Figure 69 a, b,** and **c**;*
*(**b**) replacement of the intensity channel of image **a** with the higher resolution panchromatic image from **Figure 69d**.*



a b

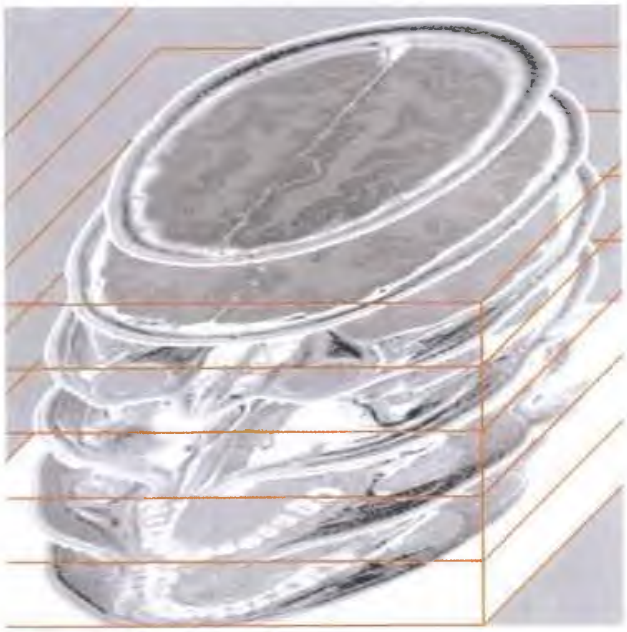


***Figure 71**. Multiple planes of pixels (sections through a human head) fill three-dimensional space. Voxels (volume elements) are ideally cubic for processing and measurement of 3D images.*

sample, which also preserves alignment. **Figure 73** shows an example of SIMS images. Sequential polishing of harder samples such as metals also produces new surfaces for imaging, but it is generally difficult to control the depth in order to space them uniformly.

The ideal situation for three-dimensional interpretation of structure calls for the lateral resolution of serial section image planes to be equal to the spacing between the planes. This produces cubic "voxels" (volume elements), which have the same advantages for processing and measurement in three dimensions that square pixels have in two; however, it is usually the case that the planes are spaced apart by much more than their lateral resolution, and special attention is given to interpolating between the planes. In the case of the SIMS, the situation is reversed and the plane spacing (as little as a few atom dimensions) is much less than the lateral resolution in each plane (typically about 1 μm).

Some techniques directly produce cubic voxel images, such as three-dimensional tomography. In this case, a series of projection images, generally using x-rays or electrons, are obtained as the sample is rotated to different orientations, and then mathematical reconstruction calculates the density of each voxel. The resulting large, three-dimensional image arrays may be stored as a series of planar slices. When a three-dimensional data set is used, a variety of processing and display modes are available. These are discussed in more detail in Chapter 12. **Figure 74** shows examples of sectioning through a series of x-ray tomographic and magnetic resonance images (MRI) in the x, y, and z planes.



Figure 72. *Serial section images formed by transmission confocal scanning laser microscopy (CSLM). These are selected views from a series of sections through the leg joint of a head louse, with optical section thickness about 0.5 μm.*

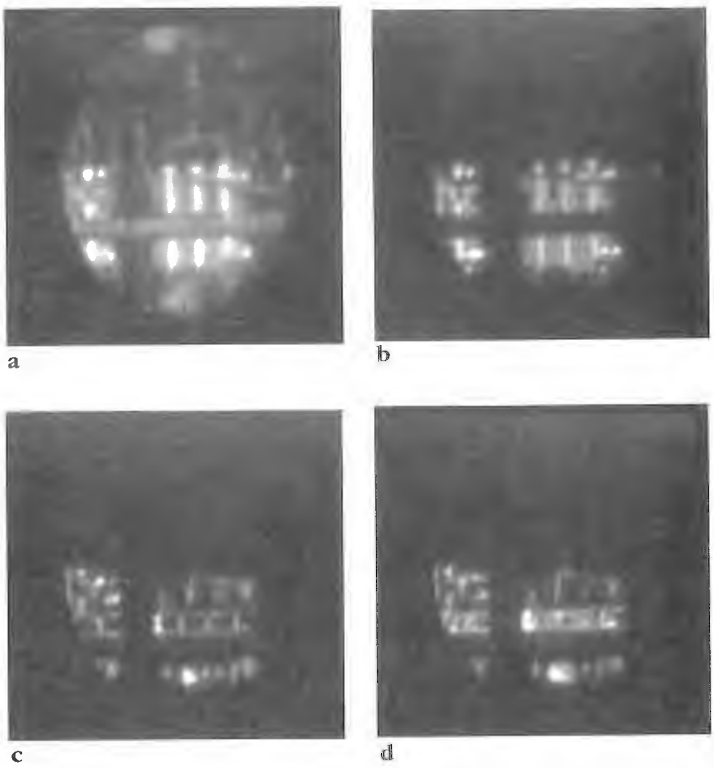
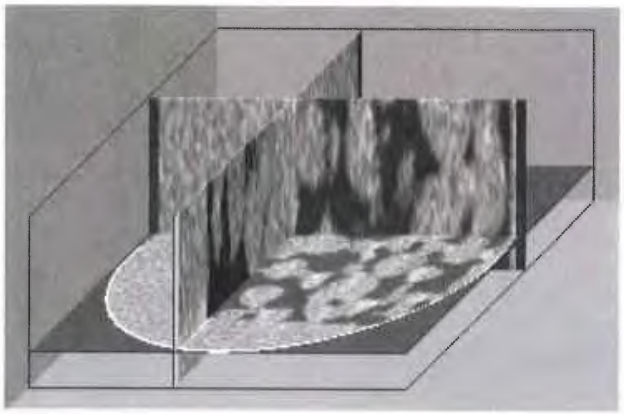
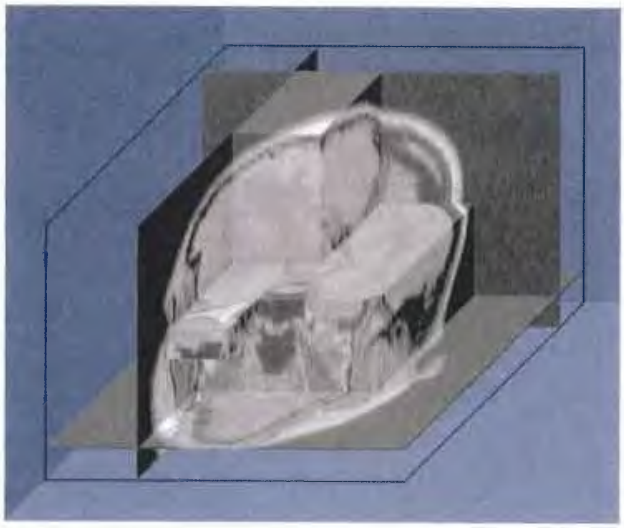


Figure 73. *SIMS (secondary ion mass spectrometer) images of Boron implanted in a microelectronic device. The images are selected from a sequence of 29 images covering a total of about 1 μm in depth. Each image is produced by physically removing layers of atoms from the surface, which erodes the sample progressively to reveal structures at greater depth.*



a



b

Figure 74. Examples of three-dimensional data sets formed by a series of planar images: *(**a**) A sintered ceramic imaged by x-ray tomography (CAT), with a characteristic dimension of a few μm; the dark regions are voids. The poorer resolution in the vertical direction is due to the spacing of the image planes, which is greater than the lateral pixel resolution within each plane. (**b**) A human head imaged by magnetic resonance imaging (MRI — the same data set as shown in **Figure 71**), with characteristic dimension of centimeters. The section planes can be positioned arbitrarily and moved to reveal the internal structure.*

Stereoscopy

Three-dimensional information can also be obtained from two images of the same scene, taken from slightly different viewpoints. Human stereoscopy gives us depth perception, although we also get important data from relative size, precedence, perspective, atmospheric haze, and other cues, which were discovered and used very effectively by artists during the Renaissance. Similar to other aspects of human vision, stereoscopy is primarily comparative, with the change of vergence angle of the eyes as we shift attention from one feature to another indicating to the brain which is closer. **Figure 75** shows schematically the principle of stereo fusion, in which the images from each eye are compared to locate the same feature in each view. The eye muscles rotate the eye to bring this feature to the fovea, and the muscles provide the vergence information to the brain. Notice that this implies that stereoscopy is only applied to one feature in the field of view at a time, and not to the entire scene. From the amount of vergence motion needed, the relative distance of the object is ascertained. Only comparative measurements are made, so only the direction or relative amount of motion required to fuse the images of each object is required.

Not all animals use this method. The owl, for instance, has eyes that are not movable in their sockets. Instead, the fovea has an elongated shape along a line that is not vertical, but angled toward

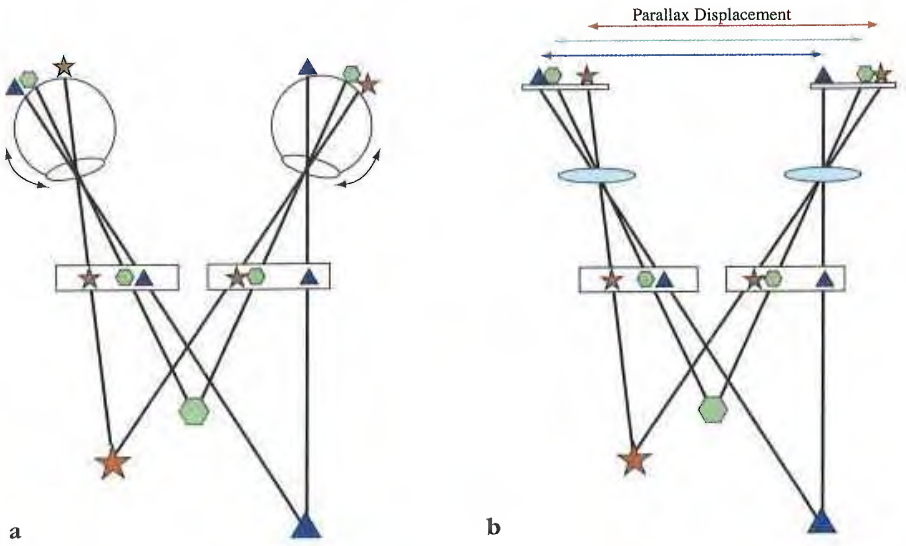


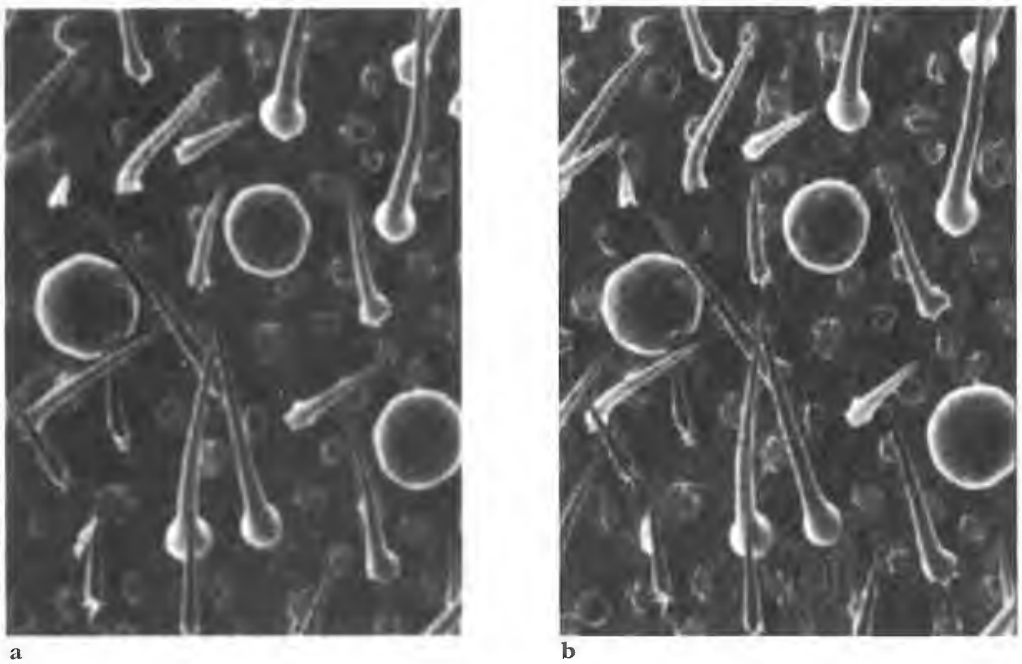
Figure 75. Stereoscopic depth perception:
(a) *The relative distance to each feature identified in both the left and right eye views is given by differences in the vergence angles by which the eyes must rotate inward to bring each feature to the central fovea in each eye. This is accomplished one feature at a time. Viewing stereo pair images provides the same visual cues to the eyes and produces the same interpretation.*
(b) *Measurement of images to obtain actual distances uses the different parallax displacements of the features in two images. The distance between the two view points must be known. Locating the same feature in both images is the hardest part of the task for automated analysis.*

the owl's feet. The owl tilts his head to accomplish fusion (bringing the feature of interest to the fovea) and judges the relative distance by the tilt required.

Although the human visual system makes only comparative use of the parallax, or vergence, of images of an object, it is straightforward to measure the relative displacement of two objects in the field of view to calculate their relative distance, or of the angle of vergence of one object to calculate its distance from the viewer. This is routinely done at scales ranging from aerial photography and map making to scanning electron microscopy.

Measurement of range information from two views is quite straightforward in principle. The lateral position of objects in these two views is different depending on their distance. From these parallax displacements, the distance can be computed by a process called stereoscopy, or sometimes photogrammetry. Computer fusion of images is a difficult task, however, that requires locating matching points in the images. Brute-force correlation methods that try to match many points based on local texture are fundamentally similar to the motion flow approach. This produces many false matches, but these are assumed to be removed by subsequent noise filtering. The alternate approach is to locate selected points in the images that are "interesting," based on their representing important boundaries or feature edges, which can then be matched more confidently. The areas between the matched points are then assumed to be simple planes.

No matter how fusion is accomplished, the displacement of the points in the two images, or parallax, gives the range. This method is used for surface elevation mapping, ranging from satellite or aerial pictures used to produce topographic maps (in which the two images are taken a short time



a b

Figure 76. *Stereo pair images from the scanning electron microscope (SEM). The specimen is the surface of a leaf; the two images were obtained by tilting the beam incident on the specimen by 8° to produce two points of view (images courtesy Japan Electron Optics Laboratory, Peabody, MA).*

apart as the satellite or airplane position changes) to scanning electron microscope metrology of semiconductor chips (in which the two images are produced by tilting the sample). **Figure 76** shows an example of a stereo pair from an SEM; **Figure 77** shows two aerial photographs and a complete topographic map drawn from them.

The utility of a stereoscopic view of the world to communicate depth information resulted in the use of stereo cameras to produce "stereopticon" slides for viewing, which was very popular more than 50 years ago. Stereo movies (requiring the viewer to wear polarized glasses) have enjoyed brief vogues from time to time. Publication of stereo-pair views to illustrate scientific papers is now relatively common. The most common formats are two side-by-side images about 7.5 cm apart (the distance between human eyes), which an experienced viewer can see without optical aids by looking straight ahead and allowing the brain to fuse the two images, and the use of different colors for each image. Overprinting the same image in red and green (or red and blue) allows the viewer with colored glasses to see the correct image in each eye, and again the brain can sort out the depth information (**Figure 78**). Some SEMs display true stereo views of surfaces using this method. Of course, this only works for grey scale images. Computer displays using polarized light and glasses are also used to display three-dimensional data, usually synthesized from calculations or simulations rather than direct imaging.

Several different measurement geometries are used (Boyde, 1973). In all cases, the same scene is viewed from two different locations, and the distance between those locations is precisely known. Sometimes this is accomplished by moving the viewpoint, for instance, the airplane carrying the camera. In aerial photography, the plane's speed and direction are known, and the time of each picture is recorded with it to give the distance traveled.

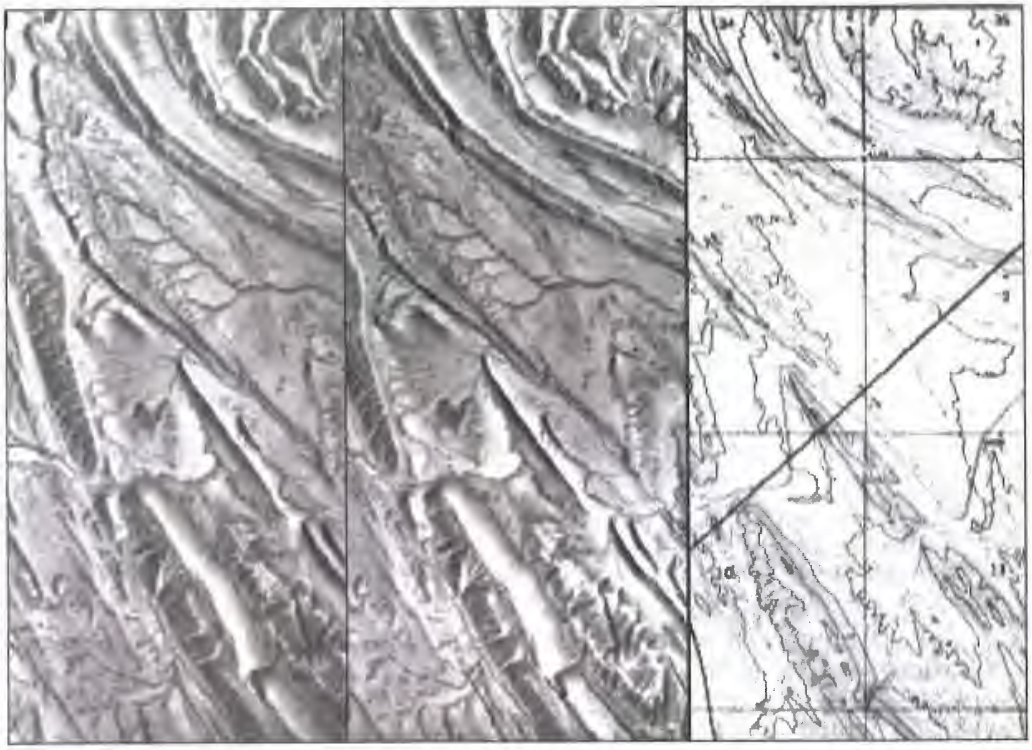


Figure 77. *Stereo pair images from aerial photography, and the topographic map showing iso-elevation contour lines derived from the parallax in the images. The scene is a portion of the Wind River in Wyoming. (From Sabins, Jr. F.F. [1987].* Remote Sensing: Principles and Interpretation, *2nd ed., W.H. Freeman, NewYork. With permission.)*



Figure 78. *Red/cyan stereo image of a fly's head. This image was captured as two separate SEM images, which were then superimposing as different color planes. To view the image, use glasses with a red filter in front of the left eye, and either a green or blue filter in front of the right eye.*

In **Figure 79**, S is the shift distance (either the distance the plane has traveled or the distance the SEM stage was translated) and WD is the working distance or altitude. The parallax ($d_1 - d_2$) from the distances between two points as they appear in the two different images (measured in a direction parallel to the shift) is proportional to the elevation difference between the two points.

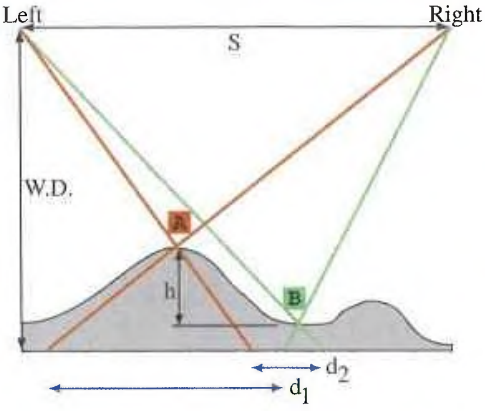


Figure 79. *Geometry used to measure the vertical height difference between objects viewed in two different images obtained by shifting the sample or viewpoint (typically used for aerial photography).*

$$h = WD \cdot \frac{(d_1 - d_2)}{S} \tag{4}$$

If the vertical relief of the surface being measured is a significant fraction of *WD*, then foreshortening of lateral distances as a function of elevation will also be present in the images. The x and y coordinates of points in the images can be corrected with the following equations. This means that rubber-sheeting to correct the foreshortening is needed to allow fitting or "tiling" together a mosaic of pictures into a seamless whole.

$$\begin{aligned} X' &= X \cdot (WD - h) / WD \\ Y' &= Y \cdot (WD - h) / WD \end{aligned} \tag{5}$$

Much greater displacement between the two eyepoints can be achieved if the two views are not in parallel directions, but are instead directed inward toward the same central point in each scene. This is rarely done in aerial photography, because it is impractical when trying to cover a large region with a mosaic of pictures and is not usually necessary to obtain sufficient parallax for measurement. When examining samples in the SEM, however, it is very easy to accomplish this by tilting the sample between two views. In **Figure 80**, the two images represent two views obtained by tilting the specimen about a vertical axis. The points A and B are separated by a horizontal distance d_1 or d_2 that is different in the two images. From this parallax value and the known tilt angle δ applied between the two images, the height difference h and the angle θ of a line joining the points (usually a surface defined by the points) can be calculated (Boyde, 1973) as:

$$\theta = \tan^{-1}\left\{\frac{\cos\delta - d_2 / d_1)}{\sin\delta}\right\} \tag{6}$$

$$h_1 = \frac{(d_1 \cdot \cos\delta - d_2)}{\sin\delta}$$

Notice that the angle θ is independent of the magnification, since the distances enter as a ratio.

When two angled views of the same region of the surface are available, the relative displacement or parallax of features can be made quite large relative to their lateral magnification. This makes it

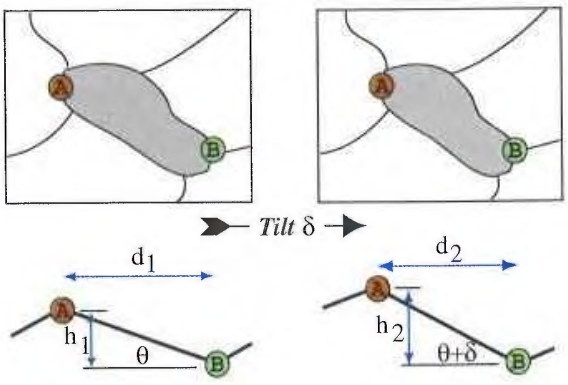


Figure 80. *Geometry used to measure the vertical height difference between points viewed in two different images obtained by tilting the sample (typically used for microscopy).*

possible to measure relatively small amounts of surface relief. Angles of 5 to 10° are commonly used; for very flat surfaces, tilt angles as great as 20° can sometimes be useful. When large angles are used with rough surfaces, the images contain shadow areas where features are not visible in both images and fusion becomes very difficult. Also, when the parallax becomes too great in a pair of images, it can be difficult for the human observer to fuse the two images visually if this step is used in the measurement operation.

Many of the measurements made with stereo-pair photographs from both SEM and aerial photography are made using extensive human interaction. The algorithms that have been developed for automatic fusion require many of the image processing operations that will be described in later chapters to make possible the identification of the same features in each image. For the moment, it is enough to understand the principle that identifying the same points in left and right images, and measuring the parallax, gives the elevation. With data for many different pairs of points, it is possible to construct a complete map of the surface. The elevation data can then be presented in a variety of formats, including a range image encoding elevation as the brightness or color of each pixel in the array. Other display modes, such as contour maps, isometric views, or shaded renderings can be generated to assist the viewer in interpreting these images, which are not common to our everyday experience. The range image data are in a suitable form for many types of image processing and for the measurement of the surface area or the volume above or below the surface.

The majority of elevation maps of the earth's surface being made today are determined by the stereoscopic measurement of images, either taken from aerial photography or satellite remote imaging. Of course, two-thirds of the earth is covered by water and cannot be measured this way. Portions of the sea bottom have been mapped stereoscopically by side-scanned sonar. The technology is very similar to that used for the radar mapping of Venus, except that sonar uses sound waves (which can propagate through water) and radar uses high-frequency (millimeter length) electromagnetic radiation that can penetrate the opaque clouds covering Venus.

The synthetic aperture radar (SAR) used by the Magellan probe to map Venus does not directly give elevation data to produce a range image. The principle of SAR is neither new, nor is it restricted to satellites and space probes. Aerial mapping of desert regions has been used to penetrate through the dry sand and map the underlying land to find ancient watercourses, for instance. The principle of SAR is that the moving satellite (or other platform) emits a series of short pulses directed downwards and to one side of the track along which it is moving. Direction parallel to the track is called the azimuth and the direction perpendicular to it is called the range (not

to be confused with the range image that encodes the elevation information). The name "synthetic aperture" refers to the fact that the moving antenna effectively acts as a much larger antenna (equal in size to the distance the antenna moves during the pulse) that can more accurately resolve directions in azimuth.

The radar records the intensity of the returning pulse, the travel time, and the Doppler shift. The intensity is a measure of the surface characteristics, although the radar (or sonar) reflectivity is not always easy to interpret in terms of the surface structure and is not directly related to the albedo (or reflectivity) for visible light. The travel time for the pulse gives the range. For a perfectly flat surface, there would be an arc of locations on the surface that would have the same range from the antenna. Because of the motion of the satellite or airplane, each point along this arc produces a different Doppler shift in the signal frequency. Measuring the Doppler shift of each returned pulse provides resolution along the arc. Each point on the ground contributes to the return pulse with a unique range and Doppler shift, which allows a range map of the ground to be reconstructed.

The surface is not flat, therefore multiple combinations of location and elevation are possible, which could produce the same return signal. Combining the measurements from several overlapping sweeps (for Magellan, several sequential orbits) allows the elevation data to be refined. The database for Venus resolves points on the ground with about 120-m spacing and has elevation data with resolutions that vary from 120 to 300 m (depending on where along the track, and how far from the center of the track, they were located). For each point, the elevation and the reflectivity are stored (**Figure 81**). In many of the published renderings of these data, the elevation values are used to construct the surface shape and the reflectivity values are used to color the surface (**Figure 82**). Of course, the colors do not reflect the actual visual appearance of the surface.

Synthetic aperture ranging with either radar or sonar is not the only way these signals can be used to construct a range map. Directing a beam straight down and measuring the echo, or return time, gives the range at a single point. Many such measurements can be used to construct a map or image. The simple "fish-finder" type of sonar can be used to construct a map of a lake bottom in this way, if the boat is steered back and forth in a raster pattern to cover the whole surface. Other signals can be used, as well. The Clementine mission to the moon used a laser beam from the orbiting space craft in a similar fashion. By pulsing the laser and waiting for the echo, points were measured every few hundred meters across the entire lunar surface with a vertical resolution of about 40 m.

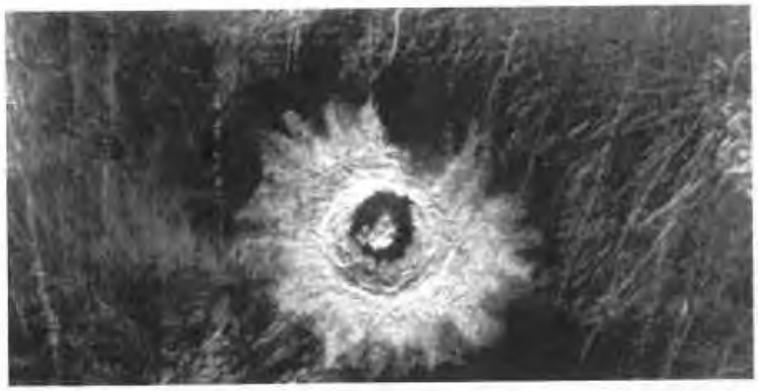


Figure 81. *Range image of Venus obtained from synthetic aperture radar (courtesy Jet Propulsion Laboratory, Pasadena, CA).*



Figure 82. *Rendered surface of Venus from SAR data (courtesy Jet Propulsion Laboratory, Pasadena, CA).*

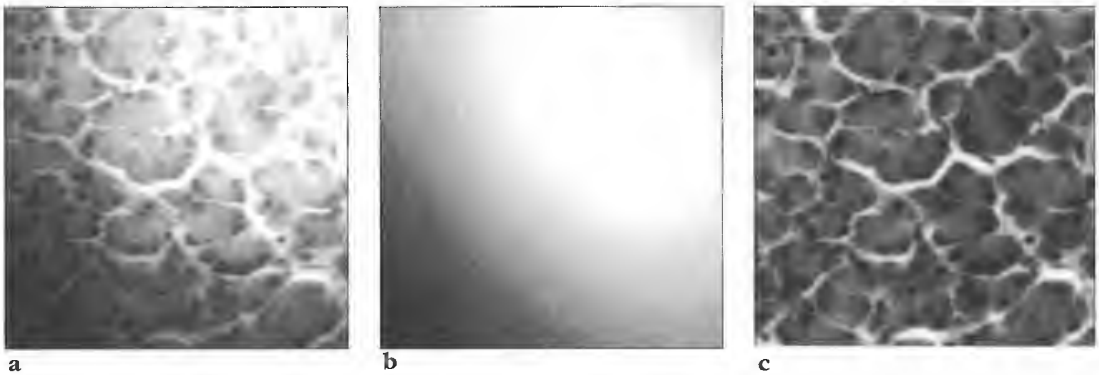
Imaging requirements

Given the diversity of image types and sources described earlier, several general criteria can be prescribed for images intended for computer processing and analysis. The first criteria is the need for global uniformity. The same type of feature should look the same wherever it appears in the image. This implies that brightness and color values should be the same and, consequently, that illumination must be uniform and stable for images acquired at different times. When surfaces are nonplanar, such as the earth as viewed from a satellite or a fracture surface in the microscope, corrections for the changing local orientation may be possible, but this usually requires extensive calculation and/or prior knowledge of the surface and source of illumination.

Figure 83 shows an example of a microscope image with nonuniform illumination. Storing a "background" image with no specimen present (by moving to a clear space on the slide) allows this nonuniformity to be leveled. The background image is either subtracted from or divided into the original (depending on whether the camera has a logarithmic or linear response). This type of leveling is discussed in Chapter 3 on correcting image defects, along with other ways to obtain the background image when it cannot be acquired directly.

The requirement for uniformity limits the kinds of surfaces that are normally imaged. Planar surfaces, or at least simple and known ones, are much easier to deal with than complex surfaces. Simply connected surfaces are much easier to interpret than ones with arches, bridges, caves and loops that hide some of the structure. Features that have precedence problems, in which some features hide entirely or in part behind others, present difficulties for interpretation or measurement. Illumination that casts strong shadows, especially to one side, is also undesirable in most cases. The exception occurs when well-spaced features cast shadows that do not interfere with each other. The shadow lengths can be used with the known lighting geometry to calculate feature heights. **Figure 84** shows an example in aerial photography. One form of sample preparation for the TEM deposits a thin film of metal or carbon from a point source, which also leaves shadow areas behind particles or other protrusions that can be used in the same way (**Figure 85**).

In addition to global uniformity, we generally want local sensitivity to variations. This means that edges and boundaries must be well delineated and accurately located. The resolution of the camera sensor was discussed earlier. Generally, anything that degrades high frequencies in the signal



a b c

Figure 83. *A microscope image obtained with nonuniform illumination (due to a misaligned condenser lens). The "background" image was collected under the same conditions, with no sample present (by moving to an adjacent region on the slide). Subtracting the background image and expanding the contrast of the difference produces a "leveled" image with uniform brightness values for similar structures.*

chain will disturb the subsequent ability to identify feature boundaries or locate edges for measurement. On the other hand, such simple problems as dust on the optics can introduce local variations that may be mistaken for image features, causing serious errors.

Measurement of dimensions requires that the geometry of the imaging system be well known. Knowing the magnification of a microscope or the altitude of a satellite is usually straightforward. Calibrating the lateral scale can be accomplished either by knowledge of the optics or by using an image of a known scale or standard. When the viewing geometry is more complicated, either because the surface is not planar or the viewing angle is not perpendicular, measurement is more difficult and requires determination of the geometry first, or the inclusion of a scale or fiducial marks on the object being viewed.



Figure 84. *Aerial photograph in which length of shadows and knowledge of the sun position permit calculation of the heights of trees and the height of the piles of logs in the lumberyard, from which the amount of wood can be estimated.*

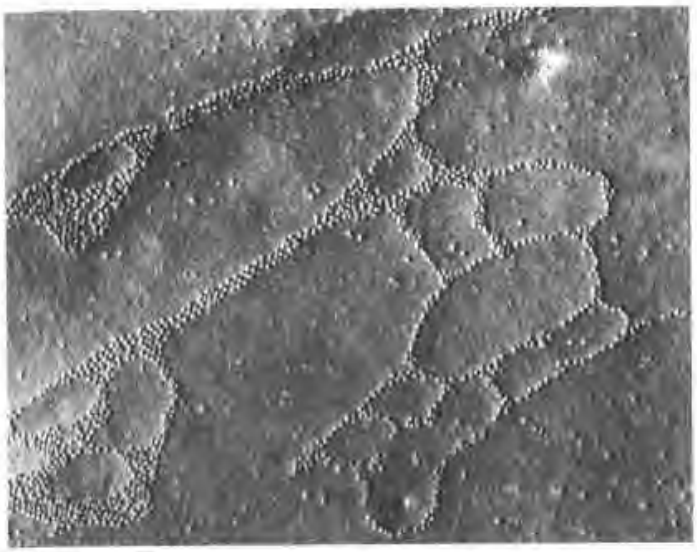


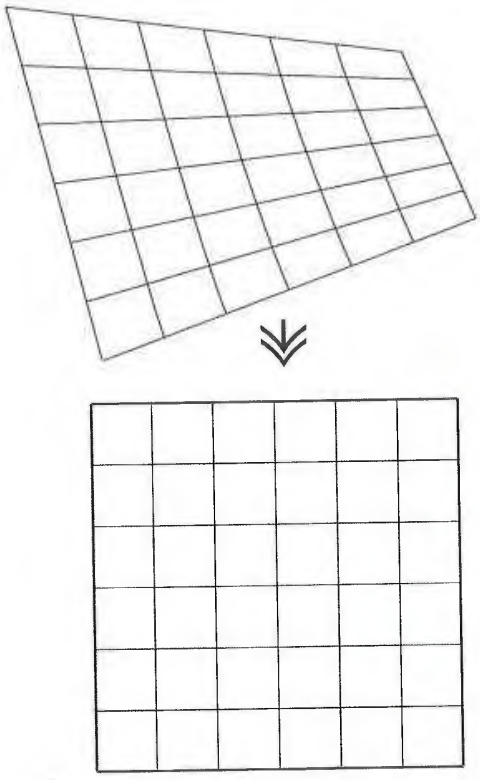
Figure 85. *Electron microscope image showing shadowed particles delineating the gap junctions between cells, revealed by freeze-fracturing the tissue.*

Figure 86 shows the simplest kind of distortion when a planar surface is viewed at an angle. Different portions of the image have different magnification scales, which makes subsequent analysis difficult. It also prevents combining multiple images of a complex surface into a mosaic. This problem is evident in two applications at very different scales. Satellite images of the surface of planets are assembled into mosaics covering large areas only with elaborate image warping to bring the edges into registration. This type of warping is discussed in Chapter 3. SEM images of rough surfaces are more difficult to assemble in this way, because the overall specimen geometry is not so well known, and the required computer processing is more difficult to justify.

Measuring brightness information, such as density or color values, requires a very stable illumination source and sensor. Color measurements are easily affected by changes in the color temperature of an incandescent bulb due to minor voltage fluctuations or as the bulb warms up or ages. Fluorescent lighting, especially when used in light boxes with x-ray films or densitometry gels, may be unstable or may introduce interference in solid-state cameras due to the high-frequency flickering of the fluorescent tube. Bright specular reflections may cause saturation, blooming, or shifts in the camera gain.

It helps to bear in mind the purpose if digitizing an image into a computer. Some of the possibilities are listed in the following paragraphs, and these place different restrictions and demands on the hardware and software used. Subsequent chapters discuss these topics in greater detail. The emphasis throughout this book is on the results produced by various processing and measurement techniques, with plain English descriptions of the methods and illustrations comparing different approaches. Numerous books provide computer codes that implement various algorithms, such as (Parker, 1997; Ritter and Wilson, 2001; Pavlidis, 1982; Myler and Weeks, 1993; Umbaugh, 1998; Seul et al., 2000; Lichtenbelt et al., 1998), with more books rolling off the presses every month.

A great many books on digital image processing are also available, some of which delve deeply into the mathematical underpinnings of the science, and others which concentrate on applications in a particular field. A representative selection includes (Gonzalez and Woods, 1993; Pratt, 1991; Rosenfeld and Kak, 1982; Weeks, 1996; Pitas, 2000; Nikolaidis and Pitas, 2001; Costa & Cesar, 2001; Sonka et al., 1999; Kriete, 1992; Hader, 1992; Sanchez and Canton, 1999; Russ, 2001). The emphasis in this text is on the generality of the methods, showing that the same techniques apply to a broad range of types of images, and trying to educate the user in the performance and results instead of the theory.



a



b

Figure 86. *Geometric distortion occurs when a surface is viewed from a position away from the surface normal. Correcting this distortion to obtain a rectilinear image that can be properly processed and measured, or fitted together with adjoining images, requires knowing the viewing geometry and/or including some known fiducial marks in the scene which can be used to determine it.*

Storing and filing of images becomes more attractive as massive storage devices (such as writable DVD disks) drop in price or where multiple master copies of images may be needed in more than one location. In many cases, this application also involves hardcopy printing of the stored images and transmission of images to other locations. If further processing or measurement is not required, then compression of the images is worthwhile. The advantage of electronic storage is that the images do not degrade with time and can be accessed by appropriate filing and cross-indexing routines. On the other hand, film storage is far cheaper and offers much higher storage density as well as higher image resolution, and it is likely that devices for examining film will still be available in 100 years, which may not be the case for DVD disks.

Enhancement of images for visual examination requires a large number of pixels and adequate pixel depth so that the image can be acquired with enough information to perform the filtering or other operations with fidelity and then display the result with enough detail for the viewer. Uniformity of illumination and control of geometry are not of great importance. When large images are used, and especially for some of the more time-consuming processing operations, or when interactive experimentation with many different operations is intended, this application may benefit from very fast computers or specialized hardware.

Measurement of dimensions and density values can often be performed with modest image resolution if the magnification or illumination can be adjusted beforehand to make the best use of the image sensor. Processing may be required before measurement (for instance, derivatives are often used to delineate edges for measurement), but this can usually be handled completely in software. The most important constraints are tight control over the imaging geometry, and the uniformity and constancy of illumination.

Quality control applications usually do not involve absolute measurements so much as watching for variations. In many cases, this is handled simply by subtracting a reference image from each acquired image, point by point, to detect gross changes. This can be done with analog electronics at real-time speeds. Preventing variation due to accidental changes in the position of camera or targets, or in illumination, is a central concern.

Microstructural research in either two or three dimensions usually starts with image measurement and has the same requirements as noted previously, plus the ability to subject the measurement values to appropriate stereological and statistical analysis. Interpretation of images in terms of structure is different for images of planar cross sections or projections (**Figure 87**). The latter are familiar to human vision, while the former are not; however, section images, such as the one in **Figure 88**, contain rich information for measurement in three dimensions that can be revealed by statistical analysis. Projection images, such as the one in **Figure 89**, may present greater difficulties for interpretation.

Three-dimensional imaging utilizes large data sets, and in most cases the alignment of two-dimensional images is of critical importance. Some three-dimensional structural parameters can be inferred from two-dimensional images. Others, principally topological information, can only be determined from the three-dimensional data set. Processing and measurement operations in three dimensions place extreme demands on computer storage and speed. Displays of three-dimensional information in ways interpretable by, if not familiar to, human users are improving, but need further development of algorithms. They also place considerable demands on processor and display speed.

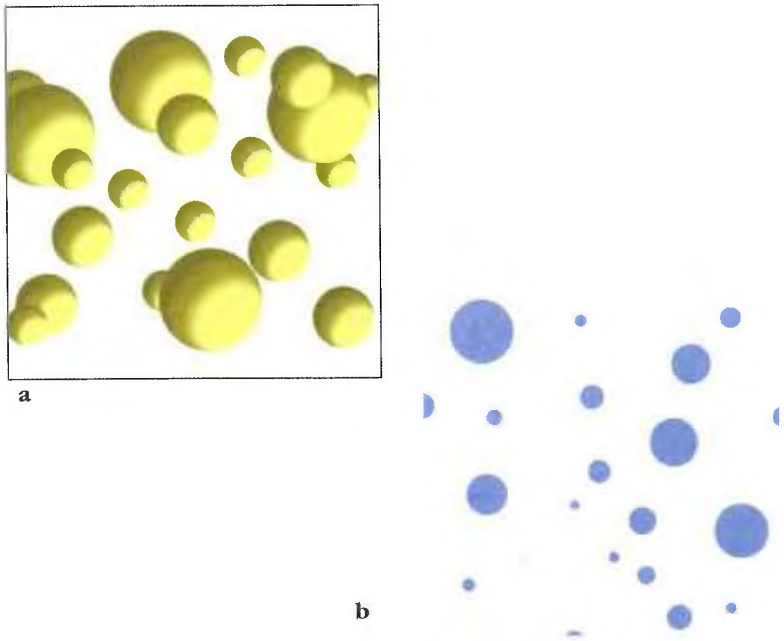


Figure 87. *Projection images, such as the spheres shown in* ***(a)****, are familiar, showing the external surfaces of features; however, some features are partially or entirely obscured and it is not easy to determine the number or size distribution. Cross-section images, as shown in* ***(b)****, are unfamiliar and do not show the maximum extent of features, but statistically it is possible to predict the size distribution and number of the spheres.*

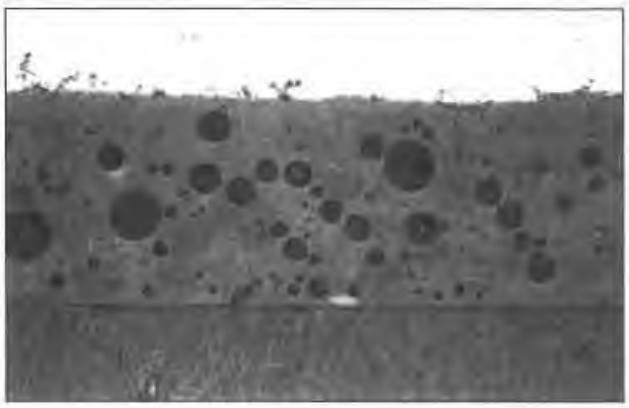


Figure 88. *Light microscope image of section through a colored enamel coating applied to steel (courtesy V. Benes, Research Institite for Metals, Panenské Brezany, Czechoslovakia). The spherical bubbles arise during the firing of the enamel. They are sectioned to show circles whose diameters are smaller than the maximum diameter of the spheres, but since the shape of the bubbles is known it is possible to infer the number and size distribution of the spheres from the data measured on the circles.*

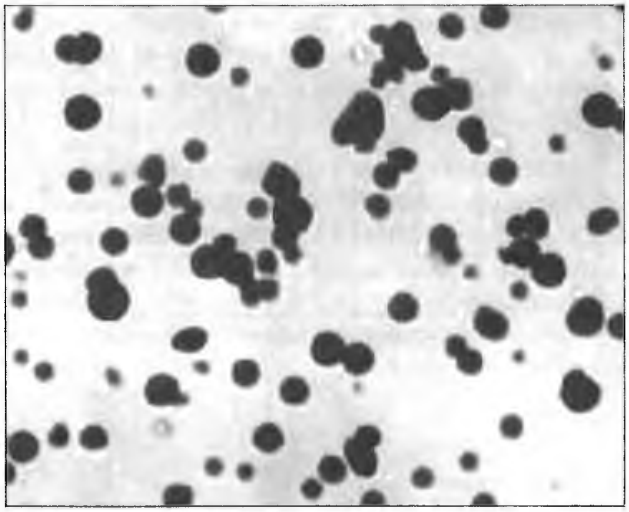


Figure 89. *Transmission electron microscope image of latex spheres in a thick, transparent section. Some of the spheres are partially hidden by others. If the section thickness is known, the size distribution and volume fraction occupied by the spheres can be estimated, but some small features may be entirely obscured and cannot be determined.*

Classification and recognition is generally considered to be the high-end task for computer-based image analysis. It ranges in complexity from locating and recognizing isolated objects belonging to a few well-established classes to much more open-ended problems. Examples of the former are locating objects for robotic manipulation or recognizing targets in surveillance photos. An example of the latter is medical diagnosis, in which much of the important information comes from sources other than the image itself. Fuzzy logic, expert systems, and neural nets are all being applied to these tasks with some success. Extracting the correct information from the image to feed the decision-making process is more complicated than simple processing or measurement, because the best algorithms for a specific application must be determined as part of the logic process.

These tasks all require the computer-based image processing and analysis system, and by inference the image acquisition hardware, to duplicate some operations of the human visual system. In many cases they do so in ways that copy the algorithms we believe are used in vision, but in others quite different approaches are used. Although no computer-based image system can come close to duplicating the overall performance of human vision in its flexibility or speed, there are specific tasks at which the computer surpasses any human. It can detect many more imaging signals than just vis-

ible light; is unaffected by outside influences, fatigue or distraction; performs absolute measurements rather than relative comparisons; can transform images into other spaces that are beyond normal human experience (e.g., Fourier, Wavelet, or Hough space) to extract hidden data; and can apply statistical techniques to see through the chaotic and noisy data that may be present to identify underlying trends and similarities.

These attributes have made computer-based image analysis an important tool in many diverse fields. The image scale may vary from the microscopic to the astronomical, with substantially the same operations used. For the use of images at these scales, see especially Inoue (1986) and Sabins (1987). Familiarity with computer methods also makes most users better observers of images, able to interpret unusual imaging modes (such as cross sections) that are not encountered in normal scenes, and conscious of both the gestalt and details of images and their visual response to them.

2

Printing and Storage

Printing and storage

Creating hardcopy representations of images, for example, to use as illustrations in reports, is important to many users of image processing equipment. It is also usually important to store the images so that they can be retrieved later, for instance to compare with new ones or to transmit to another worker. Both of these activities are necessary because it is rarely possible to reduce an image to a compact verbal description or a series of measurements that will communicate to someone else what we see or believe to be important in the image. In fact, it is often difficult to draw someone else's attention to the particular details or general structure that may be present in an image that we may feel are the significant characteristics present, based on our examination of that image and many more. Faced with the inability to find descriptive words or numbers, we resort to passing a representation of the image on, perhaps with some annotation. Arlo Guthrie describes this procedure in his song, "Alice's Restaurant," as "twenty-seven 8 by 10 color glossy pictures with circles and arrows and a paragraph on the back of each one."

Printing

This book is printed in color, using high-end printing technology not normally available to a single image processing user. But many everyday jobs can be handled quite well using quite inexpensive machines; the quality, speed, and cost of both monochrome and color printers are improving rapidly. A typical monochrome (black on white) laser printer costs about a thousand dollars and has become a common accessory to desktop computer systems. These printers are designed primarily to print text, and simple graphics such as line drawings. Most can, however, be used to print images as well. We have come a very long way since computer graphics consisted of printing Christmas posters using Xs and Os on a teletype to represent different grey levels (**Figure 1**). In this chapter, we will examine the technology for printing images that can be used in desktop computer-based image processing systems.

For this purpose, it does not matter whether or not the printers use a high level page description language such as PostScript®, which is used to produce smooth characters and lines at the maximum printer resolution, so long as they allow the computer to transmit to the printer an array of individual pixel brightness values. Most printers that can create any graphics output in addition to



Figure 1. *Portion of a 1960s-era teletype printout of a Christmas calendar poster.*

simply printing text can be controlled in this way. This means that "daisy wheel" or other formed-character printers (now virtually obsolete) are not useful for imaging. That is how the Snoopy posters were made, by creatively arranging to overprint groups of characters to produce different levels of darkness. But dot-matrix printers using inked ribbons, ink jet printers, thermal printers, and other devices that form their output as an array of dots on paper *can* be used to print images. The quality of the result is primarily a function of the size and spacing of the dots.

A basic level of confusion that often arises in interpreting the specifications of a printer in terms of image quality has to do with "dots per inch" (dpi) and "lines per inch" (lpi) (sometimes also called pixels per inch for even more confusion). For any of the printers mentioned previously, but particularly for laser printers, the specification of dpi resolution is the number of tiny black dots (or whatever color ink is used) that the printer can deposit on paper. Usually, it is the same in both the line (horizontal) and page (vertical) directions on the paper, although some printers have different dpi resolution in the two directions. Normally, the dots are used to form characters and lines. A low number of dpi will cause the characters to look rough and the lines to appear stair-stepped or "aliased." Resolution of 300 to 600 dpi or more for laser printers is now very common. It is capable of producing quite acceptable output for text and line drawings used in reports and correspondence.

However, the dots placed on the paper by these printers are black, and do not have an adjustable grey scale needed to print images. To create a grey scale for images, it is necessary to use groups of these dots, a technique generally known as halftoning. It is commonly used in newspapers, magazines, and books (including this one), and as we will see can be used for color as well as monochrome images. The differences in halftone quality between (for instance) a newspaper and a book lie fundamentally in the number of dots per inch that can be placed on the paper and the way they are organized to produce a grey scale (or color) result.

The basis of halftoning lies in the grouping of the individual black dots produced by the printer. A group of (for instance) 16 dots in a 4×4 array may be called a halftone cell. Within the cell, some or all of the dots may actually be printed. Where no dot is printed, the white paper shows through. If the cell is reasonably small, the observer will not see the individual dots but will instead

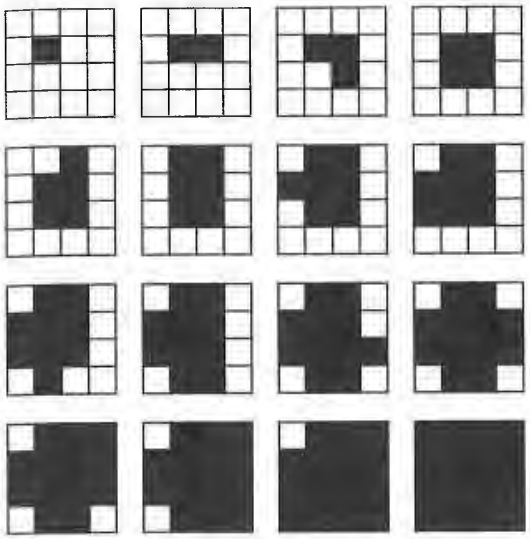


Figure 2. *Halftone grey scale produced with a 4 × 4 dot cell. Printing dots in all, some or none of the locations generates 17 different grey values.*

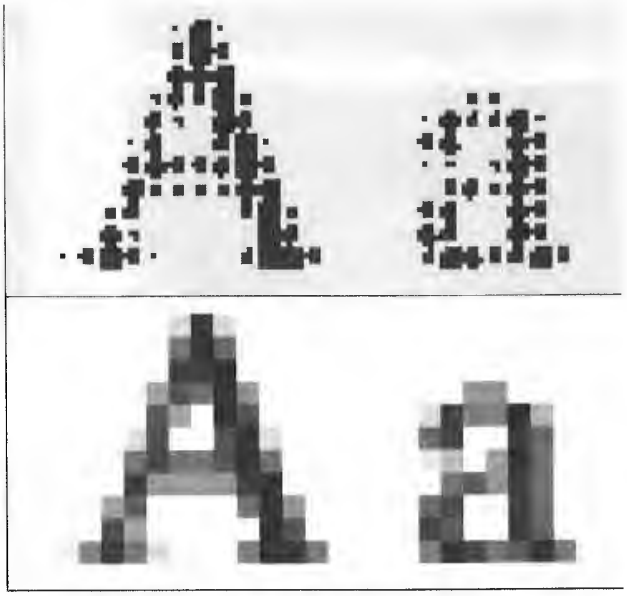


Figure 3. *Representation of a grey scale image by halftoning.*

visually average the amount of dark ink and light paper, to form a grey scale. In this example, shown in **Figure 2**, 17 possible levels of grey are available, ranging from solid black (all dots printed) to solid white (no dots printed).

In a 300 dpi printer, the individual black dots can be placed on the paper with a spacing of 1/300th of an inch in each direction. Grouping these into 4 × 4 halftone cells would produce 300 ÷ 4 = 75 cells per inch. This is close to the resolution of pixels on a typical video display used with an image-processing computer. If each pixel corresponds to a halftone cell, then an image can be printed with about the same dimension as it appears on the screen. Each halftone cell uses one of its 17 possible grey levels to represent the grey scale of the pixel. **Figure 3** illustrates how these printed cells represent the grey levels in typical anti-aliased text.

Of course, because an original image might typically have 256 grey levels, this appears to be a rather poor representation of the brightness information; however, that is not the only nor even the most serious limitation. Instant prints from Polaroid® film show only about the same number of distinct grey levels (the film is quite a bit better than the print) and are considered quite useful for many scientific as well as casual purposes (for instance, the recording of scanning electron microscope images).



Figure 4. Printed halftone images of the same image using a 300-dpi Postscript laser printer: *(**a**) original image; (**b**) 100 lines per inch; (**c**) 75 lines per inch; (**d**) 50 lines per inch; (**e**) 32 lines per inch; (**f**) 25 lines per inch. Increasing the number of halftone cells improves the lateral resolution at the expense of the number of grey levels that can be shown.*

One problem with the halftoning method illustrated in **Figure 4** is that the dots and cells are large enough to be visually distinguished by the observer. In a magazine or book, the size of the cells is smaller. The cell size is usually described as a halftone screen or grid: the spacing of the screen in number of lines per inch corresponds to twice the number of cells per inch discussed above. A screen with well over 100 lines per inch (often 115 or 133 lines per inch) is used to produce high quality printed illustrations. Even in a newspaper illustration, a screen of at least 85 lines per inch is typically used. **Figure 4** shows several examples of halftone output from a typical laser printer, in which the number of halftone cells is varied to trade off grey scale versus lateral resolution. The output from the current generation of desktop laser printers is adequate for some reports, but not for publication.

Not only are more lines per inch of resolution desired to preserve the sharpness of features in the image, at the same time more grey levels must be represented by these more finely spaced cells. That means that the printer (or imagesetter, as these higher-quality devices are generally called) must be capable of placing a much larger number of very much smaller dots. A grey scale of 65 levels can be formed with an 8×8 array of dots in a cell. With a 125-line screen, this would correspond to $8 \times 125 = 1000$ dpi. This is about the starting point for typeset quality used to print images for commercial purposes. Color (introduced below) imposes additional restrictions that require even higher resolution (smaller dots).

An additional difficulty with the halftoning method outlined above arises from the dots themselves. Each of the various printing methods produces dots in a different way. Dot-matrix printers use small pins to press an inked ribbon against the paper. Ink-jet printers produce tiny ink droplets. Some of these printers deposit the ink in a liquid form that penetrates into the paper making slightly fuzzy dots, while in others the ink has solidified and adheres to the paper without penetration. The better grades of paper have surface coatings that prevent the ink from spreading or penetrating. Thermal printers use a pin to pass an electrical current through the coating on a paper. One kind of paper is coated with a white oxide of zinc that is reduced by the current to deposit a dark spot of metal at the location; other thermal papers are based on the chemistry of silver. Laser printers work essentially like a xerographic copier. The light from the laser (or in some versions from a photodiode) falls on a selenium-coated drum and by the photoelectric effect produces a localized electrostatic charge. This in turn picks up carbon particles (the toner or "ink"), which are then transferred to the paper and subsequently heated to remain permanently.

Dots on paper

All of these technologies are limited by the ability to make a small, dark spot on the paper. The size of the carbon particles used as the toner in copier or laser printer cartridges limits the spatial resolution, and special finely ground toner is needed for resolutions of 600 dpi and higher values. The limitation in making higher resolution laser printers is not primarily in the additional memory needed in the printer, nor by the need to focus the light to a smaller spot on the drum, but in the toner particle size. Some systems disperse the toner using liquid carriers to improve the control of toner placement.

Similar restrictions limit the other printing methods. The difficulty of depositing a small but dark ink spot by the impact of a pin onto a ribbon, or the fuzziness of the dot written by thermal printing, have prevented those techniques from advancing to higher resolutions. Ink-jet printers can generate small drops and hence deposit small dots, but the inks tend to spread on the paper. Indeed, the roughness of the paper surface, and the need for special coatings to prevent the inks from soaking into the paper fibers or spreading across the surface or the toner particles from falling off, become critical issues. It is not enough to purchase a high quality printer; the use of special paper

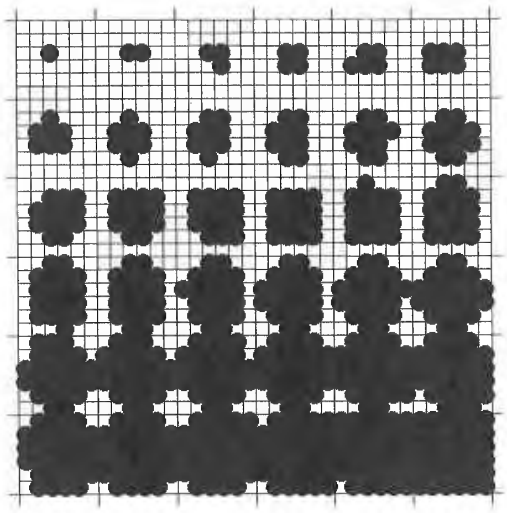


Figure 5. *A 6 × 6 dot halftone that can produce 37 grey levels. The use of approximately round dots large enough to touch diagonally causes them to overlap and produce darker cells than ideal.*

with a proper surface finish for the particular printer is needed to achieve the quality of image printing that the printer technology makes available (Lee and Winslow, 1993).

Because the dots produced by printers are generally imperfect and rough-edged, it is hard to control them so that the grey scale produced by the array of dots within the cell is uniform. Most dots are larger than their spacing so that solid black areas can be printed. This is good for printing text characters, which are intended to be solid black, although it means that the dots overlap some of the white areas in the cell, which darkens the halftone grey scale. **Figure 5** shows this for the case of a 6 × 6 dot halftone cell. At the dark end of the scale, adjacent grey levels may be indistinguishable, while at the light end the difference between the first few levels may be very great.

For the case of a 4 × 4 dot cell illustrated above using a 300 dpi printer, 17 nominal grey levels, and 75 cells per inch, the darkening or "gain" of the grey scale will produce images of marginal quality. If the printer resolution is much higher so that finer halftone cells with more steps can be created, it is possible to correct for this tendency to darken the images by constructing a mapping that translates the pixel grey value to a printed grey value that compensates for this effect. These "gamma" curves are applied within the software so that more-or-less equal steps of brightness can be printed on the page.

However, the human eye does not respond linearly to grey scale, but logarithmically. This means that to produce a printed image in which the visual impression of brightness varies linearly with pixel value, a further adjustment of the gamma curve is needed (as shown in **Figure 6**) to compress the dark values even more and expand the light ones. Because of these limitations, a printing scale with 65 grey values defined by an 8 × 8 dot halftone cell may be able to show only about half that many shades in the actual printed image.

Halftone grids in a typesetter or imagesetter are not limited by the size or perfection of the dots. Instead of coarse toner particles or contact with an inked ribbon, typesetters use light to expose a photographic emulsion, which is then developed to produce either a film or a print. The size of the silver grains in the film emulsion is far smaller than the effective dot size and the dots can be controlled in both size and shape with great precision. There is still a need for a gamma curve to compensate for the nonlinear response of human vision, however.

If the variation of brightness across an image is gradual and the total number of grey levels is small, it is possible to generate a visual effect known as banding or posterization. This is illustrated

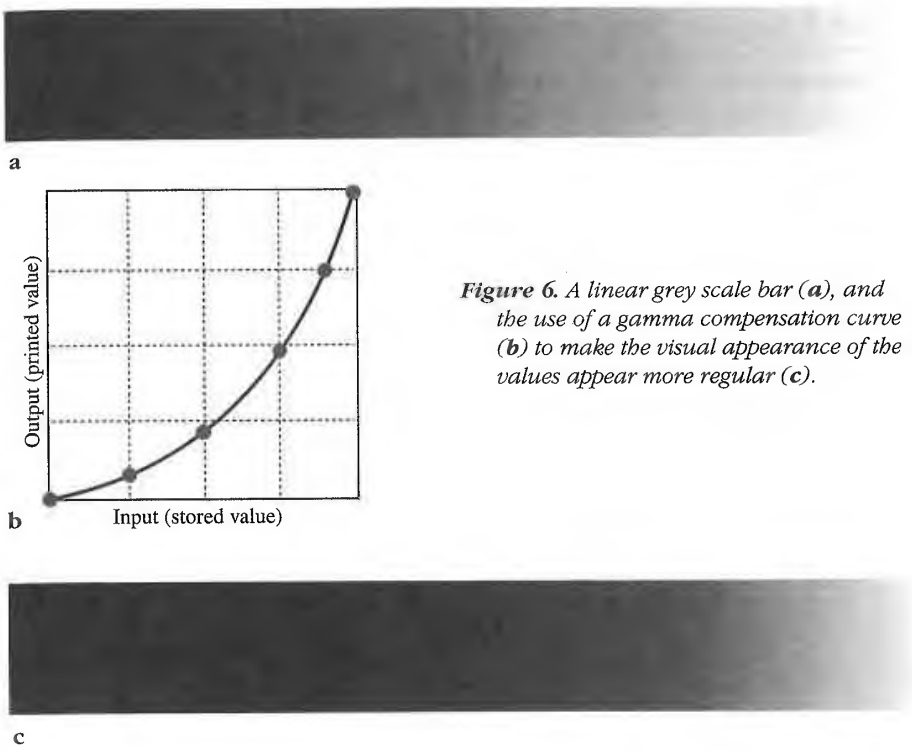


Figure 6. *A linear grey scale bar (**a**), and the use of a gamma compensation curve (**b**) to make the visual appearance of the values appear more regular (**c**).*

in **Figure 7**. The step from one brightness level to the next appears as a contour line in the printed image and is visually recognized as a feature in the picture, when in fact it is purely an artefact of the printing process. Banding can be avoided by increasing the number of grey levels used, or in some cases by modifying the way in which the dots within the halftone cell are used to generate the grey scale. Generally, the rule for the maximum number of grey shades available is $1 + (\text{dpi}/\text{lpi})^2$, where dpi is the printer resolution in dots per inch (e.g., 300 or 600 for a typical laser printer) and lpi is the lines per inch of the halftone screen (e.g., 75 lines per inch for the example discussed above). This assumes that all of the grey levels can actually be used, subject to the darkening ("dot gain") and gamma effects mentioned above.

In the examples shown in **Figures 2** and **3**, an arrangement of dots was used that produced more-or-less round regions of black within a white frame. With a larger dot array in each cell, an even more regular circular pattern can be constructed. The round dot pattern is one of the more

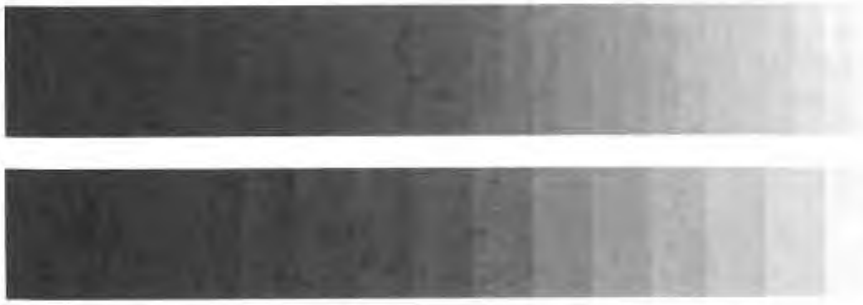


Figure 7. *A grey scale bar showing banding when printed with only 32 or 16 grey steps.*

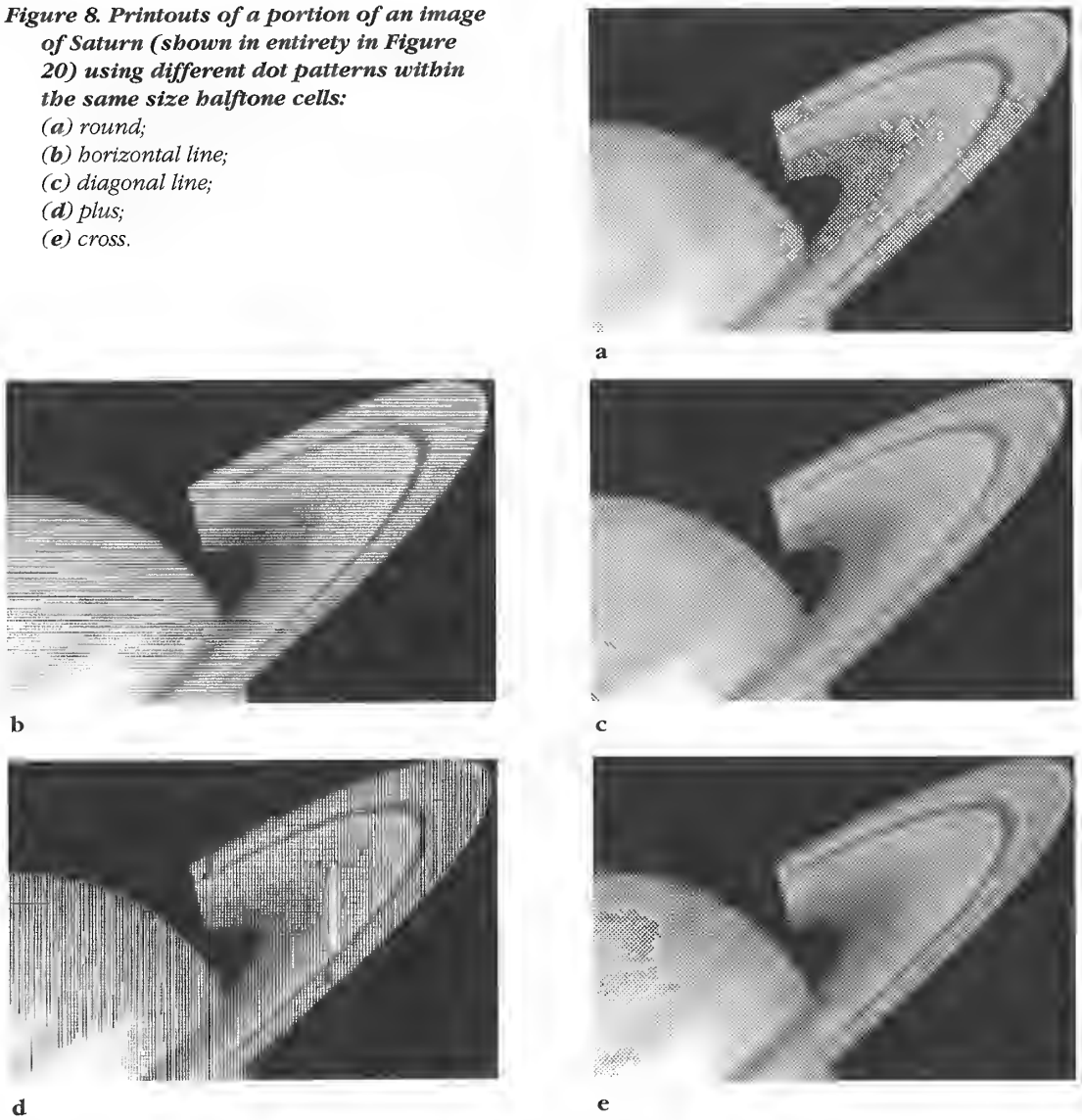


Figure 8. Printouts of a portion of an image of Saturn (shown in entirety in Figure 20) using different dot patterns within the same size halftone cells:
*(**a**) round;*
*(**b**) horizontal line;*
*(**c**) diagonal line;*
*(**d**) plus;*
*(**e**) cross.*

commonly used arrangements; however, many others are possible, including lines and crosses (**Figure 8**). Each of these produces some visual artefacts that may be useful for artistic effects but generally interfere with viewing the actual image contents. For instance, the diamond pattern used by most Postscript printers as an approximation to round dots causes dot densities less than 50% black to appear quite different from ones that are 50% or more black. The reason for this is that at the 50% point the dots touch from one cell to the next so that the eye perceives a sudden transition from dark dots on a white background to the reverse, even if the individual dots are not visually evident. Of course, all such artefacts degrade the representation of the original grey scale image.

If the dots within each halftone cell are randomly distributed, fewer visual artefacts are likely. However, this requires additional computer processing within the printer and is not a commonly available feature. Other printers use programs that analyze the neighbor cells (pixels). If cells on one side are dark and on the other side are light, the program may use a pattern of dots that is shifted toward the dark neighbors to produce a smoother edge. This works quite well for

generating smooth boundaries for printed text characters and minimizing jagged edges on lines, but it is not clear that it provides any consistent benefit for grey scale image printing.

Printers of modest dpi resolution cannot produce halftone cells that are small and still represent many grey levels, and visually distracting patterns may appear in images if regular patterns of dots are used within the cells. Hence, another approach used with these printers is to use "dithering" to represent the image. In a simple random dither, each possible position where the printer can place a black dot corresponds to a point in the image whose grey scale value is examined. If the point is very dark, there



Figure 9. Two different dither patterns for the "girl" image (shown in Chapter 1, Figure 25):
*(**a**) random dots;*
*(**b**) patterned dots.*

is a high probability that the dot should be printed (and vice versa). A random number is generated and compared to the grey scale; if the number is below the grey scale value for that point (both properly scaled), then the dot is printed on the page.

As shown in **Figure 9**, even coarse dithering can produce a viewable representation of an image. Many different types of dithering patterns are used, some of them random or pseudo-random and others with various patterned arrangements of the dots. Usually, the particular dither pattern used is determined either by the printer itself or the interface software provided for it. There is a rich literature on the design of dither patterns, but the user of the image analysis system may not have much control or choice of them. Dithered images are generally good at showing gradually varying brightness gradients, since human vision responds to an average dot density or spacing. But sharp edges will be blurred, broken up, or displaced because there is no continuous line or boundary printed, and it is difficult to compare the brightness of regions within the image. Dithered printing is usually considered to be a "low-end" method for producing crude hard copy and not suitable for use in reports or publications.

Despite the limitations discussed previously, monochrome image printing using a halftoning method and a printer with 300–600 dpi capability is adequate for many kinds of image processing applications and the resulting images are suitable in quality for reports. For higher quality work, photographic recording can be used. This process may seem like a step backward, especially when the image may have started out on film in the first place before it was digitized into the computer. But because facilities for handling photographic images, duplicating them, and making halftone screens photographically for printing are well developed, fully understood, and comparatively inexpensive, this is often the most effective solution. Forcing a "high-tech" solution in which images are directly merged into a report within the computer may not be worthwhile in terms of time or cost if only a small number of copies are needed. Mounting photographs on a few pages or inserting them as separate pages is still a quite suitable presentation method. No doubt as printing technology continues to advance, the balance will continue to shift toward direct printing from the computer.

Color printing

Printing color images is much more complicated and difficult. The usual method, which you can see by looking at the color images in this book with a magnifier, is to create halftones for each of several different color inks and superimpose them to produce the printed color image. Many complexities are involved in this process. The following section discusses the complexities that are important for most networked or desktop color printers.

Displaying a color image on a computer monitor or television set is accomplished by illuminating red, green, and blue phosphors with the electron beam in the cathode ray tube (CRT). There are exceptions, of course, the most important being the flat-panel liquid crystal (LCD) displays used for projectors and notebook computers. There are also large-screen displays (such as used at sporting events) that consist of arrays of discrete lights. Each of these methods generate the colors in different ways, but it is still the visual mixing together of red, green, and blue that produces the full range of colors that can be shown.

This range is called the "gamut" of the device (Stone et al., 1988). In a display that emits red, green, and blue light (RGB), it is possible to generate a large fraction of the total range of colors that the human eye can see. **Figure 10** shows the CIE color diagram that was introduced in Chapter 1, adding the typical color gamut for an RGB display and an ink-jet color printer superimposed on the diagram. One of the features of the CIE diagram is that colors add together along straight lines, so the three phosphor colors define the corners of a triangle that enclose all of the possible

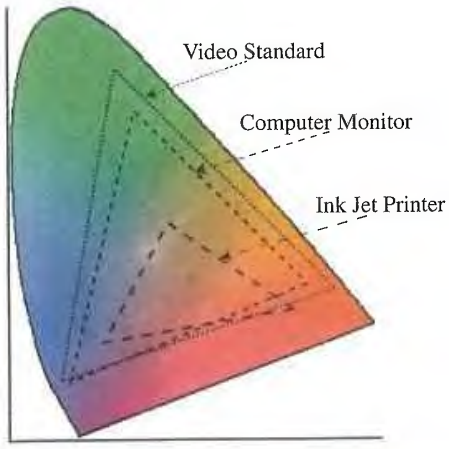


Figure 10. *The CIE color diagram discussed in Chapter 1 with gamuts shown for broadcast television, a typical computer monitor, and a typical ink-jet printer.*

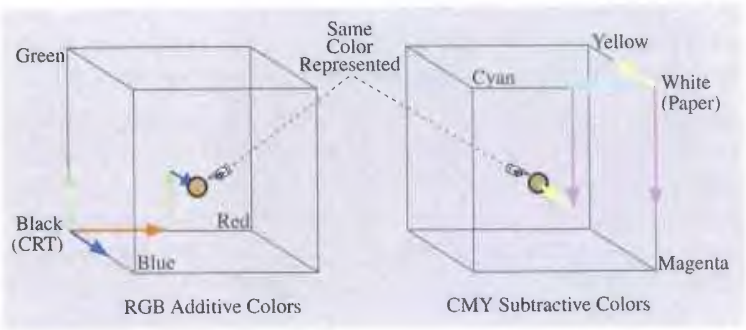


Figure 11. *Comparison of RGB (additive) and CMY (subtractive) color spaces. The additive color space adds red, green, and blue emissive colors to a black background, while subtractive color space removes cyan, magenta, and yellow colors from a white background.*

color combinations. The missing colors that cannot be generated using these particular phosphors include points near the outside of the diagram, which are the most saturated colors.

The concept of the color gamut applies equally well to other color display and recording devices. However, the gamut is not always as simple in shape as the triangle shown in the figure. For one thing, printing uses subtractive rather than additive color. In terms of the color cube (also introduced in Chapter 1 and shown in **Figure 11**) the blank paper starts off as white and the addition of cyan, magenta, and yellow inks removes the complementary colors (red, green, and blue,

Figure 12. *Combinations of RGB additive colors on a black background, and CMY subtractive colors on a white background.*



respectively) from the reflected light to produce the possible range of colors. It is common to call these colors CMY to distinguish them from the additive RGB colors.

The theory of subtractive printing, summarized in **Figure 12**, suggests that the mixture of all three colors will produce black (just as with additive RGB color, the summation of all three colors produces white); however, actual printing with CMY inks generally cannot produce a very good black, but instead gives a muddy greyish brown because of impurities in the inks, reflection of light from the surface of the ink (so that it does not pass through all of the ink layers to be absorbed), and difficulty in getting complete coverage of the white paper.

The most common solution is to add a separate, black ink to the printing process. This reduces the need for large amounts of the colored inks, reduces the thickness of ink buildup on the page, and reduces cost. But from the standpoint of image quality, the most important factor is that it allows dark colors to be printed without appearing muddy. The four-color, cyan-magenta-yellow-black system (CMYK) uses four halftone screens, one for each ink, but although converting from RGB colors (or CIE, HSI, or any other equivalent color space coordinate system) to CMY (often called process color) is straightforward, converting to CMYK is not. Rules to calculate how much black to put into various color representations depend on visual response to colors, the kind of paper to be printed, the illumination the print will be viewed with, and even the contents of the images.

Algorithms for converting from CMY to CMYK involve specifying levels of undercolor removal (UCR) and grey component replacement (GCR) which are essentially arbitrary, are little documented, and vary considerably from one software program to another (Agfa, 1992). The general approach is to use the value of whichever of the three components (CMY) is darkest to determine an amount of black to be added. For instance, for a color containing 80% cyan, 50% magenta, and 30% yellow, the 30% value would be taken as an index into a built-in calibration curve or lookup table. This might indicate that a 15% value for the black ink should be chosen for grey component replacement. Then, the amount of the principal color (in this example cyan), or, according to some algorithms, the amounts of all of the colors, would be reduced (grey component replacement). It is difficult to design algorithms for these substitutions that do not cause color shifts. Also, the substitutions do not work equally well for printing with different combinations of ink, paper finish, etc.

Color prints are generally not as vivid or saturated as the image appears on the CRT. In addition, the colors depend critically on the paper finish and viewing conditions. Changing the room light will slightly alter the visual appearance of colors on an RGB monitor, but because the monitor is generating its own illumination this is a secondary effect. Since a print is viewed by reflected light, changing the amount of light or the color temperature of room light with a print can completely alter the appearance of the image. The color temperature (a handy way of describing the spectrum of intensity vs. color) of incandescent bulbs, fluorescent bulbs, direct sunlight, or open sky are all quite different.

Human vision can be tricked by combinations of illumination and shadow, inks or other colored coatings, surface finish (smooth, or textured in various ways), and even the presence of other adjacent colors in the field of view to change the way we judge colors in an image. These other colors may even lie outside the image itself; consider how a colored mat can change the appearance of an art print. When "true" color prints of an image are required, it is necessary to perform extensive calibrations of a specific printer and monitor so that acceptable fidelity is obtained. This is of great concern in advertising; if you purchase clothing from a mail order catalog, you expect the colors of the cloth to match the printed photograph, which is no easy task. The process usually works by adjusting the monitor output (with lookup tables in the display hardware) so that the appearance of colors there is tailored to match the ability of a specific printer/paper/ink combination to reproduce them.

For most (but certainly not all) applications of image processing, the purpose of printing in color is to distinguish the variously colored regions present; some inaccuracy in the fidelity of the colors is acceptable. If exact color matching is not necessary in a particular application then the task becomes much easier, although you will still need to be concerned about the color gamut of the printer, the consistency of colors (to allow comparison of different images or regions), and of course the resolution of the printer. The color gamut is important because colors of increasing saturation in the original image, which can be distinguished on the video display, may become similar in the print image if the saturation exceeds the range of the printer.

Producing CMYK halftones (so-called color separations) and superimposing them to produce color prints sounds simple enough, but it harbors many problems that commercial printers must deal with routinely. One is how to superimpose the halftone screens so that they do not produce visual

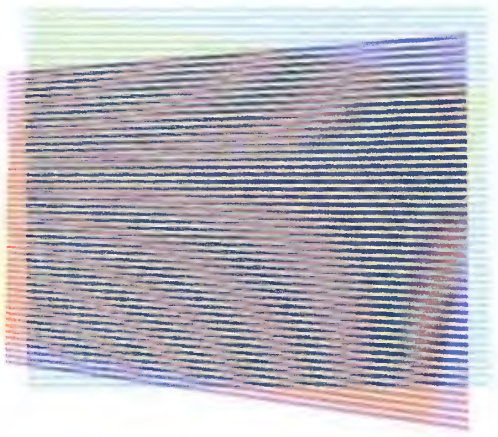


Figure 13. *Moiré pattern produced by overlaying arrays of color dots.*

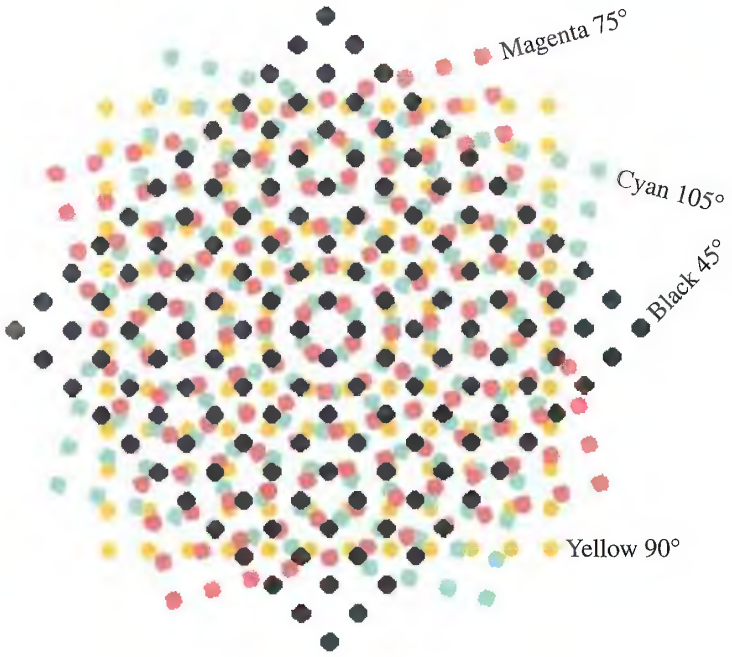


Figure 14. *Ideal screen angles for CMYK color printing place the halftone screens at angles of 45° (black), 75° (magenta), 90° (yellow), and 105° (cyan).*

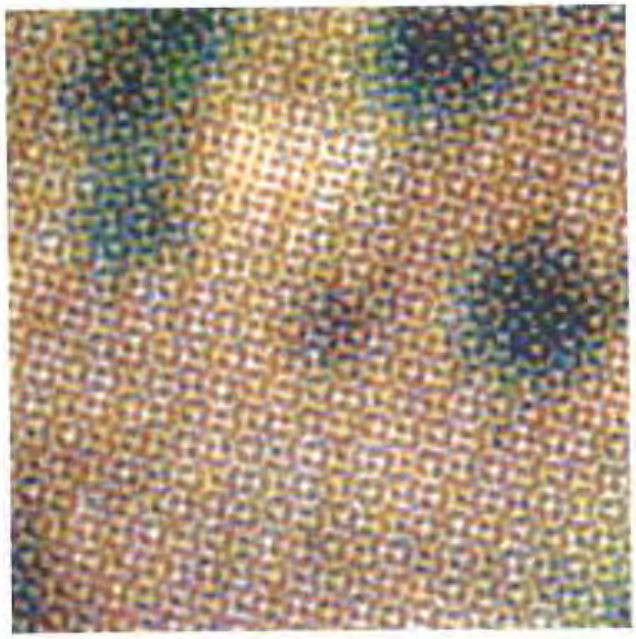


Figure 15. *Example of color halftone image printed in an earlier edition of this textbook.*

moiré patterns in the image. **Figure 13** shows an example of a moiré effect produced by different color patterns. The lines that appear are an artefact of the spacing and alignment of the individual patterns. Traditional offset printing defines "ideal" screen angles for the four CMYK screens as 45, 70, 90 and 105° respectively, as shown in **Figure 14**. This aligns the colored dots to form small rosettes that together make up the color. **Figure 15** shows an example; note that the dots vary in size to control the amount of each color, and that some of the halftone spots may be partly superimposed on each other. Because some inks are partially transparent, this can produce various color biases depending on the order of printing.

In most printing, the colors are not intended to be superimposed on each other, but printed adjacent to each other, and to allow some white paper to show. Most color printing methods suffer from the fact that as more color is added, the white paper is covered and the appearance becomes darker. This is another reason that high-resolution halftone screens and very small printing dots must be used. Generally, when four halftone grids or color separations are used, the dpi of the printer must be doubled for each color to get an equivalent printing resolution to monochrome printing. In other words, each pixel now requires four interleaved halftone cells, one for each color. Because the dot size for each of the four colors is only one-fourth of the area, printing a solid expanse of one color is not possible, which further reduces the maximum saturation that can be achieved and hence the gamut of the printer.

Most desktop printers do not provide control over the screen angles, and many printers simply use a zero angle for all four screens due to the way the printing mechanism works (for instance, a typical color ink-jet printer). Some page description languages (e.g., PostScript Level 2) include provision for such control, but only high-end typesetters and imagesetters generally respond to those commands.

All of the problems present in printing monochrome or grey scale images, such as banding, posterization, limited numbers of intensity levels, and so forth, are also present in producing the color separations. In addition, color printing introduces the additional problems of screen angles and moiré patterns. Finally, alignment and registration of the colors must be considered, since in many printers they are printed one at a time and the paper must be handled in several passes. Some

alternative printing methods exist that deposit multiple colors in a single pass, but these have their own problems of allowing inks to dry without mixing.

The printing and registration of colors presents one additional problem in some cases. Consider a region of a uniform color that is a mixture of two or more of the primary printed colors and is adjacent to a region of a different color. At the boundary, the colors must change abruptly from one color to the other; however, the two-color screens are not aligned dot for dot, and along the boundary there will be cases in which one of the two left-side color dots may be present close to one of the two right-side color dots. This can give rise to false color lines along boundaries. In graphic arts, such artefacts are usually avoided by trapping to remove the colors at the boundaries in a particular order, or covering up the boundary with a black line. Obviously, such tricks are not available in printing real images where every pixel may be different.

Printing hardware

The discussion so far of converting the image to CMY or CMYK values and superimposing them as separate halftone screens on the printed page has ignored the physical ways that such color printing is performed (Kang, 1997, 1999). Few computer users will have ready access to imagesetters and offset printing presses that have high enough resolution to produce such high quality results. There is a considerable variation in the quality, cost, and performance of different types of desktop printers. Some of the more common currently available methods include ink jet, thermal wax, dye sublimation, and color laser printers. This section will offer some comparison of these technologies, recognizing that continuous progress in print quality is being made as many vendors compete for their slice of a rapidly growing market. In all of these comparisons it is important to keep cost in mind. Many of the printers themselves have dropped significantly in price, but the cost of consumables — paper, ink, etc. — keep the overall cost of each copy quite high for most of these devices.

To assist in comparing the various printer types, **Figure 16** shows a portion of an image printed using several of the technologies discussed in the following section.

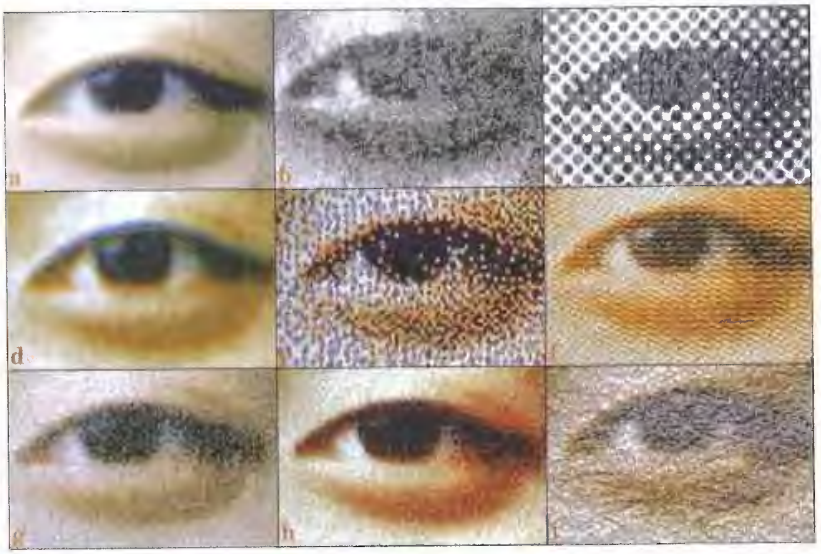


Figure 16. Magnified (30x) views of the same image printed using different devices: *(**a**) photograph; (**b**) Xerox copy; (**c**) black and white laser printer; (**d**) dye-sublimation printer; (**e**) phase change solid ink printer; (**f**) color laser printer; (**g**) ink jet printer in low-resolution mode; (**h**) ink jet printer in high-resolution mode; (**i**) ink jet printer using uncoated bond paper (images from M. Jurgens, Queens University).*

Color laser printers are the most economical for many applications, because they use plain paper and have relatively high speeds. The high initial cost of the printer is offset by very low cost per copy. They work by separating the image into CMY or CMYK colors and using each to control the illumination of a separate charged drum that picks up the corresponding color toner and deposits it on the paper. One major problem with these printers is the difficulty of keeping everything in alignment, so that the paper does not shrink or shift as it goes from one drum through a heater to bake the toner and on to the next drum. The use of a relatively thick paper with a smooth finish helps. In addition, the order in which the toners must be applied is critical, because with most models presently available the dots are not really laid down as a set of halftone screens at different angles (as discussed previously). Instead, they tend to overlay each other, which makes it difficult to achieve good color saturation or accurate color blending.

Figure 17 shows an enlarged example of a color laser printer output. Note that the saturated colors such as red are formed by combining two toner colors (red = blue + magenta) and give good coverage of the white paper, but more subtle colors are not so clear and may appear muddy. In fact, a plot of the gamut (maximum saturation limits for colors) of this type of printer is not too meaningful because the most saturated colors are rather well defined but the less saturated ones are not, and are more difficult to distinguish from each other. Color laser prints are generally a marginal choice for image reproduction, although they may be satisfactory for reports, or for computer graphics consisting primarily of saturated colors.

Dye-sublimation (or dye-diffusion) printers continue to represent the high end of image quality available in desktop printers. They require special coated paper (or transparency material). The inks or dyes are transferred one at a time from a master color sheet or ribbon to the paper, with a typical resolution of about 300 dpi. A portion of the color ribbon as large as the paper is consumed in making each print, even if only a small amount of the color is needed.

The low resolution value of 300 dpi, as compared with 1200+ dpi for most of the other types of color printers described here, does not indicate the actual print quality because the print is formed



Figure 17. *Enlarged view of color laser printer output, with areas of fully saturated and unsaturated colors.*

in a different way. Rather than assembling tiny printer dots to build up the color, these printers can control the amount of each color more directly. The amount of dye that is transferred from the ribbon to each pixel on the hardcopy is controlled by heating the print head to sublime or vaporize varying amounts of dye. A control to one part in 256 is typical, although the use of a corrective gamma function for each color to balance the colors so that the print appears similar to the screen image (so-called "color-matching") may reduce the effective dynamic range for each color to about 1 part in 100. Still, that means that about 1 million (100^3) different color combinations can be deposited at each point in the recorded image.

Dye-sublimation printers do not use a halftone or dither pattern. The dyes diffuse into the polyester coating on the paper and blend to produce continuous color scales. Each of the 300 printed regions per inch on the paper contains a uniform proportioned blend of the three or four dye colors. The lack of any halftone or dither pattern and the blending and spreading of the color dyes within the coating (instead of lying on the surface as inks do) produces an image that is very smooth and pleasing in its appearance. Magnification (**Figure 16d**) reveals the continuous tone and the just-visible pixelation pattern.

It is sometimes claimed that these printers are "near photographic" in output quality, but in several important respects that is not true. The saturation is less than a good photographic print, so that the color gamut is lower, and the 300 dpi resolution compounded by the spreading of the dye in the coating, produces much poorer sharpness than a photographic print (which can easily resolve several thousand points per inch). This spreading is somewhat worse along the direction of paper movement than across it, which can cause some distortion in images or make some edges less sharp than others. On the dye-sublimation print, large areas of uniform or gradually varying color will appear quite smooth because there is no dither or halftone to break them up, but edges and lines will appear fuzzy.

In addition to these technical limitations, the relatively high cost of dye-sublimation printers and of their materials has limited their use as dedicated printers for individual imaging systems. At the same time, their rather slow speed (because each print requires three or four separate passes with a different color ribbon or a ribbon with alternating color panels) has made them of limited use on networks or for making multiple copies. Still, they have been the preferred choice for making direct, hardcopy printouts of images. They are even finding use in commercial outlets, where consumers can insert their slides or color negatives into a scanner, select the image and mask it if desired, and then directly print out a hard copy to take home. The glossy finish of the coated paper probably causes many customers to think they have a traditional photographic print, although the effects of temperature, ozone, and humidity on useful life are much worse.

Some ***dry ink*** printers (also called phase change printers since the ink is melted and transferred to the paper) and thermal-wax printers are used for printing images, although they are best suited to computer graphics because of their bright colors and glossy results. They work well on overhead transparencies, which are fine for producing business graphics presentations, but are not usually an acceptable medium for image-processing output. The print head in a thermal-wax printer contains heating elements that melt the wax and transfer the color to the paper. This process is repeated for each color, which may cause alignment problems. In many cases, the dots are too large and the resulting number of shades for each color too small, and the resolution too poor, to show details well in images. The image appears as flat, shiny plates of ink pressed onto but not into the paper surface (**Figure 16e**).

Ink-jet printers are certainly at this time the most widely used devices for producing hard copies of images. The printers are quite inexpensive (although the consumable supplies are not), and the resolution is quite high (many current models offer 1200–1440 dpi). Ink jet printers are by far the most sensitive to paper quality among all of the technologies discussed here. Quality prints

Figure 18. Wicking of ink on an uncoated paper.

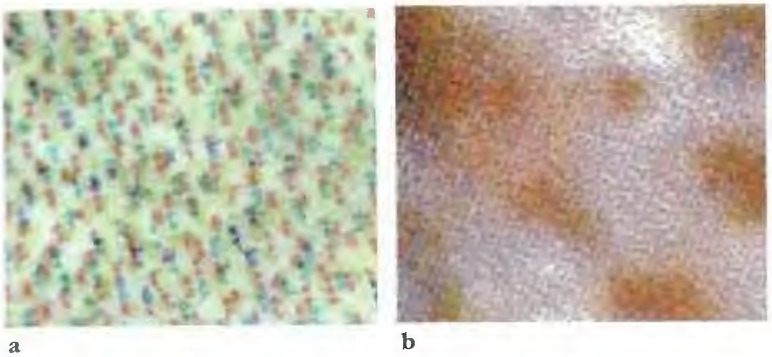


require a paper with both internal sizing and surface coatings to prevent the ink from wicking into the cellulose fibers (**Figure 18**), while promoting adhesion and providing a bright white background. Pigments such as $CaCO_3$ or TiO_2 also serve to whiten the paper. Surface coatings such as gelatin or starch are hydrophobic to prevent the ink from spreading, are smoother than the raw cellulose fibers, and may additionally be rolled (calendered) to provide an even more uniform surface. Inks must be carefully formulated with very tiny pigment particles (if these are used instead of dyes) to allow small droplets (the current state of the art is 4 picoliter drops for each tiny printer dot) to be uniformly and consistently transferred to the paper.

Longevity of the prints is strongly affected by exposure to ultraviolet light, humidity, temperature, ozone, acidity of the paper, etc. Some manufacturers now offer "archival" quality inks and papers, meaning that the paper has a high pH (usually due to added alkaline buffers to reduce acidity that causes discoloration or physical deterioration of the paper) and a high rag content (fibers other than cellulose), and the ink contains pigments that do not break down with exposure to UV radiation; however, it should not be expected that even these prints will survive well for long periods unless storage conditions are carefully controlled. Achieving acceptable lifetimes for computer-printed hardcopy continues to present challenges to equipment manufacturers.

Two types of ink are used in these printers, some containing pigment particles and others containing dyes. The inks containing pigment particles are less affected by paper variables (flatness, surface coatings, etc.) and suffer less environmental degradation by light, moisture, or ozone, or other airborne contaminants. Thus, they are often called "archival" inks, although the actual life of prints and the resistance to color shifts during storage still depend upon storage conditions. The pigment particles are not absorbed into the paper or its surface coatings, which makes them more sensitive to smearing or scratching. Dye-based inks, which are more common, dry quickly, and because they are more reflective than pigment-based inks (which produce color mainly by absorbing light) produce brighter colors and a broader range of colors.

In order to expand the gamut of the printers, and to improve the ability to render subtle shades such as skin tones, many ink jet printers now use more than the four CMYK inks. In a typical six-color system called CcMmYK, two less saturated cyan and magenta inks are available as well. Eight-color systems introduce less saturated yellow and black inks, and others may be used in addition. Algorithms for converting the stored RGB values for each pixel to appropriate blends of multiple inks are complex, proprietary, and highly ad hoc. Many ink-jet printers offer multiple quality levels. The faster settings also consume less ink, producing acceptable draft images with less saturated colors, while the slower settings are used for final copies.



a b

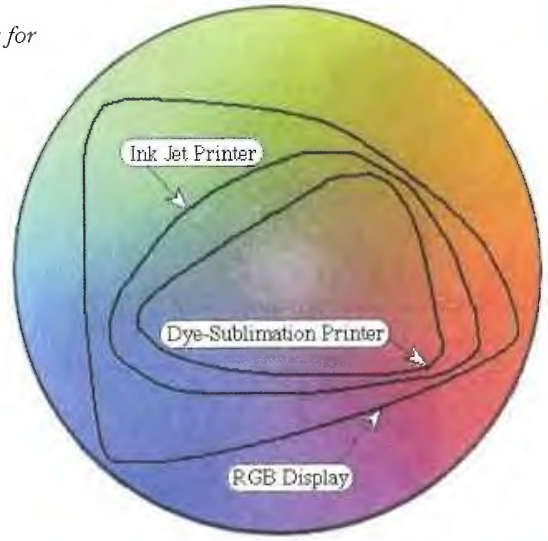
Figure 19. Ink-jet printer output (six-color CcMmYK inks): *(**a**) light area with well-defined separate dots; (**b**) superposition of more colors produces greater saturation but covers more white paper for a darker result.*

In light areas of printouts, the individual droplets can be seen (**Figure 19a**). Ideally these would all be perfectly round, identical size, well spaced color dots. In practice, they often deteriorate rapidly as printers age, are affected by temperature, and of course by the paper being used. When more color is present and more different ink colors are used, they superimpose on each other and cover the paper, producing a darker image (**Figure 19b**). As noted previously, this is the opposite behavior to the appearance of the image on the CRT display, where more color produces a brighter pixel.

As noted earlier, the gamut of a printer is usually described by an outline on a color wheel or CIE diagram of the maximum color saturations achievable, as shown again in **Figure 20**. The actual situation is much more complicated because saturation varies with brightness as mentioned above. By printing out swatches covering the entire three-dimensional color space, it is possible to construct a more complete representation of a printer's gamut. **Figure 21** shows an example, for an Epson six-color ink-jet printer, using the L·a·b representation of color space described previously. Printers with multiple inks can produce thousands of different hues, and modulate colorant density to produce high tonal resolution and subtle difference in color; however, notice that, as with all such gamut plots, certain colors cannot be represented on the hard copy.

Ink jet printers can also be equipped with a selection of grey inks, which when controlled by appropriate software can do an extremely good job of printing grey scale images. By selecting

Figure 20. *Color wheel comparing typical gamuts for ink-jet and dye sublimation printers.*



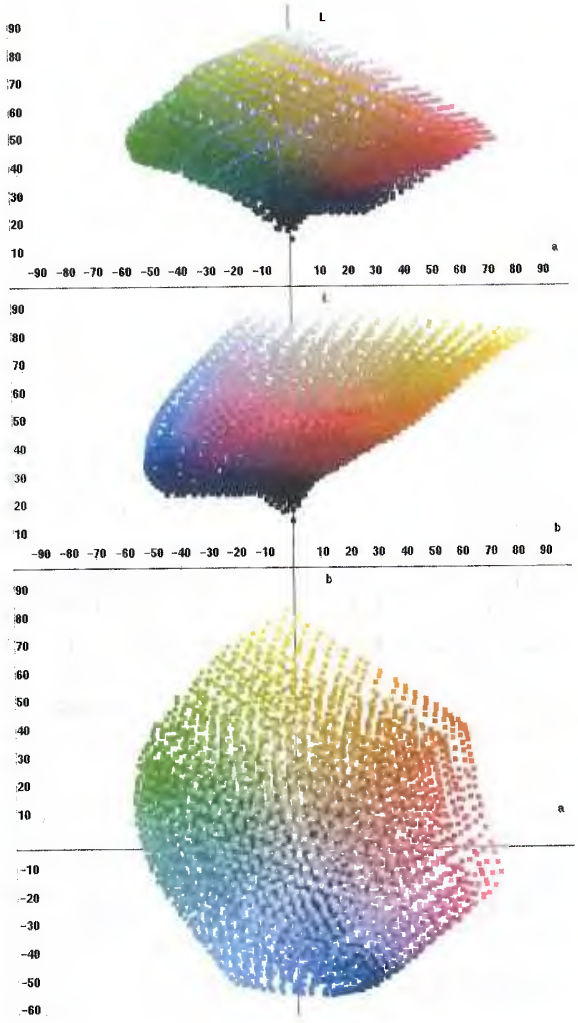


Figure 21. *Three-dimensional gamut for a six-color CcMmYK ink jet printer, displayed as cross sections through L·a·b color space (From Rezanaka, I. and Eschbach, R., Eds. (1996).* Recent Progress in Ink-Jet Technologies. *Soc. Imaging Science and Tech., Springfield, VA. With permission.)*

different ink densities as well as controlling the placement of tiny drops, a grey scale can be constructed that is very similar to photographic prints, and with a very acceptable spatial resolution. Selection of paper with good surface finish is still important.

Film recorders

In many cases, especially for color images, printing directly from the computer to paper is expensive and the quality limited. But the image looks good on the computer screen and it is tempting just to photograph it from there to create slides or prints using conventional film methods. Certainly, this is convenient. I have hundreds of slides that I once used in lectures (before the advent of affordable compact projectors that connected directly to my laptop computer allowed for more convenient and higher quality presentations) that were produced in exactly this way. I kept a 35-mm camera loaded with slide film on a tripod near my desk. When there was something on the screen that might be useful as a slide, I just moved the camera into position and, with a permanent exposure setting of f8 at 1/2 second (for ASA 100 slide film), snapped the picture. You do need to make sure that there are no reflections of windows or room lights on the monitor. When the roll was finished, I sent it off and got back a box of quite useful slides. I have done this enough times

to know that there is no need even to bracket the exposures. The exposure setting of f8 is about optimal for lens sharpness and gives plenty of depth of field for the slight curvature of the computer monitor, while 1/2 second is long enough to capture enough full raster scans so that no visible line shows where the scan was when the shutter opened or closed.

Although these pictures are fine for lecture slides, they are not useful for many other purposes. It is always obvious that they are photographs of the computer screen and not the actual images that have been captured or processed. This is due partly to the fact that they usually include the menu bars, cursor, and other open windows on the screen. It is possible to eliminate those, and many image-processing programs have a "photo mode" that displays just the image with the rest of the screen dark.

Unfortunately, other clues reveal how such an image is recorded. It is difficult to align the camera so that it is exactly perpendicular to the center of the screen. This misalignment produces a distortion of the image. In addition, the curvature of the screen causes a pincushion distortion of the image that is hard to overcome, and even with flat screen displays there tends to be vignetting. Special cameras intended for recording images from computer monitors have hoods to shield against stray light and align the camera lens, and may incorporate corrective optics to correct some of the distortions and minimize these problems. Still, other more serious difficulties remain.

Most color displays use an array of red, green, and blue phosphors arranged as triads of round dots. The electron beams are focused onto the phosphors and further restricted by an aperture screen (shadow mask) that prevents the electrons from straying from one dot to another. The other approach to monitor design, used in the Sony Trinitron, has a single electron gun, no shadow mask and the color phosphor arranged in vertical stripes. This design provides brighter images, since more of the screen is covered with phosphor and there is no shadow mask. On the other hand, the colors are less pure and lines or edges less sharp depending on their orientation. Varying the intensity of the electron beam controls the relative brightness of the phosphors and the resulting color that is perceived by the human observer, who blends the individual dots together. Modern computer displays have color triads with a spacing of about 0.25 mm. Human vision does not resolve these at normal viewing distances, so the colors are visually blended.

The image is made up of a regular pattern of color dots that the film records, perhaps with some systematic distortion due to the screen curvature and camera position. This pattern may be visible when a slide is projected onto a large screen. There is a temptation to use these films to prepare prints for publication, which reveals their major flaw. When the prints are subsequently converted to halftones, the array of dots photographed from the screen will interact with the screen of the halftone to produce a moiré pattern that can be quite objectionable.

Film has high resolution, good color saturation and dynamic range, and inexpensive processing, which make it an excellent recording medium. But photographing a color monitor is not the right way to record the image. Dedicated film recorders solve the problems mentioned above by using a monochrome CRT with a continuous phosphor and a flat face. The continuous phosphor gives a very high resolution with no structure to cause moiré patterns, while the flat face eliminates distortions. With only a single electron beam, sharp focus and uniform intensity can be maintained from center to edges much better than with video displays; the fixed optics also provide sharp focus across the image with no vignetting or distortion.

A motorized filter wheel containing red, green, and blue filters is placed between the CRT and the camera. Software much like that used to drive a printer is used to separate the image into red, green, and blue components and these are displayed one at a time on the CRT with the appropriate filter in place. The much higher display resolution allows entire images to be displayed with full pixel resolution, which is not usually possible on the computer monitor (unless you have a very large one). The filter densities are adjusted to balance the film sensitivity. All three color

components are recorded on the same frame of the film. With most recorders, the entire process is automatic, including film transport and compensation for different film speeds and sensitivities (for instance, slide or print film).

The only drawbacks are the exposure time of several seconds and the delay until the film is developed, and of course the cost of the entire unit. The cost varies as a function of resolution (from 2000 dots horizontally up to as much as 16,000 dots), and the flexibility to handle different sizes and types of film. High-end recorders are used to expose movie film frame by frame to produce movies containing "virtual reality" and rendered computer images (e.g., *Toy Story*), or to restore old classics by removing dirt and making color corrections (e.g., *Snow White*). Recorders that are price-competitive with color printers can be used for routine production of 35-mm slides. Nevertheless, the use of such recorders has not become widespread, and there is still a general preference for printing images directly onto paper.

File storage

When working on a computer-based, image-analysis system, images can be saved as disk files. There seems to be no reason to consider these files as being any different from other disk files, which may contain text, drawings, or programs. In one sense this is true: files contain a collection of bytes that can represent any of those things as well as images. But from a practical point of view, there are several reasons for treating image files as distinct, because they may require somewhat different storage considerations:

1. Image files are usually large. In terms of computer storage, the old adage that a picture is worth one thousand words is clearly a vast understatement. A single video frame (typically 640 × 480 pixels) in monochrome occupies about 300 kilobytes (KB), while one in full color requires about 1 megabyte (MB). A series of images forming a time sequence, or a three-dimensional array of voxel data (which can be considered as a series of planes) can be far larger. A 500 × 500 × 500 voxel tomographic reconstruction requires 125 MB, or twice that if the density values for the voxels have a dynamic range that exceeds 256 levels. This means that the storage capacity used must be large, preferably open-ended by allowing some kind of removable media, and reasonably fast. It also increases the interest in storage methods that utilize compression to reduce file size.
2. Imaging usually requires saving a large number of files. This reinforces the requirement for large amounts of fast storage, but it also means that access to the stored images will be needed. Constructing a database that can access images in a variety of ways, including showing the user small thumbnail representations, using keywords, and other indexing tools, is an important need that has been recognized by many software producers. Automatically extracting classification data from the image to assist in searching is a more difficult problem.
3. Data management in the era of computers has not yet fully exploited the possibilities of coping with relatively straightforward records that contain text and numbers. The "file cabinet" metaphor is a poor and limited one. Search engines on the internet compile lists of the words used on trillions of sites so they can produce fast results, but everyone who has used them knows that the process finds many extraneous sites and misses many of interest. The real issue for computerized file storage is access to files and finding documents. Adding a field to a database primarily constructed to hold text entries does not make it into an imaging database. For instance, keeping a picture of each employee in a personnel file may be worthwhile, but it would hardly allow a user to locate employees by knowing what they looked like. That would require looking at every image.

One type of database that involves images works in essentially the opposite direction. A geographical information system (GIS) stores multiple images and maps. These record different kinds of information, which is keyed to locations. There may also be text records keyed to those

locations. By overlaying and correlating the multiple map representations, it becomes possible to compare the locations of features on one map (e.g., roads or buildings) with those on others (types of soil or vegetation, underground aquifers, etc.). There are important issues to resolve in constructing a GIS, since the various maps and images (including aerial and satellite photos) must be aligned and registered, taking into account their different resolutions. In addition, displaying such rich multidimensional data presents challenges beyond those faced in most image databases.

There may be some other factors to consider in designing an optimum system for image storage. For instance, in forensic, medical and some other applications, it may be important to keep a record of all accesses to the image and of any processing or editing steps that may be applied. Also, some standardization of storage formats is important if images are to be shared between users or between different systems. Some users may want to archive images essentially forever but rarely access any particular image, while others may make continued and repeated accesses but be willing to discard images after a certain time (or when a project is completed). The data may be intended for access by many users, over a network, or be restricted to a single user. If a set of images is to be distributed widely as a permanent and unalterable resource, the requirements change again.

There are some beginnings of solutions to most of these problems and indications that many more will be forthcoming. The introduction of the Photo-CD format by Kodak indicates their expectation of considerable consumer usage of this type of storage. It is easy to imagine someone trying to find an old picture of "Aunt Sarah," for example. If the photo is stored as a faded print in an old shoe box, the searcher may have a rough idea of where to look in the attic for the box and then be willing to sort through the contents looking for the picture. But in the era of computer technology, it will seem natural that somehow the computer can find the picture just by being asked for it. Obviously, that is not a trivial task, but many companies are trying to respond to the challenge.

Storage media

Kodak's Photo-CD is an example of one kind of media that is being used for image storage. Actually, we will see in the following paragraphs that Photo-CD also defines a storage format, not just the use of CD disks; it is possible to store images on CDs in other formats as well. Projects as diverse as archaeological digs and remote sensing (satellite imagery) have found it convenient and inexpensive to distribute collections of images on CDs as a reference resource. The CD (Compact Disk) is just a plastic platter with reflective dots imprinted onto it, physically identical to audio CDs. The format of the dots, which mark changes from 1s to 0s rather than the actual bit values, is shown in **Figure 22**.

The reflective dots on a CD duplicated in quantity are physically imprinted in a pressing operation and covered by a hard plastic that is quite resistant to damage. This technique lends itself to

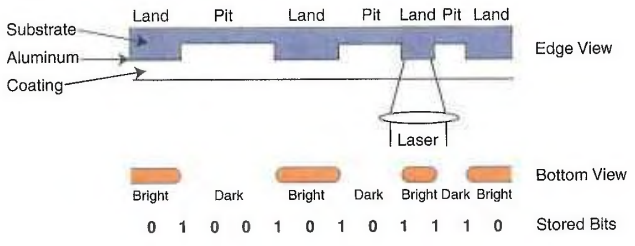


Figure 22. *Storage format for CD data. Pits and lands in the aluminum layer reflect laser light, with the pits scattering the light and producing a darker reflection. Transitions from bright to dark indicate ones and places where no transition occurs indicate zeroes.*

creating many copies of a disk inexpensively. Current prices for making duplicates from a master are less than a dollar per copy and continue to decline. Distributing a library of images in this way makes sense, as does distributing software, etc., because practically every modern computer has a reader for these disks.

Data on CDs are written on one single continuous spiral track, in blocks of 2048 bytes (plus the block address and some error detection sums). The entire track is written at one time, which requires the data to be continuously supplied to the write head as the disk spins. The CD format is much more difficult to use as a random-access device for either writing or reading than a conventional disk with its multiple circular tracks and fixed sectors, which make locating particular blocks of data much easier.

Pressing CDs from a master creates pits in the plastic substrate that are coated with aluminum and then sealed with a protective plastic layer. The plastic layer not only prevents physical damage, but also prevents oxidation of the aluminum that would reduce its reflectivity. In a CD-R (recordable) disk, the writing is done by a laser whose light is absorbed in a dye layer. The dye decomposes and is deposited on the metal layer, darkens it, and produces the same pattern of reflective and dark regions that the physical lands and pits do in the pressed CD.

CD-R disks are widely used as "write-once" storage devices for archiving files. CD-R drives are inexpensive and can be connected to a computer just like any other disk drive (for instance, using a standard SCSI or USB interface); they are standard equipment on many models. Files are written to a blank disk (which cost less than 50 cents at this time and continue to become less expensive). The writing process is not very fast; with the 4X drive in the laptop on which I am writing this, it takes 15 minutes to write a full disk. But that disk provides storage of more than 600 megabytes, enough to hold this entire book. After it has been written, the CD-R disk can be read in exactly the same drive as a pressed CD-ROM, or used as a master to duplicate the more durable pressed disks in quantity. CD-R disks have become so reliable and inexpensive that many people, myself included, use them instead of floppy disks to transmit even small amounts of data, when a physical disk is more suitable than transmission over the internet. The entire text and all of the nearly 2000 illustrations in this book were written onto a CD-R for backup and for transmission to the publisher.

One of the limitations of the CD is that in most cases, this writing operation must be performed in a single session. All of the files to be placed on the disk must be available on the host computer and written at the same time, so that the disk directory can be created for subsequent reading. In creating the Photo-CD format, Kodak introduced the idea of multiple directories (or multiple sessions) so that a disk could be taken in to the photo finisher to add more images (e.g., another roll of film) to an existing disk; however, not all drives or software that can read CDs are able to access the additional directories, and there is considerable overhead (about 9 MB) associated with the directories for each additional session.

At present, the most practical use of CDs is to produce entire disks of images or other data for storage or distribution. The images are archival in nature, since users are not able to modify them on the original disk. Reading from CD drives is generally rather slow. Even the newer drives that operate at 12 to 20 times or more the speed of the platter from that used for audio disks have reading speeds an order of magnitude less than a standard magnetic hard disk used in the computer. However, for accessing archival images that speed may be quite acceptable.

Higher-density formats such as those used in digital video disks (DVD) offer an order of magnitude more storage on the same size disk, with the same advantages and drawbacks as CDs, but as yet user-writable drives or media are less common. Re-writable CDs (CD RW) that use disks of the same size that can be erased and rewritten are slightly more expensive, but share the same problems of slow speed and lose the advantage of being tamper-proof. Writable but not alterable disks are especially useful in situations (medical and forensic imaging are the most oft-cited examples)

in which images must be stored permanently. Such drives are also used for financial records and other applications in which assurances are needed that data cannot be tampered with, either intentionally or accidentally.

Magnetic recording

The most common type of hard disks in computers use magnetic storage. The disk platter (of glass or metal) is coated with a thin layer of magnetic material much like that used in audio tapes. Reading and writing are done using a magnetic head that "flies" just above the disk surface, riding on a cushion of air pressure produced by the spinning disk surface. This method has exceptions: some drives use a flexible polymer substrate more like simple floppy disks, with the magnetic head in physical contact with the disk coating. Such differences affect the long-term durability of the media, the reading and writing speed, the physical density of storage and the amount of information that can be written onto a single disk. But from the user's point of view the details of the design technology are secondary considerations.

Removable magnetic storage disks with capacities of several hundred megabytes are commonly available (e.g., the popular "Zip" disk). Removable types of storage are desirable, because any conventional hard disk, even one with tens or hundreds of gigabytes of storage, will sooner or later fill up with images (usually sooner). Any system that is intended for serious image analysis work is going to require some form of removable and hence unlimited storage.

The speed of access of a conventional hard disk is somewhat faster than the removable types, and quite a bit faster than slow methods such as CDs. This speed makes it attractive to copy files onto a hard disk for access, and to write them to the removable media only when work is at least temporarily complete. With that approach, the hard disk serves as a short-term storage location for work in progress, but the removable media are still used to store images for long-term purposes. The hard disk is also the place where the computer operating system, and all of the various programs including those used for image analysis, are stored.

No consideration needs to be given to floppy disks because of their inability to hold large amounts of data. The typical diskette size of 1.4 MB can usually only hold one or a few images, and sometimes not even that. A single color image obtained from a color scanner with 600 dpi resolution (a widely available level of performance), taken from an original 8 × 10-inch photograph, would require about 81 MB of storage. Such files are not conveniently stored on floppy disks. Even a typical video image, requiring about 900 KB, takes up most of one floppy disk. That may be acceptable for sending one image to a colleague but is hardly useful for storage of many images, even ignoring the rather slow reading and writing speeds of these drives and the comparative fragility of the disks.

Tape drives are even slower. A single 4-mm tape, the same type of cassette used for digital audio recording, can hold several gigabytes of files, with a drive costing a few hundred dollars. But because access to any particular location on the tape requires rewinding and searching, these devices are really only useful for backing up a random access hard disk. Saving a collection of images on a tape for archival storage might make sense, however, as it is less expensive than any other form of digital storage except CD-R. Generally, tapes are less robust than disks because they are subject to stretching even in normal use and can easily be damaged by mishandling.

Likewise, magnetic storage is generally less "archival" than optical storage because stray fields or high temperatures can cause accidental erasure. The CD has the best claim to be archival, but this claim applies to the pressed disks; the CD-R recordable disks claim only a 5–10 year life with good storage conditions. Instead of using magnetic tapes and drives simply to back up a large disk drive, it is possible with existing software to treat the tape as though it is a large (and rather slow) disk,

that is, to access files randomly. This requires loading a directory into memory identifying where each file is on the tape, which takes memory space. Also, the reading times are not suitable if the files must be accessed very often. But such a medium might be used as a large and inexpensive data storage medium over a network.

One of the most recent developments in storage is network servers with arrays of redundant hard disks (RAID storage) that work in parallel for extremely high speed with error protection. The transmission of large images over ethernet still takes more time than local storage, but the advantages of shifting the responsibilities for maintaining and backing up storage to the information technology (IT) experts in your organization and the availability of images to all computers on the network makes this an attractive option in many situations.

If your purpose is archival storage of images, then regardless of what medium you select, you should prepare to invest in extra drives and interfaces. It is not likely that any of the current media will be readable by common storage devices in 10 years, so you will need to keep your own spare parts and working systems. And even then it may be difficult to transfer the images to the next generation of computers through different interfaces and networks. If you doubt this, consider the storage media in common use 10 years ago (8-inch floppy disks, mag-optical disks, Bernoulli, hard disk cartridges, etc.), none of which can be read with any ease today. Storage such as punched cards and DECtape, from 20 years ago, are unreadable except in a few very unusual circumstances because the machines are simply gone or in museums.

Another cost of archival storage is maintaining backup copies at some remote site, with appropriate security and fire protection, and having a routine method of storing regular backups there. It is really not enough simply to back up your hard disk to a tape or CD every Friday and drop the backup off in your safety deposit box on the way home, if your data are really worth keeping for many years. This is certainly the case for remote sensing tapes (a good example, by the way, of the changing face of storage — most of the early tapes can be read only on a handful of carefully preserved drives), medical records, and so on. For many kinds of records, and particularly images, photographic film is still the method of choice; it offers decades of storage, takes very little space, and can be accessed with standard equipment.

But for most people, the purpose of storing large numbers of images (certainly hundreds, perhaps thousands, probably not tens of thousands) is not so much for archival preservation as for access. The ability to find a previous image, to compare it to a current one, and to find and measure similarities and differences, is an obvious need in many imaging tasks. Choosing a storage technology is not the really difficult part of filling this need. A selection between any of the technologies mentioned above can be based on the tradeoffs among cost, frequency and speed of access, and the size of the storage required. The technical challenge lies in finding a particular image after it has been stored.

Databases for images

Saving a large number of images raises the question of how to locate and retrieve any particular image. If the storage uses removable media, the problem is compounded. These problems exist for any large database, for instance one containing personnel or sales records. But some unique problems are involved in searching for images. Database management programs are being introduced with some features intended specifically for images, but much remains to be done in this area.

Most database management routines offer the ability to search for entries based on some logical combination of criteria based on keywords or the contents of search fields. For images, it might be useful to search, for example, for images recorded between October 1 and 15, from a camera attached to a particular microscope, using transmitted light through a 500 nm color filter, obtained

from slide #12345 corresponding to patient ABCDE, etc. It might also be nice to find the images that contain a particular type of feature, for example satellite images of lakes in a certain range of sizes, whose infrared signature indicates that they have a heavy crop of algae.

The first of these tasks can be handled by many existing database searching routines, provided that the classification data have been entered into the appropriate fields for each image. The second task calls for much more intensive computer processing to extract the desired information from each image. It requires close coupling of the image analysis software with the database management routine. Such a system can be implemented for a specific application, but general and flexible solutions are not yet widely available in spite of considerable effort.

Searching through multiple fields or lists of keywords is typically specified by using Boolean logic, for instance that the creation date must lie before a certain value and that either one or another particular keyword must be present (and that multiple keywords must occur close together in the text), but that the image must not be in color. These types of searches are similar to those used in other kinds of database managers and internet search engines. A potentially more useful type of search would use fuzzy logic. For instance looking for features that were "round" and "yellow" does not specify just what criterion is used to judge roundness, nor what range of numeric values are required, nor what combination of color components or range of hue values is sufficiently yellow.

Sometimes described as "query by image content" (QBIC), this approach seeks to include visually important criteria such as color, texture, and shape of image objects and regions (Pentland et al., 1994; Yoshitaka and Ichikawa, 1999; Rui et al., 1999). Depending on the field of use, quite different criteria may be important for searching. For instance in a medical application the user might want to find "other images that contain a tumor with a texture like this one" (which implies the concept of the object named *tumor*, as well as the description of the texture), while in surveillance the target might be "other images that contain objects of similar shape to this airplane." Key issues for such a search include derivation and computation of the attributes of images and objects that provide useful query functionality, and retrieval methods based on similarity as opposed to exact match. Queries may be initiated with one or a few example images ("query by example"), perhaps with subsequent refinement. Effective human interfaces for such query systems are also in development.

Implementing a search for the example of "round" and "yellow" features would require first measuring a representative sample of features in the images in the database to obtain measures for roundness (perhaps the ratio of shortest to longest dimension, but other shape criteria are discussed in Chapter 9) and hue. Some systems also use spatial frequency data to characterize texture. Histograms of frequency versus value for these parameters would then allow conversion of the adjectives to numbers. For instance, "round" might be taken to mean objects falling within the uppermost 5–10% of the range of actual values, and "yellow" the range of hues values bracketing true yellow and enclosing a similar fraction of the observations.

A program that might potentially be shown an example image and then told to search the internet for similar ones is still very far away. Part of the problem is that it is difficult to know beforehand what aspects of an image may later prove to be important. When we search through pictures in a shoebox, they vary widely. This usually allows them to be visually distinguished at a glance, but how will a search program know who is Aunt Sarah? And the meaning of the image content is also obscure except to a human with much additional background knowledge.

For example, **Figure 23** shows several images found on the internet (each from a different web site) in a fraction of one second using a search engine with the target "ACC + basketball + tournament + photo." Notice that several of the pictures show several players and the ball, but the uniform colors vary (and one image is not in color). Some do not show the ball, while one shows only the ball and net. Three show neither players nor the ball, but other things that are recognizably



Figure 23. *Basketball images as discussed in the text.*

connected to basketball (a referee, a tournament bracket chart, and a basketball court). At the present and foreseeable state of the art, showing any one picture to a query-by-example program would not find the others.

An even more difficult scenario is present in many scientific imaging situations. There may be a very large number of quite similar images, such as liver tissue sections with the same colored stain from many patients, or satellite images of populated areas with many buildings, in which visual and machine methods of locating a particular image are hardly useful at all. A very experienced operator may be able to recognize some key features of interest, but not to describe them to the software in a meaningful or exclusive way. In such cases a search based on keywords, dates, patient records, location, etc. may be more appropriate. On the other hand, query by content systems have been used successfully for locating paintings and other artwork (Holt and Hartwick, 1994).

Several commercial programs claim to offer some degree of search capabilities for images. By far the best known and most fully documented is the IBM QBIC system, a prototype development kit from IBM Almaden Research with more than 10 years of experience and development (Niblack et al., 1993; Faloutsos et al., 1994; Flickner et al., 1995). It allows a user to query a collection of images based on colors, textures, shapes, locations and layout. For example, "find an image with a green background and a round red object in the upper left." This and other similar systems first process the images in the files to extract information such as the histogram (for mean and standard deviation of brightness and hue), and to segment the image into homogenous regions for measurement of size and shape, and then search through these signatures for specific numeric values (De Marsicoi et al., 1997).

An even more powerful approach to resolving fuzzy criteria into numeric values ranks all of the observations according to dimension, hue or some other objective criterion and then uses the rank numbers to decide how well a particular object conforms to the adjective used. If a feature was ranked 10th in roundness and 27th in yellowness out of a few hundred features, it would be considered round and yellow. On the other hand, if it was ranked 79th in one or the other attribute, it would not be. Both of these methods ultimately use numeric values, but judge the results in terms of the actual range of observations encountered in the particular type of application. And they do not use traditional "parametric" statistics that try to characterize a population by mean and standard deviation, or other narrow descriptors.

Fuzzy logic offers some powerful tools for classification without requiring the user to decide on numerical limits; however, although fuzzy logic has been enthusiastically applied to control circuits and consumer products, its use in database searching is still rather limited. That, combined with the time needed to measure large numbers of representative images to give meaning to the users' definition within a particular context, seems to have delayed the use of this approach in image database management. Consequently, most searching is done using specific values (e.g., a creation date after December 1, 2001, instead of "recent") and Boolean matching with user-specified criteria.

Typically, it is up to the user to establish the criteria that may be interesting for future searches before the database is established and the images are stored away. If the proper fields have not been set up, there is no place to save the values. It is impossible to overstate the importance of designing the proper search fields and lists of keywords ahead of time, and the difficulty of adding more fields retrospectively. No simple guidelines are used for setting up fields, because each imaging application has unique requirements. Establishing dozens or even hundreds of search fields with large numbers of keywords is a minimum requirement for any image database management routine.

Of course, it is still necessary to fill the search fields with appropriate values and keywords when the image is saved. Some of the fields may be filled automatically, with things like date and time, operator name, perhaps some instrument parameters such as magnification or wavelength, location (this could either be latitude and longitude, or coordinates on a microscope slide), and so forth. Even patient or sample ID numbers can in principle be logged in automatically; some laboratories are using bar code labeling and readers to automate this function. Doing this at the time the image is acquired and saved is not too burdensome, but obviously supplying such information retrospectively for a set of images is difficult and error-prone.

But the really interesting keywords and descriptors require the human observer to fill in the fields. Recognition and identification of the contents of the image, selection of important characteristics or features while ignoring others, and then choosing the most useful keywords or values to enter in the database is highly dependent on the level of operator skill and familiarity, both with the images and with the program.

Not surprisingly, the entries can vary greatly. Different operators, or the same operator on different days, may produce different classifications. One method that is often helpful in constraining operators' choice of keywords is setting up a glossary of words that the operator can select from a menu, as opposed to free-format entry of descriptive text. The same glossary is later used in selecting logical combinations of search terms.

Values for measurement parameters that are included in the database can usually be obtained from an image analysis program. In later chapters on image measurement, we will see that automatic counting of features, measurement of their sizes, location, densities, and so forth, can be performed by computer software. But in general this requires a human to determine what it is in the image that should be measured. These programs may or may not be able to pass values directly

to the database; in many practical situations manual retyping of the values into the database is required, offering further opportunities for omissions and errors to creep in.

Even in cases in which the target is pretty well known, such as examining blood smear slides, or scanning the surface of metal for cracks containing a fluorescent dye, there is enough variation in sample preparation, illumination conditions, etc., to require a modest amount of human oversight to prevent errors. And often in those cases the images themselves do not need to be saved, only the numeric measurement results. Saving entire images in a database is most common when the images are *not* all nearly the same, but vary enough to be difficult to describe by a few numbers or keywords.

A very different type of database uses the image as the organizing principle, rather than words or numbers. Numerical or text data in the database is keyed to locations on the images. This approach, mentioned previously, is generally called a geographical information system (GIS) and is used particularly when the images are basically maps. Imagine a situation in which a number of aerial and satellite photographs of the same region have been acquired. Some of the images may show visible light information, but perhaps with different resolutions or at different times of day or seasons of the year. Other images may show the region in infrared bands or consist of range images showing elevation. There may also be maps showing roads and buildings, land use patterns, and the locations of particular kinds of objects (fire hydrants, electrical transformers). Tied to all of this may be other kinds of data including mail addresses, telephone numbers, and the names of people, for example. Other kinds of information may include temperatures, traffic patterns, crop yields, mineral resources, and so forth.

Organizing this information by location is not a simple task. For one thing, the various images and maps must somehow be brought into registration so that they can be superimposed. The resolution and scales are typically different, and the "depth" of the information is quite variable as well (and certainly consists of much more than a grey scale value for each pixel). Finding ways to access and present this data so that comparisons between quite disparate kinds of information can be made presents real challenges. Even searching the database is less straightforward that might be imagined. For instance, selecting coordinates (e.g., latitude and longitude) and then asking for various kinds of information at that location is comparatively straightforward. Computerized maps (distributed on CD) are presently available that can be accessed by a laptop computer connected to a GPS (Global Positioning System) receiver that picks up timing signals from the network of satellites to figure out location anywhere on the earth with an accuracy of a few meters. That can obviously be used to bring up the right portion of the map. But if you are driving a car using such a system to navigate, your question may be "show me the routes to follow to the nearest gas station, or hospital, or to avoid the 5:00 p.m. traffic jam and reach the main highway." Clearly, these questions require access to additional information on streets, traffic lights and timing, and many other things. And that same database can also be accessed by entering a telephone number or zip code. This is current technology; the promise is for much richer databases and more flexible ways to access and display the information.

Browsing and thumbnails

This section started out with the idea that saving images so that they can later be compared to other images is necessary precisely because there is no compact way to describe all of the contents of the image, nor to predict just what characteristics or features of the image may be important later. That suggests that using brief descriptors in a database may not provide a tool to locate particular images later. Consequently, most image databases also provide a way for the user to see the image, or at least a low-resolution "thumbnail" representation of it, in addition to whatever logical search criteria are employed.

Some approaches to image databases write the full image and one or more reduced resolution copies of the image onto the disk so that they can be quickly loaded when needed. This may include, in addition to the full original copy of the image, a very small version to use as a display thumbnail for browsing, a version with a resolution and color gamut appropriate for the system printer, and perhaps others. Since most of the auxiliary versions of the image are much smaller in storage requirements than the original, keeping the lower resolution copies does not significantly increase the total space required, and it can greatly speed up the process of accessing the images. Kodak's Photo-CD is an example of a storage format that maintains multiple resolution versions of each image. In some cases it is practical to use a lossy compression technique to save the lower resolution versions of the images, to further reduce their storage requirements or to facilitate internet transmission.

The reason for thumbnails, or some kind of reduced size representation of the images in the database, is that allowing the user to "browse" through the images by showing many of them on the screen at the same time is often an essential part of the strategy of looking for a particular image. The difficulties with keywords and multiple data fields noted previously mean that they can rarely be used in a Boolean search to find one particular or unique image. At best, they may isolate a small percentage of the stored images, which can then be presented to the user to make a visual selection. Even highly successful and widely reported search systems, such as the Automated Fingerprint Identification System (AFIS), rely on skilled human operators to review the 10–20 best matches found by the system based on measured values (the minutiae in each fingerprint) to perform conclusive identification.

Browsing through images is very different than the search strategies that are used for most other kinds of data. It is common on most computer systems, either as part of the system software itself, or as basic utility routines, or as part of applications, to incorporate intelligent search capabilities that can locate files based not only on the values of data fields (e.g., a creation date for the file) or keywords, but also by examining the actual contents of the files. For instance, the word processor being used to write this chapter can search through all of the files on my hard disk for any document containing the phrases "image" and "database" in close proximity to each other, and then show me those files by name and with the phrases displayed in context.

But that approach is possible because the files are stored as text. There may be special formatting characters present (and indeed, it is possible to search for those as well: "Find documents containing the word 'large' in Times New Roman italics") but the bulk of the file content is a straightforward ASCII representation of the letters that make up the words and phrases. Searching strategies for text, including ignoring capitalization or requiring an exact match, and allowing for "wild card" characters (e.g., "find documents containing the phrase 'fluorescen# dye'" would find both fluorescent and fluorescence), are widely used. Much effort has gone into the development of efficient search algorithms for locating matches.

A relatively new development in text searches uses natural language rules to interpret the text. This search technique distinguishes among nouns, verbs, and adjectives to extract some meaning from the text. That, combined with the use of a thesaurus and dictionary so that words can be substituted, allows specifying target text by entering a few topical sentences, and then having the search engine look for and rank matches.

How can this technique be used for images? Certainly there is no way to simply match a specific series of bytes. The image data are typically stored on disk as a sequence of pixel values. For many monochrome images, storage requires one byte per pixel, and the values from 0 to 255 represent the grey scale of the data. For images that have greater dynamic range, two bytes per pixel may be needed; some computers store the high byte and then the low byte, and some the reverse. For color, at least three bytes per pixel are needed. These may be stored with all three

values in some fixed order for each pixel, or the entire row (or even the entire image) may be stored separately for the red, green and blue values (or other color space representation).

Dozens of different storage formats are available for images. Few image database management routines support more than a handful of formats, expecting that most users will select a format according to specific needs, or as used by a few particular programs. When images must be imported from some "foreign" format, it is usually possible to translate them using a dedicated program. This is particularly needed when different dedicated computers and programs acquire images from various instruments, and then transmit them to a single database for later analysis.

A few relatively "standard" formats, such as TIFF ("tagged image file format") files, are used on several different computer platforms, while others may be unique to a particular type of computer (e.g., PICT on the Macintosh, BMP on the PC) or even proprietary to a particular program (e.g., the PSD format used by Adobe Photoshop®). In the latter case, the widespread use of the program can make the format a kind of standard in its own right. Some of these "standards" (TIFF is an excellent example) have so many different options that many programs do not implement all of them. The result is that a TIFF file written by one program may not be correctly read by another program that uses a different subset of the 100 or so options.

Some storage formats include various kinds of header information that describes formats, color tables and other important data more or less equivalent to the formatting of a text document. Some compress the original data in various ways, and some of the methods do so in a "lossless" manner that allows the image to be reconstructed exactly, while others accept some losses and approximate some pixel values in order to reduce the storage requirement.

For instance, the Macintosh PICT format is a lossless method that represents a line of pixels with the same value by listing the value once, and then the number of times it is repeated. For computer graphics, animation, and rendered drawings from drafting programs, this "run-length encoding" (RLE) method is very efficient. It is also used to transmit faxes over telephone lines. But it does not offer much compression for typical real-world images because groups of pixels are not usually uniform.

If compression of the data is present, the computer must read the entire image file and reconstruct the image in memory before anything can be done with it. But even if the data can be scanned directly within the disk file, how can it be searched to locate a particular image based on the contents? There is not generally a specific sequence of pixel values along one line of the image (most images are stored in a raster format, as a series of lines) that is the target. There is not even usually a specific two-dimensional pattern of pixel values. Features in images are more irregular than that, and may occur in unexpected locations, sizes and orientations.

Statistical averages of the image, as may be summarized in its brightness histogram, most predominant color, etc., may sometimes be useful, but they are rarely computed while searching for a particular image in the database. Instead, if such parameters are considered important, they are determined beforehand, when the image is stored, and written into specific fields so that they can be searched using standard logical tests.

There is one approach to data matching that is sometimes applied to this kind of searching. Cross-correlation of a target image with each image in a database is a way to look for images that are similar to the target, in a very particular sense. The use of cross-correlation is discussed and illustrated in Chapter 5, because it is usually implemented using Fourier transforms. These can be speeded up by using dedicated hardware, but even so it can be quite time consuming to search through a large database for a best match.

One application of this approach is matching surveillance photos to identify targets of possible military interest. For example, cross-correlation of an aerial image of an airport against a database of

images of airplanes, each type viewed in many different orientations, will match the type of airplanes and their locations in the image. For more diverse images, or ones containing features that are more variable in their shape, contrast or color, this method is less suitable.

When images are stored in a database that extends over several physical disks, particularly when removable media are used, many of the images may not be accessible or online when the search is made. The usual solution to this problem is to keep the descriptive keywords and other search fields, plus at least a thumbnail representation of the image, in a file with the main program. This allows rapid searching to find a subset of the images that may be examined visually by the user. The program has the location of each image stored in its file, so it can then request the user to insert the particular disks to load the actual images. Even with this approach, the search file can be quite large when hundreds or thousands of images are included in the database. The search file can itself require a large (and fast) disk for storage.

On the other hand, for several reasons, the search fields, keywords, thumbnails, and other ancillary information should be stored with the image instead of in a central data file. Such storage makes the file containing the image self-contained, so that it can be copied with all its information intact. It is also possible to maintain a record of who has accessed the image and when, or to keep a detailed record of whatever changes have been made to an image. Such information may be very important in reconstructing the history of processing an image.

Image database programs may also be required to limit access to images, for instance with passwords. A networked file server may allow one group of users to read images from the files, another group to add images to the database, and a third group to process or modify images. Since images are large, moving them across local area networks from a central location to many workstations is a far from trivial consideration. This is particularly a concern in medical imaging and remote sensing applications, where a large number of images are to be accessed by a moderate number of users.

Finally, the question for any image searching routine is just how it is to be used. Finding one or several images according to some criteria is usually not the end result but the beginning. How can the image(s) now be loaded into whatever program is to be used for processing or measurement? Some database management programs can act as a filter that is used by any program opening a file. This is convenient for loading images but may not lend itself to adding images to the database. Other management programs can simply locate the images and copy them onto a local disk (and perhaps convert their format or decompress them), so that the user may more easily open them into the desired application.

Lossless coding

Image compression was mentioned previously. This is desired for storage or transmission in order to reduce the rather large size of most image files, and is quite an active area of study. We will not review all of the techniques for compression described in the literature, since most of them are not implemented in standard software packages available for dedicated image processing. Most of the methods fall into just a few categories, and representative examples of each are discussed in the following paragraphs.

Image compression methods can be judged by two criteria. One is the time needed to accomplish the compression and decompression and the degree of compression achieved. This is particularly important when images are being compressed for "real-time" transmission, as in video conferencing, or perhaps when transmitting large images via the internet. The second criterion is the degree of preservation of the image. It is this latter area of concern that will be primarily discussed here.

The first and most important distinction between compression methods is whether they are lossless or lossy techniques. A lossless method is one that allows exact reconstruction of all of the individual pixel values, while a lossy method does not. Lossless methods, often referred to as image coding rather than compression, have been around for some time, with much of the original development being directed toward the transmission of images from the space probes. The communication bandwidth provided by these low-power transmitters did not allow sending many images from the remote cameras unless some method was used to reduce the number of bits per pixel.

A simple, early approach was to send just the difference between each pixel and the previous one (sometimes called "delta compression"). Because most areas of the image had little change, this reduced the average magnitude of the numbers, so that instead of requiring (for instance) 8 bits per pixel, fewer bits were needed. This is another way of saying that images are highly correlated. A histogram of the differences between adjacent pixels has a peak near zero and few large values, as shown in **Figures 24** and **25** for images that have different histograms of pixel brightness values.

Further approaches to compression used algorithms that examined several preceding pixels, predicted the next value based on some kind of fitting algorithm, and then just stored the difference from that. Advances on those methods use pixels on preceding lines as well, for further improvement in the predicted value and hence reduction in the difference values (Daut et al., 1993). Obviously, a method that looks at preceding lines is suitable for so-called progressive scan imaging rather than an interlaced scan as used in broadcast television. Most of these compression algorithms were originally designed and utilized with live image sources. If the image has already been stored in memory, then the full array of pixels is available.

An additional refinement codes the differences between a pixel and either its predicted value or its predecessor more efficiently. For instance, in recording each wavelength band in Landsat images a 4-bit number is used for most differences, with two values of the 16 possible numbers reserved

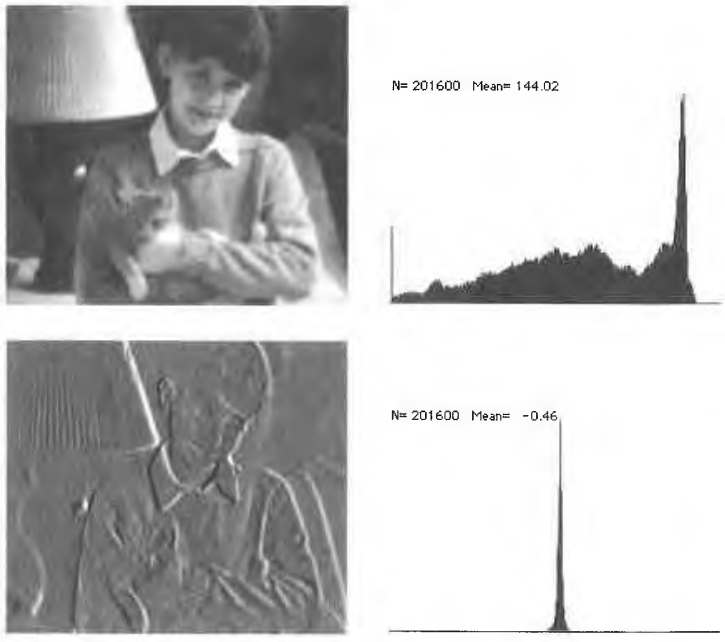


Figure 24. *Example image ("Girl") with its histogram, and the results of compression by calculating differences between each pixel and its left-hand neighbor. The original image has a broad range of pixel brightness values. The histograms of the original and compressed image show that the latter has most values near zero. Contrast of the displayed compression image has been expanded to show pixel differences.*

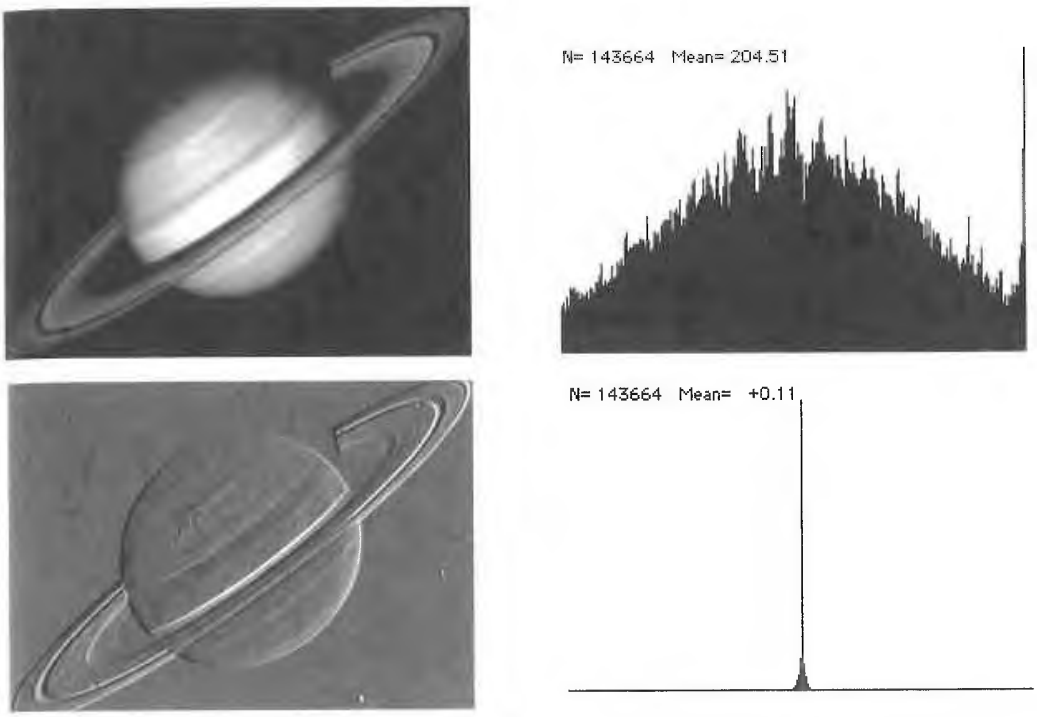


Figure 25. *Example image ("Saturn") with its histogram, and the results of compression by calculating differences between each pixel and its left-hand neighbor. The original image contains many black pixels. The histograms of the original and compressed image show that the latter has most values near zero. Contrast of the displayed compression image has been expanded to show pixel differences.*

as flags indicating that a (rare) larger difference value follows, either positive or negative. An average of about 4.3 bits per pixel is needed to store the images, instead of the full 8 bits.

If further optimization is made to allow variable length codes to be used to represent differences, storage requirements can be further reduced to about 3.5 bits per pixel for the Landsat data. One of the most widely used variable-length coding schemes is Huffman coding. This uses the frequency with which different grey values occur in the image in order to assign a code to each value. Shorter codes are used for the more frequently occurring values, and vice versa. This can be done either for the original grey values, or for the pixel difference values. Huffman coding is also used for other types of data, including text files. Because some letters are used more than others in English, it is possible to assign shorter codes to some letters (in English, the most frequently used letter is E, followed by TAOINSHRDLU). Morse code for letters represents an example of such a variable length coding for letters (not an optimal one).

As a simple example of the use of Huffman codes for images, consider an image in which the pixels (or the difference values) can have one of 8 brightness values. This would require 3 bits per pixel ($2^3 = 8$) for conventional representation. From a histogram of the image, the frequency of occurrence of each value can be determined and as an example might show the following results (**Table 1**), in which the various brightness values have been ranked in order of frequency. Huffman coding provides a straightforward way to assign codes from this frequency table, and the code values for this example are shown. Note that each code is unique and no sequence of codes can be mistaken for any other value, which is a characteristic of this type of coding.

Table 1. Example of Huffman codes assigned to brightness values

Brightness Value	Frequency	Huffman code
4	0.45	1
5	0.21	01
3	0.12	0011
6	0.09	0010
2	0.06	0001
7	0.04	00001
1	0.02	000000
0	0.01	000001

Notice that the most commonly found pixel brightness value requires only a single bit, but some of the less common values require 5 or 6 bits, more than the three that a simple representation would need. Multiplying the frequency of occurrence of each value times the length of the code gives an overall average of:

$$(0.45 \cdot 1) + (0.21 \cdot 2) + (0.12 \cdot 4) + (0.09 \cdot 4) + (0.06 \cdot 4) + (0.04 \cdot 5) + (0.02 \cdot 6) + (0.01 \cdot 6) = 2.33 \text{ bits/pixel}$$

Using software to perform coding and decoding takes some time, particularly for the more "exotic" methods, but this is more than made up in the decreased transmission time or storage requirements. This is true of all of the coding and compression methods discussed here, and is the justification for their use. Information theory sets a lower limit to the size to which an image (or any other file of data) can be reduced, based on the distribution of actual values present. If, for example, an image consists of 256 possible grey levels with actual frequencies of occurrence (taken from a brightness histogram of the image) of $p_0, p_1, \ldots, p_{255}$, then the entropy of the image is

$$H = -\sum_{i=0}^{255} p_i \cdot \log_2 p_i \tag{1}$$

Information theory establishes this as a theoretical limit to the number of bits per pixel needed to represent the image, and provides a performance criterion for actual coding methods. If this calculation is applied to the same example with eight grey levels as used above to illustrate Huffman coding, it calculates H = 2.28 bits per pixel as a minimum. Huffman coding is not optimal except in the unique case in which the frequencies of occurrence of the various values to be represented are exactly integral powers of 1/2 (1/4, 1/8, . . .). But this example indicates that it does offer a useful degree of compression with modest computational needs. Other coding techniques are available that can approach the theoretical limit, but simple methods like Huffman coding are often good enough to be widely used.

Table 2. Entropy values (bits per pixel) for representative grey-scale images shown in Figures 24, 25, and 26

Image	H (original)	H (difference-coded)
Girl (**Figure 24**)	7.538	3.059
Saturn (**Figure 25**)	4.114	2.019
Bone marrow (**Figure 26a**)	7.780	4.690
Dendrites (**Figure 26e**)	7.415	4.262
Bug (**Figure 26i**)	6.929	3.151
Chromosomes (**Figure 26m**)	5.836	2.968

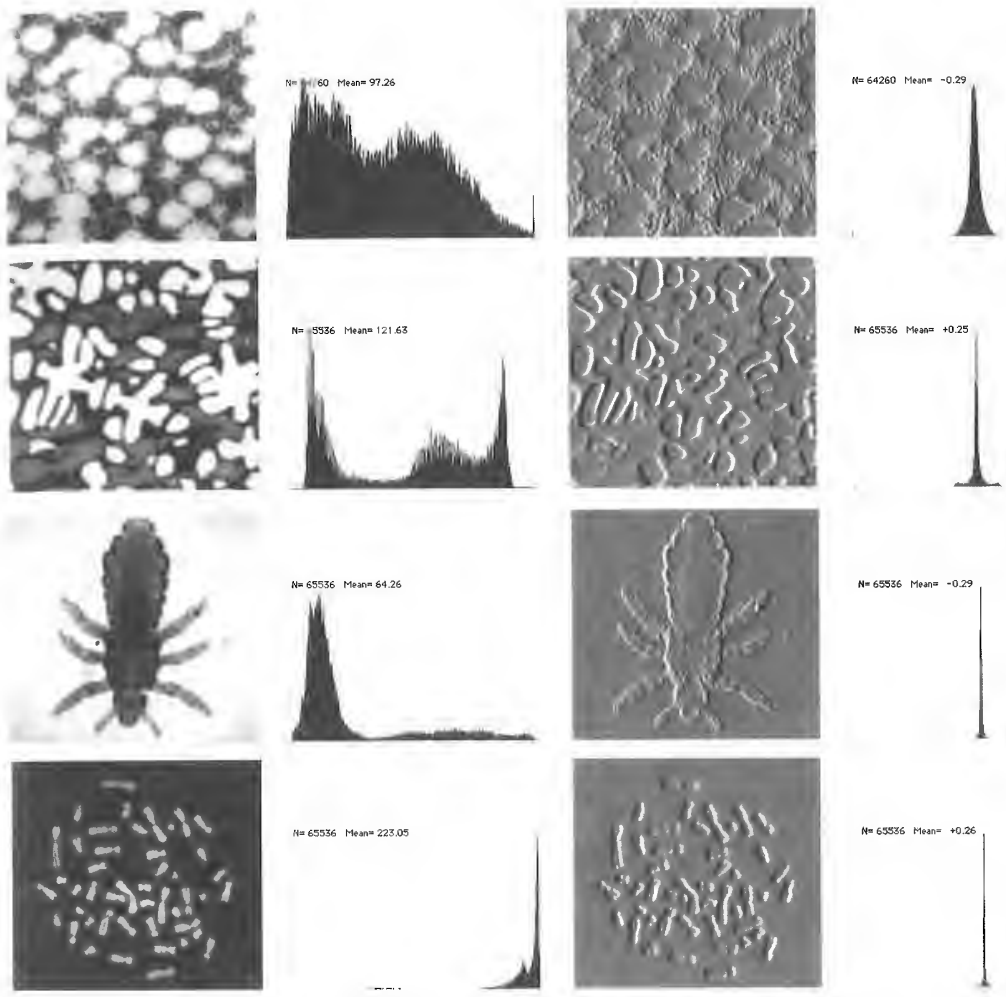


Figure 26. *Application of difference compression (autocorrelation) to various test images, with histograms of pixel values. The entropy values are listed in* ***Table 2****.*

Table 2 lists the entropy values for a few typical images, which are shown in **Figure 26** along with their original brightness histograms and the histograms of difference values between horizontally adjacent pixels. Some of these images have large peaks in their original histograms that indicate there are many pixels with similar brightness values. This affects both the entropy value and as discussed above the generation of an optimal Huffman code. Notice the reduction in information density (H) as the visual appearance of images becomes simpler and larger areas of uniform brightness are present. Calculating the H values for the difference coding of these images reveals a considerable reduction as shown in the table, averaging about a factor of 2 in additional compression.

The entropy definition can be expanded to cover neighbor pairs by summing the joint probabilities of all pairs of two (or more) grey values for neighbors in the same way. This would apply if a coding scheme was used for pairs or larger combinations of pixels, which in theory could compress the image even further.

Although it is unusual to apply coding methods to all possible pairs of grey values present, a related coding scheme can be used with images just as it is with text files. This approach scans through the file once looking for any repeating patterns of byte values. In a text file, the letter patterns present in English (or any other human language) are far from random. Some sequences of letters such as "ing," "tion," "the," and so forth occur quite often, and can be replaced by a single character. In particular cases, common words or even entire phrases may occur often enough to allow such representation.

A dictionary of such terms can be constructed for each document, with segments selected by size and frequency to obtain maximum compression. Both the dictionary itself and the coded document must be stored or transmitted, but for a large document the combination requires much less space than the original. Compression of typical text documents to half of their original size is commonly achieved in this way. For small documents, using a standard dictionary based on other samples of the language may be more economical than constructing a unique dictionary for each file. Dictionary-based methods, known as a "Lempel-Ziv" technique (or one of its variants, such as Lempel-Ziv-Welch, or LZW), is commonly used to compress text to reduce file sizes.

Applying the same compression algorithm to images, treating the sequence of bytes just as though they represented ASCII characters is not so successful (Storer, 1992). Few exactly repeating patterns exist in most images; even small amounts of random noise cause changes in the pixel values. These noise fluctuations may not be important, and in fact it is easy to argue that much of this "noise" arises in the camera, electronics and digitization and is not part of the scene itself. But our goal in coding the image is to preserve everything so that it can be reconstructed exactly, and there is no *a priori* reason to discard small fluctuations. Deciding that some variations in pixel values represent noise while others represent information is very difficult. Because of these fluctuations, whatever their source, few repeating patterns are found, and the compressed image is usually not significantly smaller than the original.

A close cousin of these coding methods is run-length encoding. This was mentioned before as being used in some of the standard file formats, such as the Macintosh PICT format and the compressed TIFF format used on most computer platforms. Run-length encoding (RLE) looks for any row of pixel brightness values that repeats exactly, and replaces it with the value and the number of pixels. For natural grey scale images, such rows do not occur very often and little compression is achieved; however, for computer graphics, drawings and animation images, this method can achieve very high compression ratios.

Run-length encoding is particularly appropriate for binary (black and white) images. Fax machines use RLE to send images of pages containing text or images via telephone lines. Thresholding is often used as discussed in a subsequent chapter to reduce images to black and white representations of features and background. Such binary images can be efficiently represented by run-length encoding, and in addition the encoded image is directly useful for performing some measurements on images.

Most of the common image formats used in desktop computers and workstations are lossless representations of images. Some simply record all of the pixel values in some regular order, perhaps with header information that gives the dimensions of the image in pixels, the colors represented by the various values, or other similar information. Some use modest coding schemes such as run-length encoding to speed up the reading and writing process and reduce file size. Some, such as TIFF or HDF, are actually a collection of possible formats that may or may nor include coding and compression, and have within the header a flag that specifies the details of the format in the particular file.

Fortunately, program users rarely need to know the details of the format in use. Either the images have been stored by the same program that will later read them back in, or several programs share

one of the more-or-less standard formats, or there is a translation facility from one format to another within the programs or provided by a separate program. Quite a few such translation programs have been developed specifically to cope with the problems of translating image files from one computer platform to another.

Reduced color palettes

As part of the storage format for images, many systems try to reduce the number of colors represented within each image. One reason that color images occupy so much storage space is that three color components, usually RGB, are stored for each pixel. In most cases, these occupy one byte each, giving 256 possible values for each component. In a few cases, reduction to 32 grey levels ($2^5 = 32$) is used to allow reducing the storage requirements to 2 bytes, but the visual artefacts in these images due to the magnitude of the changes between colors on smoothly varying surfaces may be distracting. In some other cases, more than 8 bits per color component are used, sometimes as many as the 12 bits ($2^{12} = 4096$) which are generally considered to be required to capture all of the dynamic range of color slide film, and this of course represents a further increase in storage requirements.

Some images can be reduced for storage, depending on their ultimate use, by constructing a unique color coding table. This allows each pixel to have a stored value from 0 to 255 and occupy one byte, while the stored value represents an entry in a table of 256 possible colors to be used to display that image. This method may seem to reduce the amount of information in the image drastically, but it corresponds nicely to the way that many computer screens actually work. Instead of a 24-bit display with 8 bits (256 intensity values) for R, G, and B, many low cost displays use an 8-bit display memory in which each of the possible 256 pixel values is used to select one of 16 million colors (2^{24}). The palette of colors corresponding to each pixel value is stored with the image (it occupies only $3 \times 256 = 768$ bytes), and written to the hardware of the display to control the signals that are output to the cathode ray tube (CRT) or other device. This is sometimes called a lookup table (LUT) since each pixel value is used to "look up" the corresponding display color. Manipulation of the same LUT to produce pseudocolor displays was described in Chapter 1.

If the LUT or palette of color values is properly selected, it can produce quite acceptable display quality. Since the selection of which 256 colors are to be used to show an entire image (typically containing from 1/4 million to 1 million pixels) is critical to the success of this method, various algorithms have been devised to meet the need. The color triplet can be considered (as discussed in Chapter 1) as a vector in a three-dimensional space, which may be RGB, HSI, etc. The process of selecting the best palette for the image consists of examining the points in this space that represent the colors of all the pixels in the image, and then finding the clusters of points in a consistent way to allow breaking the space up into boxes. Each box of points of similar color are then represented by a single color in the palette. This process is usually called vector quantization; there are several iterative algorithms that search for optimal results and are all rather computer intensive (Heckbert, 1982; Braudaway, 1987; Gentile et al., 1990).

Improvements in both visual quality and speed are obtained by using the YCC (broadcast video, also called YIQ and YUV) space with the Y (luminance, or brightness) axis given more precision than the two chrominance signals (yellow-blue and red-green), because human vision is more sensitive to changes in brightness than in color. This is the same argument used for reducing the bandwidth used to transmit the color information in television. The scaling of the values is also nonlinear (Balasubramanian et al., 1994), corresponding to the response of the display CRT to changes in signal intensity. In other words, the goal of the color palette compression is to reproduce the image so that visual observation of the image on a television screen will not show objectionable artefacts such as color banding. And for that goal these methods work well. But if any

further quantitative use of the image is intended, the loss of true color information may create obstacles. It may even produce difficulties for printing the images, since the gamut of colors available and the response of the color intensities is different than for CRT displays.

JPEG compression

Much higher compression ratios can often be achieved for images if some loss of the exact pixel values can be tolerated. There is a rich literature discussing the relative merits of different approaches. For instance, the encoding of differences between neighboring pixels can be made lossy and gain more compression simply by placing an upper limit on the values. Since most differences are small, one might represent the small differences exactly but only allow a maximum change of ±7 grey levels. This restriction would reduce the number of bits per pixel from 8 to 4, without any other coding tricks. Larger differences, if they occur, would be spread out over several pixels. Of course, this might distort important edges or boundaries.

Three approaches have become common enough to be implemented in many desktop computers and are fairly representative of the others. The popular JPEG (Joint Photographers Expert Group) standard is widely used in digital cameras and Web-based image delivery. The wavelet transform is newer but will become part of the JPEG standard soon, and claims to minimize some of the visually distracting artefacts that can appear in JPEG images. Fractal compression has also shown promise, and claims to be able to enlarge images by inserting "realistic" detail beyond the resolution limit of the original. Each method will be discussed and examples shown.

The JPEG technique is fairly representative of many of these transform-based compression methods. It uses a discrete cosine transform (DCT) that is quite similar to the Fourier transform method discussed in Chapter 5. The JPEG standard is a collaborative effort of the CCITT (International Telegraph and Telephone Consultative Committee) and the ISO (International Standards Organization), and actually comprises a variety of methods that are not explicitly intended for computer storage; the algorithm deals with a stream of bytes as might be encountered in image transmission.

The JPEG discrete cosine transform consists of several steps:

1. The image is separated into HSI channels and subdivided into 8 × 8 pixel blocks. If the image is not an exact multiple of 8 pixels in width or height, it is temporarily padded out to that size.
2. Each 8 × 8 pixel block is processed using the discrete cosine transform. This is closely related to the more familiar Fourier transform, except that all of the values are real instead of complex. The transform produces another 8 × 8 block of values for the frequency components. Although the original pixel values are 1 byte = 8 bits (0 ... 255), the transformed data are stored temporarily in 12 bits, giving 11 bits of precision plus a sign bit. Except for the possibility of round-off errors due to this finite representation, the DCT portion of the algorithm does not introduce any loss of data (i.e., the original image can be exactly reconstructed from the transform by an inverse DCT).
3. The 64 coefficients for each block are quantized to a lower precision by dividing by a fixed table of values that gives the least precision for high frequency terms. Adjusting the "quality" factor in most implementations increases the factors and reduces more terms to low precision or erases them altogether. This is the "lossy" step in the compression. In most cases, more precision is retained for the intensity or luminance than for the color data. This is because in the intended use of the compression method for human viewing of images, it is generally accepted that more fidelity is needed in image brightness than is needed in color, as mentioned before.
4. The first of the 64 coefficients for each block is the average brightness or "DC" term. It is represented as a difference from the same term for the preceding block in the image. The blocks are listed in raster-scan order through the image.

5. The remaining 63 coefficients for each block are scanned in a zig-zag diagonal order that starts with the lowest frequencies and progresses to the highest. The entire data stream is further compacted by using a Huffman coding as discussed earlier. This step is loss-free.

The decompression or image reconstruction procedure reverses these steps to produce an image that is similar to the original image. Compression and decompression for the DCT are symmetric (same computational complexity and time). Some other compression methods, such as fractal compression of images and Moving Pictures Experts Group (MPEG) compression of movies, are asymmetric and take much longer to achieve the compression than is needed for decompression during playback.

The loss of high-frequency terms results in some image defects and distortions. Since the loss of precision depends on the magnitude of the values, results are different in the various 8×8 pixel blocks in the original image, and the exact nature of the defects will vary from place to place. In general, sharp boundaries, edges, corners, and lines require the highest frequencies to accurately reproduce, and it is these that will show the greatest degradation. The results will depend on exactly where the line or corner lies with respect to the 8×8 block boundaries. An 8×8 block of pixels with a uniform grey value would be compressed to a single coefficient that would be accurately encoded, and all of the remaining coefficients would actually be zero so that no loss would occur. Small deviations from this uniform grey might or might not be preserved.

JPEG, or any similar other approach based on transforms, can be improved in several ways. One is to choose the best possible color space in which to represent the image before starting. For instance, Photo-CD (discussed in the following paragraphs) uses the CIE color space while JPEG uses HSI. Second, instead of dividing the image into non-overlapping tiles, a system using blocks that overlap in both the horizontal and vertical directions can suppress some of the artefacts that appear at block boundaries (Young and Kingsbury, 1993). Third, and perhaps most important, the quantization of terms can be made more flexible. Different scaling factors can be used for each color plane or for different colors, for different frequencies, and perhaps different directions, depending on the intended use of the image (for viewing, printing, etc.). These methods can improve the reconstructed image quality at a given level of compression, with no change in reconstruction time.

The use of JPEG compression or indeed any "lossy" compression technique for images should be restricted to images intended for visual examination and printing, and should not be used for images intended for measurement and analysis. This is true even for relatively high "quality" settings which result in only modest compression. At high-compression settings, even visual examination of images may be affected due to aliasing of edges and lines, loss of resolution, and suppression of contrast.

Figure 27 shows a real-world image that has been JPEG compressed by about 30:1 and then reconstructed. The major features are still quite recognizable, but upon closer inspection there are many artefacts present. Besides the blocky appearance, fine details are missing or altered. There has also been a slight shift in the colors present. For measurement purposes, the image fidelity has been seriously compromised. **Figure 28** shows a much simpler test image and the results of compression. As the degree of compression is increased, the artefacts become more and more serious until the figure is entirely unrecognizable.

All of the commonly used compression methods take advantage of the fact that human vision tolerates (and detects) less spatial resolution in the color information than in the brightness. This provides a useful way to detect that compression have been applied. **Figure 29** shows the hue channel from the images from **Figure 27**; the uncompressed image has values of hue that vary from pixel to pixel, while the compressed version shows the loss of spatial resolution in the color data.



Figure 27. *Example image ("Flowers") and the reconstruction after JPEG compression by a factor of 29:1. Note the differences in highly textured areas at the center of each flower, artefacts along edges of petals, and the discontinuity in the stem at the left.*

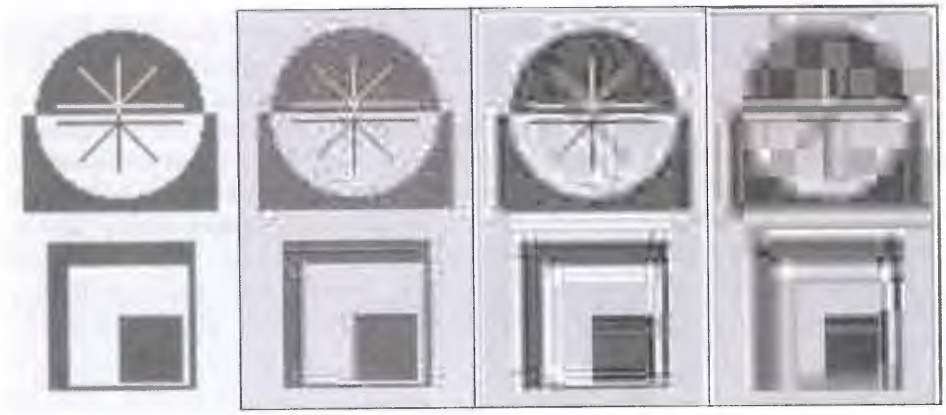


Figure 28. *A test pattern and the result of JPEG compression by factors of 3.5:1, 5.8:1, and 7.9:1 (from left to right).*

JPEG compression has been accepted for use in consumer applications and for transmission of images on the Web primarily because most images contain a wealth of redundant information, and human vision and understanding can recognize familiar objects in familiar settings based on only a few clues. For scientific imaging purposes the artefacts introduced by lossy compression are unacceptable. **Figure 30** shows an example. The image of the film has been compressed by less than 17:1 and visually appears little changed, but a plot of the average intensity across the film shows that minor peaks have been completely eliminated.

The proprietary Kodak Photo-CD algorithm is also a transform method, and shares many of the same advantages and drawbacks as the JPEG method. Because it is intended to work from traditional photographic materials, which have a wide latitude, Photo-CD makes some provision for the extended range that may be present in such images. Whereas JPEG separates the image first into HSI components, as discussed in Chapter 1, Photo-CD uses a modification of the YCC format used for broadcast television, which has one luminance (Y) and two chrominance (C_1 and C_2 or U and V) components (one the red-green balance and the other the yellow-blue balance),

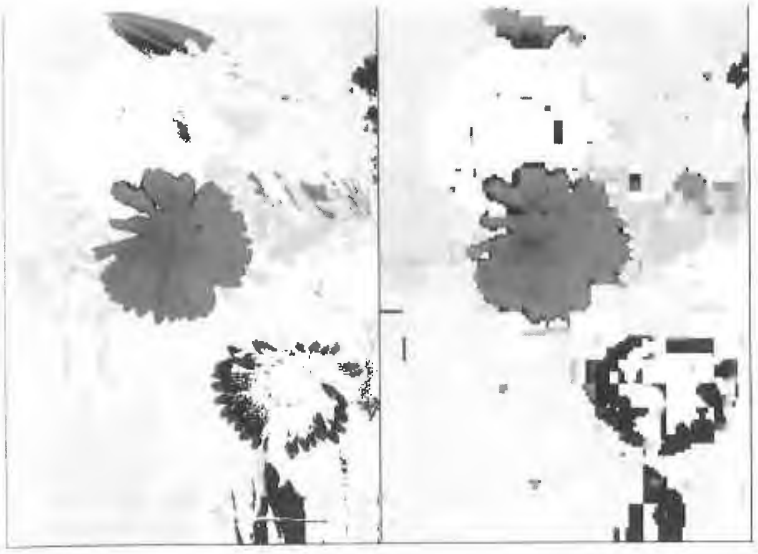


Figure 29. *Hue channels from the images in* ***Figure 27****. The original image (left) has hue values that vary from pixel to pixel, while in the compressed image the hue values are uniform over larger regions.*

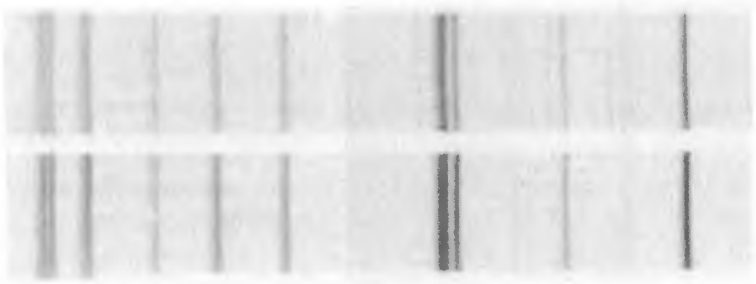
Although they cover an extended range in which $Y' = 1.36 \cdot Y$, and the modified C' components are related to RGB by

$$\begin{aligned} R &= Y' + C_2' \\ G &= Y' - (0.194 \cdot C_1') - 0.509 \cdot C_2' \\ B &= Y' + C_1' \end{aligned} \tag{3}$$

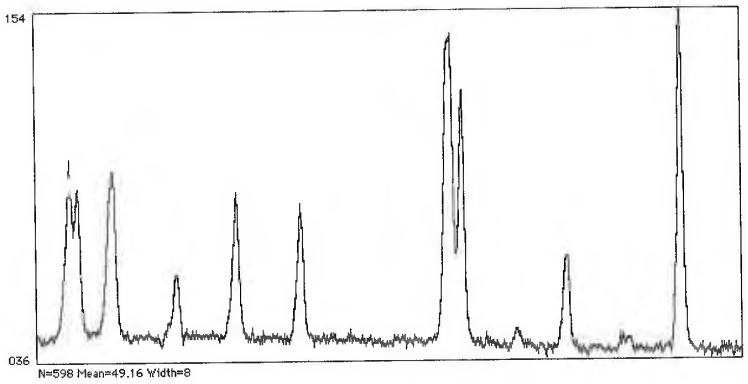
When each of these is mapped to a 256-level (one-byte) value, a nonlinear relationship is used as shown in **Figure 31**. In addition to mapping the film density to the output of a typical CRT display for viewing of the stored image, this method allows recording information beyond the nominal 100% white, which gives visually important highlights to reflections and other features that can be recorded by film, but not by most charge-coupled device (CCD) cameras. These differences, however, do not alter the basic similarity of approach, fidelity, and efficiency that Photo-CD shares with JPEG.

Judging the quality of compressed and restored images is not a simple matter. Calculating the statistical differences between the original pixel values and the reconstructed values is often used, but does not provide a measure that agrees very well with human judgment, nor with the needs of image measurement and analysis. Human vision responds differently to the same absolute variation between pixels depending on whether they lie in light or dark areas of the image. Differences are usually more objectionable in the dark areas, because of the logarithmic response of human vision. Differences are also judged as more important in regions that are smooth, or strongly and regularly patterned, than in areas that are perceived as random. Displacing an edge or boundary because of pixel differences is usually considered more detrimental than a similar brightness change in the interior of a feature.

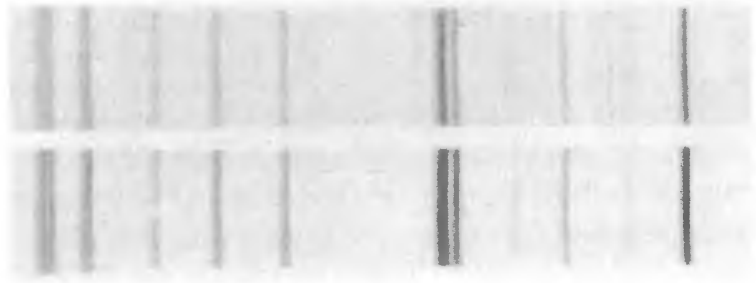
Ranking of image quality by humans is often employed to compare different methods, but even with careful comparisons the results are dependent on lighting conditions, the context of the image and others that have been seen recently, and fatigue. It was the result of extensive human comparisons that led to the selection of the JPEG standard that is now widely implemented. The JPEG method is one of many that rely on first transforming the image from the familiar spatial domain to another one that separates the information present according to a set of basis functions.



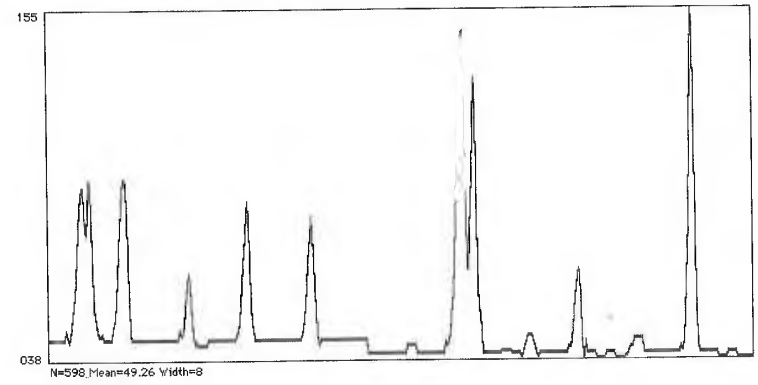
a



b



c



d

Figure 30. *Image of a Debye–Scherer x-ray film before (**a**) and after (**c**) JPEG compression (from 136.3K to 8.1K bytes). The position and density of the vertical lines provide information about crystallographic structure. A plot of the density profile across the images (**b** and **d**, respectively) shows that statistical noise in the spectrum has been removed along with some small peaks, and a new artefact peak introduced, by the compression.*

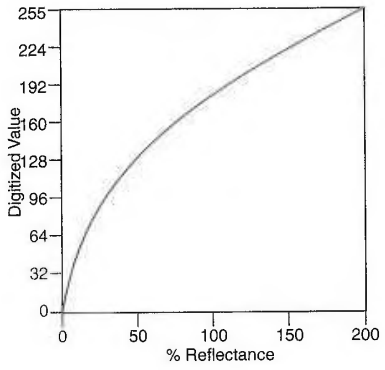


Figure 31. *Nonlinear encoding of brightness and extended luminance range used in Kodak's Photo-CD format.*

Wavelet compression

The virtue of transforming the image into another space is that it collects the information together in a different way. The Fourier method (and its close cousin the DCT used in JPEG compression) separates the information present according to frequency and orientation. If a complete set of these functions is used, it can exactly and completely reconstruct the original image. Therefore, using the transform by itself is not a lossy method; but if the complete set of functions is used, the numbers required to specify the magnitude and phase take up just as much storage space as the original image. In fact, usually they take up more because the magnitude and phase values are real numbers, and the original pixel values are likely to have been integers that require less storage space.

Of course, many other sets of basis functions besides the sinusoid functions are used in the Fourier approach. Some provide a more efficient representation of the image, in which more of the terms are very small, while some are simply easier to compute. One of the more recently popular methods is the wavelet transform, which offers some computational advantages and can even be obtained optically with lenses and masks (McAulay et al., 1993), or electronically with a filter bank.

The wavelet transform provides a progressive or "pyramidal" encoding of the image at various scales, which is more flexible than conventional windowed approaches like the Fourier transform. The wavelets comprise a normalized set of orthogonal functions on which the image is projected (Chui, 1992). The wavelet functions are localized rather than extending indefinitely beyond the image as sinusoids do, so the wavelet transform tends to deal better with the edges of regions and of the image, and its use for compression avoids the "blockiness" or "quilting" sometimes seen in JPEG compression, where the image is subdivided into 8×8 pixel blocks before performing a discrete cosine transform.

Several different wavelet functions are commonly used. The simplest is the Haar, which is just a square step as shown in **Figure 32**, that can be shifted across the image and expanded horizontally. Other wavelet functions, such as the Daubechies order-4 wavelet shown, are less obvious but work the same way and may reduce visual artefacts when compression is performed as described below (Welstead, 1999). Just as the Fourier summation of sinusoids can be used to reconstruct any arbitrary function, so can the summation of wavelet functions (Daubechies, 1992, 1996; Mallat, 1989).

To understand the process of the wavelet transform using the Haar functions, consider the following process. First, find the average intensity for the entire image, and put this value into a single pixel in the upper left corner of the transform (which will be the same size as the original). Now subdivide the image into quadrants, and find the difference between the average values of

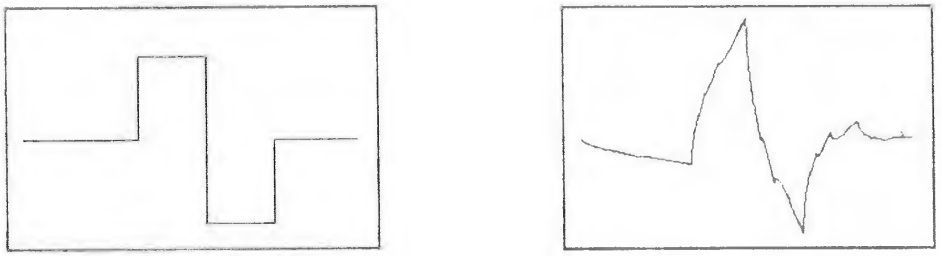


Figure 32. Two wavelets: (a) Haar; (b) Daubechies D4.

each quadrant and the global mean. Place these values into a 2 × 2 block of pixels. Similarly place the difference between each of the two left hand quadrants and the corresponding right hand quadrants into one pixel, and the similar vertical differences into one pixel each, as shown in **Figure 33**. Repeat this process by subdividing the quadrants, and continue until the individual pixels have been reached.

It is clear that the squares and rectangles in the transform that contain the higher frequency differences (which lie downwards and to the right) are much larger and represent much more data

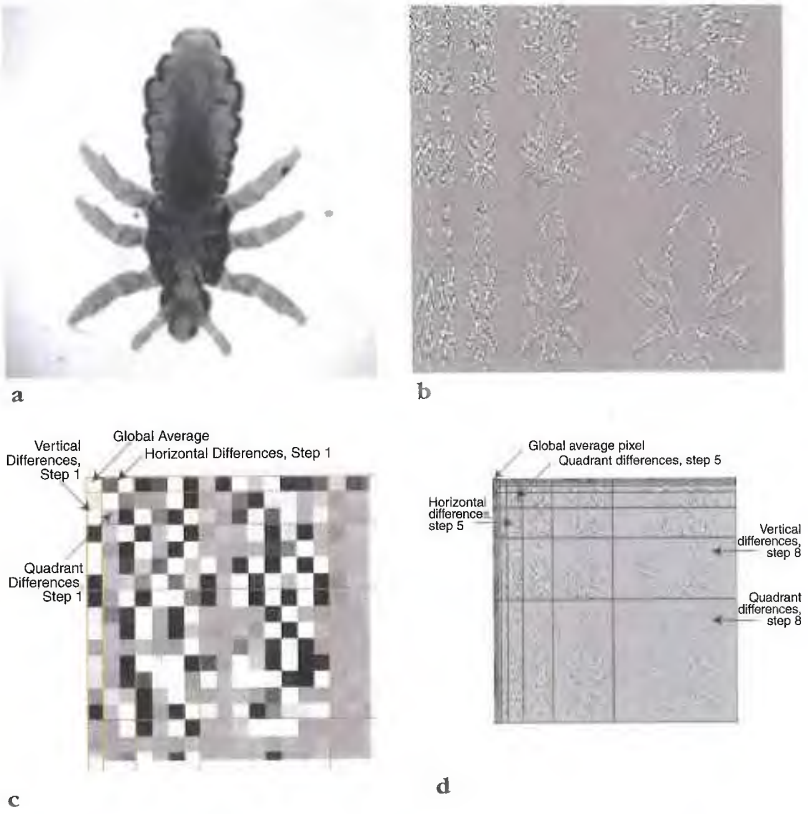


Figure 33. The wavelet transform process applied to an example image ("Bug"). The transform (b) consists of the differences between average values at different scales. These are marked and labeled for the upper left corner of the transform (c) and for the entire transform (d).

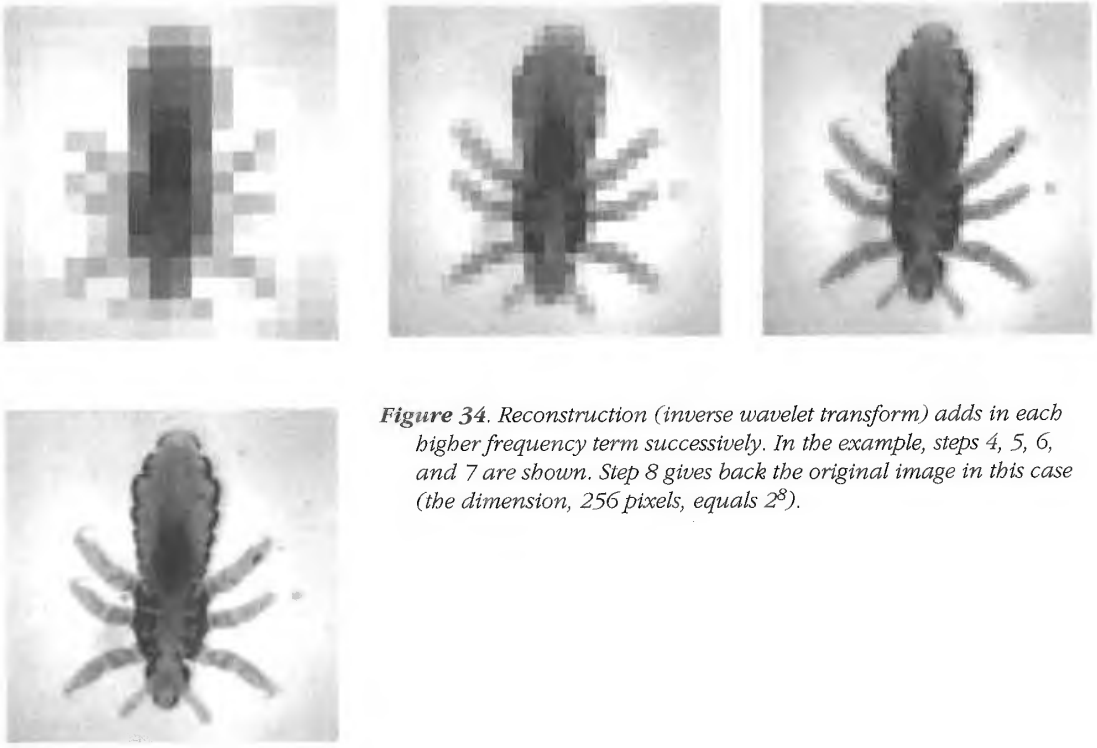


***Figure 34**. Reconstruction (inverse wavelet transform) adds in each higher frequency term successively. In the example, steps 4, 5, 6, and 7 are shown. Step 8 gives back the original image in this case (the dimension, 256 pixels, equals 2^8).*

than those corresponding to low frequency information. The subdividing process amounts to working with higher and higher frequencies in the original image, and the reconstruction of the original image can be carried out using selected frequencies. **Figure 34** shows the process of reconstruction of the image in **Figure 33**, by adding back the differences between pixels at progressively smaller steps. The process reconstructs the original image exactly, without loss. Applications of this approach to a wide variety of time- and space-dependent signals are taking advantage of its efficiency and compactness. In some cases, analysis of the coefficients can provide information about the image content, just as is commonly done with the Fourier transform.

In principle, all these transforms take up just as much space as the original and are loss-free. To achieve compression, the coefficients in the transform are quantized, just as for the JPEG cosine transform. Lossy compression of the image is accomplished by quantizing the magnitudes of the various terms in the transform so they can be represented with fewer bits, and eliminating those that are very small. This can reduce the amount of space required, but also produces loss in the quality of the reconstructed image and introduces various artefacts.

Figure 35 shows the flowers image from **Figure 27** after wavelet compression. Although the artefacts are not visually objectionable in the reconstructed image, examining the difference between the original and the compressed image reveals many differences in detail such as color variations and shifts in the location of edges, which would seriously impact quantitative measurements.

The justification for compression is that the loss of particular terms from the transform may not be visible or at least objectionable to a human viewer, because the information each term represented was spread throughout the entire image and may have contributed little to any particular feature. This is not always the case. The selection of terms with a small amplitude often means the elimination of high frequency information from the image, and this may be important to define edges and boundaries in particular. Just as the most visually objectionable defect in the JPEG cosine



Figure 35. *Wavelet transform applied to the "flowers" image from* ***Figure 27****. Subtracting the original from the reconstructed image shows the differences, which are greatest in those areas of the image with more detail.*

transform is "blocky" artefacts, so the most annoying aspect of the wavelet transform is that details are suppressed more in "busy" parts of the image, so that it is difficult to predict which details will survive and which will not.

Compression using lossy methods is often suitable for images to be used in particular applications such as printing in which the printing device itself may set an upper limit to the fidelity that can be reproduced. It has already been suggested that one use of compression may be to store multiple copies of an image in a database, each with appropriate quality for a specific purpose such as printing or viewing as a thumbnail for searching and recognition. It is also important to determine just how much loss of quality is produced by varying degrees of image compression, so that artefacts capable of altering the analysis of reconstructed images are avoided.

Another application in which compression is normally used (wavelet compression, in fact) is fingerprint files, used by law enforcement agencies for identification. The spacing of the friction ridges on fingertips varies only slightly from one individual to another, so the corresponding set of frequencies is preserved while lower frequencies (e.g., due to shading in the image) and higher frequencies (e.g., due to noise or dust) are discarded.

Fractal compression

Barnsley (Barnsley and Hurd, 1993) has shown another way to transform an image, using what are often referred to as self-affine distortions of the image (shrinking and displacing copies of the original) as the basis functions. In principle, a complete set of these operations would give a set of parameters as large as the original image. Yet, for many images, a small number of functions are required to reconstruct the original image with acceptable fidelity, providing significant compression. The best-known example of this self-affine image generation is the fern produced by iteratively combining smaller copies of the same basic shape.

The four rules in **Table 3** are able to generate a realistic image of a fern. Each rule corresponds to one rotation, displacement and shrinkage of a sub-element of the structure. The rules are applied by starting at any point, selecting one of the rules (with the frequency shown by the probability values, from 1% to 84%), and then moving from the current point to the next point according to the rule. This point is plotted and the procedure iterated to produce the entire figure. The more points, the better the definition of the result as shown in **Figure 36**. The entire fern with 20,000 points shows the self-similarity of the overall shape. As a portion of the image is blown up for examination, more points are required. Finally, the limit of magnification is set by the numerical precision of the values in the computer, as indicated in the figure.

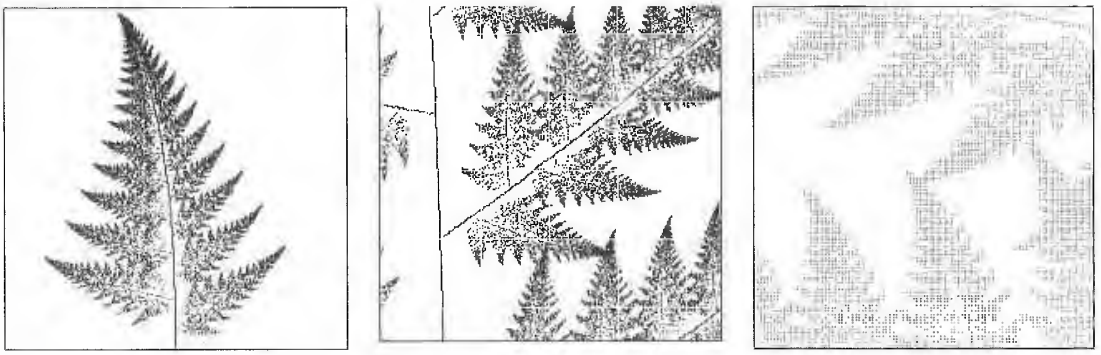
Table 3. Transformations (basis functions) for the Fern image (Figure 36)

1 ($p = 0.840$)	$x' = +0.821 \cdot x + 0.845 \cdot y + 0.088$
	$y' = +0.030 \cdot x - 0.028 \cdot y - 0.176$
2 ($p = 0.075$)	$x' = -0.024 \cdot x + 0.074 \cdot y + 0.470$
	$y' = -0.323 \cdot x - 0.356 \cdot y - 0.260$
3 ($p = 0.075$)	$x' = +0.076 \cdot x + 0.204 \cdot y + 0.494$
	$y' = -0.257 \cdot x + 0.312 \cdot y - 0.133$
4 ($p = 0.010$)	$x' = +0.000 \cdot x + 0.000 \cdot y + 0.496$
	$y' = +0.000 \cdot x + 0.172 \cdot y - 0.091$

From a mathematical point of view, the four transformations are basis functions or mappings which can be added together in proportion (the p, or probability values) to produce the overall object. The same principle has been applied to the compression of photographic images with grey scale or color pixels, where it is known as the Collage Theorem. That there must be such basis functions, or rules for the self-affine distortions, and that they are in principle discoverable, is known (Barnsley et al., 1986; Barnsley, 1988; Khadivi, 1990; Barnsley and Sloan, 1992; Barnsley and Hurd, 1993). The problem of finding the correct mappings is, however, far from trivial. Knowing that such functions must exist gives few leads to discovering them. Each mapping consists of a combination of translation, scaling, rotation, and warping. One proprietary method for finding them has been patented (U.S. Patent #5065447). There are formal procedures for performing fractal compression (Fisher et al., 1992), although the results are not necessarily optimal. It has been shown that nonlinear self-affine transforms can also be used, and may be more efficient.

This technique is described as "fractal compression" because the reconstruction is carried out iteratively (as shown for the fern), and because it provides ever finer levels of detail. In fact, the method can be continued to produce reconstructed images with detail at a much finer scale than the original used to find the basis functions. Such detail looks very impressive, because the enlarged image never shows flat or smooth areas that indicate loss of resolution. Of course, the detail is not real. It is generated under the assumption that whatever patterns are present at large scale in the image are also present with progressively less amplitude at all finer scales. Unlike the JPEG and wavelet methods, fractal compression is not symmetrical. The time needed to compress the image is typically much greater than that needed to reconstruct it.

Fractal compression also has characteristic artefacts. **Figure 37** shows the "Flower" image again, after a 40:1 compression. Subtracting the original from the reconstructed image shows that the colors have been significantly altered, features have been shifted in position, and details inserted that were not present in the original.



a b c

Figure 36. Image of a fern generated with the four transformation rules shown in the text. The structure remains self-similar when expanded, except for the limitation of finite numerical precision in the computer, which rounds off the values in the 100 times expanded image: *(a) fern image with 20,000 points; (b) 5× expansion; (c) 100× expansion.*

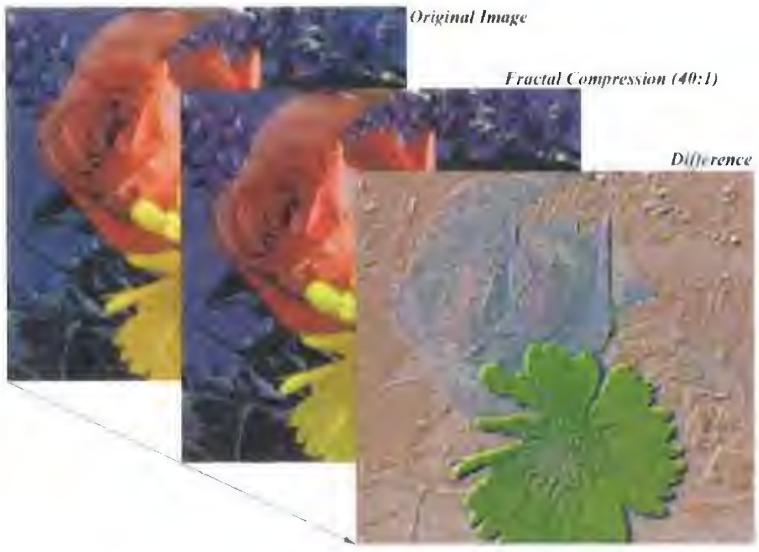


Figure 37. *Fractal compression of the "flowers" image from* ***Figure 27****. Color changes, shifts in position of features, and insertion of detail are all evident in the difference image.*

Digital movies

There is growing interest in using small computers to produce, edit and display digital moving pictures. Two emerging standards (or sets of standards) are Apple's Quicktime and Microsoft's Video for Windows. Both of these methods allow the use of various compression techniques, usually called CODECs (**CO**mpressor-**DEC**ompressors). A CODEC is a program implementation of one of the compression algorithms such as those described earlier. In fact, JPEG is one of the methods provided by currently existing codecs available for both Quicktime and Video for Windows. Some of the codecs run in software only, and others use additional hardware. For JPEG, this can be either a digital signal processor chip programmed to execute the discrete cosine transform, or a dedicated chip designed specifically for the JPEG algorithm.

The JPEG method, however, does not provide as much compression for digital movies as is needed to compress an entire movie onto a single CD-ROM, to broadcast high-definition television (HDTV) images, or to allow digital video conferencing with images transmitted over relatively low bandwidth networks because it does not take into account the considerable redundancy of sequential images. In most cases, a series of images on movie film or videotape has large areas of pixels that are not changing at all with time, or are doing so in relatively smooth and continuous ways. It should be possible to compress these, and in fact quite high compression ratios can be tolerated because for such areas the needed quality of the reconstructed image is not high. Human vision tends to ignore portions of images in which nothing interesting is happening, and no changes are occurring.

The MPEG standards follow the same approach as JPEG. They are based primarily on the need to compress images in order to transmit "consumer" video and the forthcoming HDTV, in order to compress the images enough to permit transmission within the narrow bandwidth used by present television stations, and to fit entire movies onto DVD disks. Of course, for more technical purposes such as tracking features in a sequence of images, or measuring density or color changes as a function of time, different requirements may place different limits on the acceptable quality of reconstruction.

For sequences of images, an additional compression based on similarity between successive frames is employed. MPEG adds several additional steps to reduce the amount of data that must be transmitted. It looks for blocks of similar pixels in successive images, even if they have moved slightly. Only an average of 2 frames per second are normally sent in their entirety. The rest are either encoded as differences from preceding frames, or as interpolations between frames. This approach allows overall compression to approach 200:1 with "acceptable" visual appearance.

A typical HDTV proposal presents 1920 × 1080 pixel images at the rate of 30 per second, for a total data rate exceeding 1 gigabit per second. The images use progressive scan (not interlaced), square pixels, and a wider image aspect ratio of 16:9 instead of the 4:3 used in NTSC video (and most computer display screens). A typical broadcast TV station has a bandwidth that is 250 times too small. Without examining the debate over HDTV, whose standards are still evolving, we can consider if and how application of these methods may be useful for technical images.

It is important to remember that the criterion for these compression methods is the visual acceptance of the reconstructed image, as it is displayed on a television screen. For instance, the MPEG standard, similar to today's broadcast television, encodes the chrominance (color) information with less precision than it does the luminance (brightness) information. This is because tests indicate that human vision of images displayed on a CRT is more tolerant of variations in color than in brightness. The use of images for technical purposes, or even their presentation in other forms such as printed hardcopy viewed by reflected light, may not be so forgiving.

The audio compression used in MPEG is used by itself for music recording (the popular MP3 format). The requirement for fidelity is that high frequencies must be well enough preserved so that (for example) overtone sequences allow the listener to distinguish an oboe from a muted trumpet.

Most of the moving-picture compression methods use key frames, which are compressed using the same method as a still picture. Then, for every image in the sequence that follows the key frame, the differences from the previous image are determined and compressed. For these difference images, as we have seen previously, the magnitude of the values is reduced. Consequently, in performing a further compression of this image by keeping only the "most important" basis functions, the number of terms eliminated can be increased significantly and much higher levels of compression achieved.

Similar to JPEG, MPEG consists of several options, some of which require more computation but deliver more compression. For instance, motion compensation provides a higher degree of compression because it identifies an overall translation in successive images (for instance, when the camera is slowly panned across a scene), and adjusts for that before comparing locations in one image with the previous frame. The MPEG approach is asymmetric: it requires more computation to compress the original data than to decompress it to reconstruct the images.

One of the consequences of this approach to compression is that it is intended only to go forward in time. From a key frame and then successive differences, you can reconstruct each following image, but you cannot easily go back to reconstruct a previous image, except by returning to the nearest preceding key frame and working forward from there. In principle, one key frame at the beginning of each scene in a movie should be enough. In practice, key frames are inserted periodically.

Other compression methods are being developed for sequential images. One is predictive vector quantization that attempts to locate boundaries in each image, track the motion of those boundaries, and use prediction to generate successive images. Sequences of 8-bit grey scale images compressed to an average data rate of less than 0.5 bits per pixel have been reported (Nicoulin et al., 1993; Wu and Gersho, 1993; Wen and Lu, 1993; Hwang et al., 1993). Fractal compression has also been extended to deal with image sequences (Li et al., 1993).

High compression ratios for moving images are appropriate for video conferencing, where the image quality only has to show who is speaking and perhaps what they are holding. For many consumer applications, in which the final image will be viewed on a television screen of only modest resolution, adequate image quality can be achieved at high compression ratios. Tests with human television viewers have long suggested that it is the quality of the sound that is most important, and significant noise and other defects in the individual images are not objectionable.

For most technical applications, however, the types of artefacts produced by still image compression are not acceptable, and the additional ones introduced as a result of temporal compression make matters worse. The user intending to perform analysis of images from a sequence should certainly begin with no compression at all, and only accept specific compression methods if tests indicate they are acceptable for the particular purposes for which the images are to be used.

3

Correcting Imaging Defects

The first class of image processing operations, which is considered in this chapter, is those procedures applied to correct some of the defects in as-acquired images that may be present due to imperfect detectors, limitations of the optics, inadequate or nonuniform illumination, or an undesirable viewpoint. It is important to emphasize that these are corrections that are applied after the image has been digitized and stored, and therefore will be unable to deliver the highest quality result that could have been achieved by optimizing or correcting the acquisition process in the first place.

Of course, acquiring an optimum-quality image is sometimes impractical. If the camera can collect only a small number of photons in a practical time or before the scene changes, then the noise present in the image cannot be averaged out by acquiring and adding more photons or video frames, and other noise reduction means are needed. If the source of illumination cannot be controlled to be perfectly centered and normal to the viewed surface (for instance the sun), or if the surface is curved instead of planar, then the image will have nonuniform illumination that must be corrected afterwards. If the viewpoint cannot realistically be adjusted (for instance the path of a space probe or satellite), or if the surface is irregular (as in the case of a metal fracture), then some parts of the scene will be foreshortened; this must be taken into account in comparing sizes or measuring distances.

Even in typical laboratory setups such as light microscopy, keeping the instrument in ideal alignment may be very time-consuming, and achieving adequate stability to collect dim fluorescence images for a long time very difficult, so that it becomes more practical to trade off some of the ultimately achievable image quality for convenience and speed, and to utilize image processing methods to perform these corrections. When the first space probe pictures were obtained and the need for this type of correction was first appreciated, it required lengthy computations on moderate sized computers to apply them. It is now possible to implement such corrections on desktop machines in times measured in seconds, so that they can be practically applied to routine imaging needs.

Contrast expansion

In Chapter 1, it was noted that the typical digitization process for images produces values from 0 (black) to 255 (white), producing one-byte (8-bit) values, or for color images one byte each for red, green and blue. If the camera and digitizer have more precision, the values may have 10, 12, or

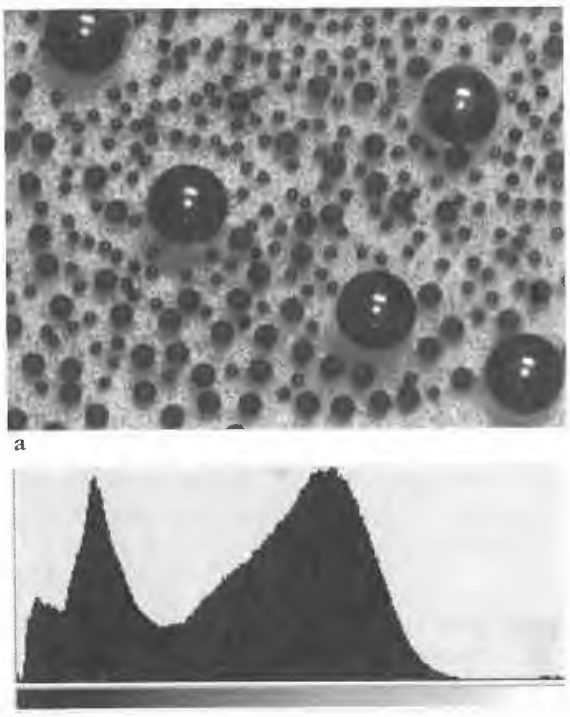


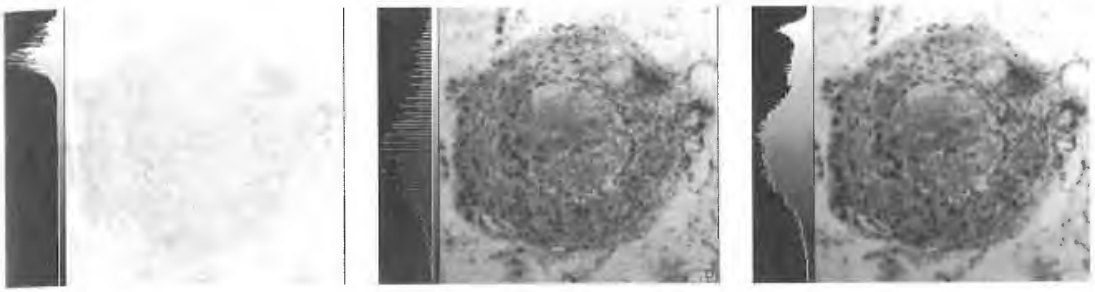
Figure 1. *Image of ball bearings and lead shot from a macro camera. The histogram (b) covers the full range from white to black. The small number of white pixels correspond to the specular light reflections from the larger balls.*

even more bits of precision and typically occupy two bytes each. Although this is the full dynamic range available to the output of the camera sensors, there is no reason that the actual image data will cover the full range. In many situations, the recorded image will have a much smaller range of brightness values, which may either lie toward the middle of the range (intermediate grey values), or toward either the bright or dark end of the range.

The image histogram, a plot of the number of pixels with each possible brightness level, is a valuable tool for examining the contrast in the image. **Figure 1** shows an example image in which the histogram covers the full dynamic range and indicates good contrast. The small number of very bright pixels corresponds to the specular highlights where light is reflected from the largest polished metal balls.

If the inherent range of variation in brightness of the image is much smaller than the dynamic range of the camera, subsequent electronics, and digitizer, then the actual range of numbers will be much less than the full range of 0 through 255. **Figure 2a** shows an example. The specimen is a thin section through tissue, with a blood vessel shown in cross section in a bright field microscope. Illumination in the microscope and light staining of the section produce very little total contrast. The histogram shown next to the image is a plot of the number of pixels at each of the 256 possible brightness levels. The narrow peak indicates that only a few of the levels are represented.

Visibility of the structures present can be improved by stretching the contrast so that the values of pixels are reassigned to cover the entire available range. **Figure 2b** shows this. The mapping is linear and one-to-one. This means that the darkest pixels in the original image are assigned to black, the lightest images are assigned to white, and intermediate grey values in the original image are given new values which are linearly interpolated between black and white. All of the pixels in the original image that had one particular grey value will be assigned the same grey value in the resulting image, but it will be a different one than in the original.



a b c

Figure 2. Light microscope image of a blood vessel:
(a) *has very low initial contrast, as shown by its brightness histogram;*
(b) *Linear expansion of the brightness range by manipulating the display shows a full range of black to white values but causes gaps in the histogram;*
(c) *Acquiring the image with optimum illumination and camera exposure produces similar contrast but without gaps in the histogram, and with less noise.*

This histogram plotted with the image in the figure now shows counts of pixels for grey levels that are spread out across the available brightness scale, but notice that most of the grey values still show zero values in the histogram, indicating that no pixels have those values. The reassignment of grey values has increased the visual contrast for the pixels present, but has not increased the ability to discriminate subtle variations in grey scale that were not recorded in the original image. It has also magnified the brightness difference associated with noise in the original image.

Figure 2c shows the same field of view recorded to utilize the entire range of the camera and digitizer. This may require adjusting the illumination, camera gain or exposure time, etc., and generally requires a lot of trial and error in the settings. The mean brightness of various structures is similar to that shown in **Figure 31b**; however, all of the 256 possible grey values are now present in the image, and very small variations in sample density can now be distinguished or measured in the specimen.

This problem is not restricted to bright images. **Figure 3a** shows a dark image from a scanning electron microscope, with its histogram. The structures on the integrated circuit are revealed when the contrast is stretched out (**Figure 3c**), but this also increases the visibility of the noise, or random "speckle" variations for pixels that represent the same structure and ideally should be uniform in brightness. The problem of image noise will be dealt with as the next topic in this chapter.

These cases are rather extreme. It is often not practical to adjust the illumination, camera gain, etc., to exactly fill the available pixel depth (number of grey levels that can be digitized or stored). Furthermore, increasing the brightness range too much can cause pixel values at the dark and/or light ends of the range to exceed the digitization and storage capacity and to be clipped to the limiting values, which also causes loss of information. **Figure 4** shows an example, a micrograph of a cross section through an enamel coating in which the grey scale range of the bubbles is fine but the polished flat surfaces were brighter than the white limit of the camera and consequently were clipped to a value of 255, losing any detail that might have been present. To avoid such problems, it is common for images to be acquired that do not completely cover the available brightness range.

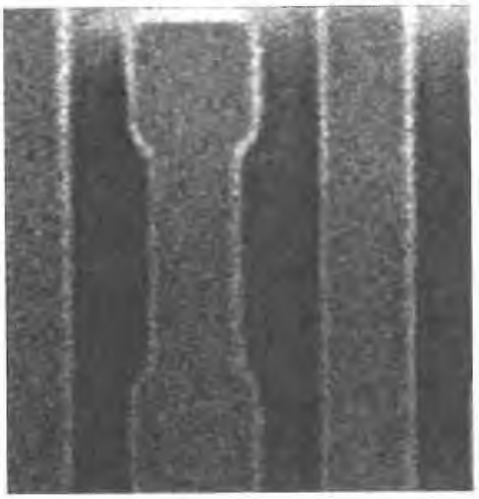
When contrast expansion is applied to color images, the correct procedure is to convert the image from its stored RGB (red, green, blue) format to HSI (hue, saturation, intensity), and expand the intensity scale while leaving the color information unchanged, as shown in **Figure 5.** This prevents color shifts that would occur if the individual red, green, and blue histograms were linearly expanded to full scale, as shown in **Figure 6**. Such shifts are especially prevalent if the original image consists primarily of just a few colors, or is predominantly of one color.



a



b



c

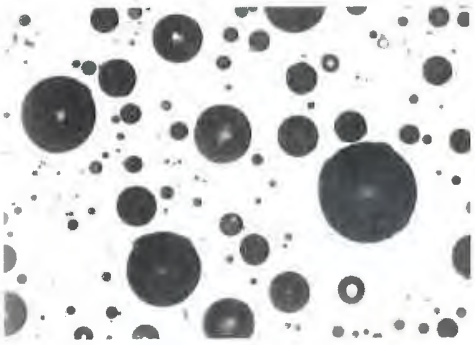


d

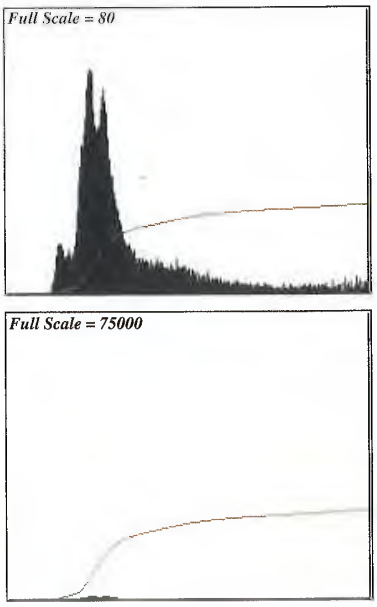
Figure 3. *A rather dark scanning electron microscope (SEM) image of lines on an integrated circuit*
(a) *with its histogram*
(b) *After linear contrast expansion*
(c) *the features and noise are both more visible, and the histogram*
(d) *shows gaps between values.*

If these images still have enough different grey levels to reveal the important features in the specimen, then linear contrast expansion may be a useful and acceptable method to increase the viewer's visual discrimination. More important, this expansion may make it possible to more directly compare images acquired with slightly different brightness ranges by adjusting them all to the same expanded contrast scale. Of course, this only works if the brightest and darkest class of features are present in all of the images and fields of view.

Other manipulations of the pixel brightness values can also be performed. These are described as point operations, meaning that the new values assigned depend only on the original pixel value and not on any of its neighbors, and one-to-one, meaning that all pixels which originally had a single grey scale value are assigned to another single value, although the process may not be linear. An example would be one which converted brightness to density, which involves a logarithmic relationship. For color images, a transfer function can be used to correct colors for distortion due to the color temperature of the light source, or for atmospheric scattering and absorption in satellite images. These functions may be implemented with either a mathematical function or a lookup table.



a

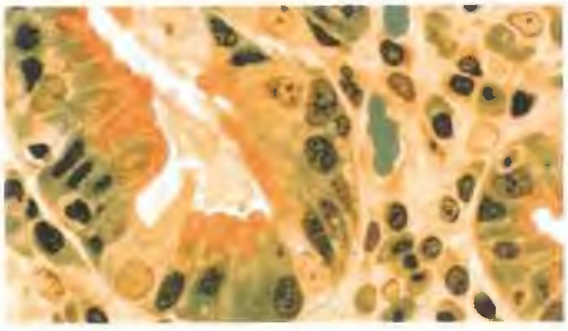


b

Figure 4. *Light micrograph of polished section through an enamel coating (**a**). The histogram (**b**) shows that the majority of the pixels are full white, indicating that they were brighter than the dynamic range and were clipped to 255. It is necessary to expand the vertical scale of the histogram to see the peaks for the pixels with grey values corresponding to the bubbles in the coating.*



a



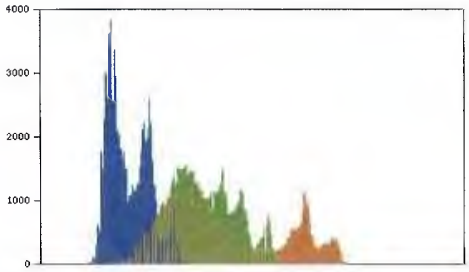
b

Figure 5. *Light microscope image of stained tissue (**a**), and the result of expanding contrast by converting to HSI space and linearly expanding the intensity only (**b**).*

Chapter 4 illustrates many of these contrast manipulations to enhance the visibility of structures present in the image.



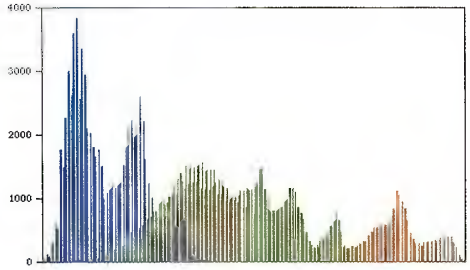
a



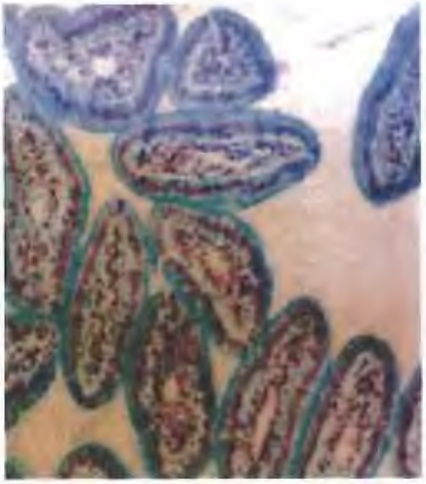
b



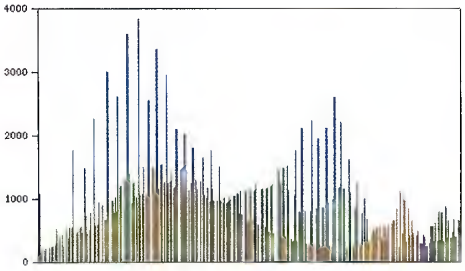
c



d



e



f

Figure 6. *Light microscope image of stained tissue* ***(a)*** *with histograms of the red, green and blue values* ***(b)****. Expanding the intensity while leaving hue and saturation unchanged* ***(c)*** *expands the range of RGB values but they do not cover the full range* ***(d)****. Expanding the RGB values individually to full range* ***(f)*** *produces color shifts in the image* ***(e)****.*

Noisy images

The linear expansion of contrast shown in the examples described earlier is often accompanied by an increased visibility for noise (random fluctuations in pixel values) that may be present. Noise is an important defect in images that can take many different forms and arise from various sources.

In Chapter 1, improvement in image quality (technically, signal-to-noise ratio) by averaging a number of frames was demonstrated. One unavoidable source of noise is counting statistics in the image detector due to a small number of incident particles (photons, electrons, etc.). This is particularly the case for x-ray images from the scanning electron microscope (SEM), in which the ratio of incident electrons to detected x-rays may be from 10^5 to 10^6. In fluorescence light microscopy, the fluoresced-light photons in a narrow wavelength range from a dye or activity probe may also produce very dim images, compounded by the necessity of acquiring a series of very-short-duration images to measure activity as a function of time.

Noisy images may also occur due to instability in the light source or detector during the time required to scan or digitize an image. The pattern of this noise may be quite different from the essentially Gaussian noise due to counting statistics, but it still shows up as a variation in brightness in uniform regions of the scene. One common example is the noise in field emission SEM images due to the variation in field emission current at the tip. With a typical time constant of seconds, the electron emission may shift from one atom to another, producing a change of several percent in the beam current. The usual approach to minimize the effects of this fluctuation in the viewed image is to use scan times that are either much shorter or longer than the fluctuation time.

Similar effects can be seen in images acquired using fluorescent lighting, particularly in light boxes used to illuminate film negatives, resulting from an interference or beat frequency between the camera scan and flickering of the fluorescent tube. This flickering is much greater than that of an incandescent bulb, whose thermal inertia smooths out the variation in light emission due to the alternating current. When the noise has a characteristic that is not random in time and does not exhibit "normal" statistical behavior, it is more difficult to place numeric descriptors on the amount of noise. However, many of the same techniques can be used, usually with somewhat less efficacy, to reduce the noise. Periodic noise sources, resulting typically from electrical interference or vibration, may be removed using Fourier transform representation of the image, discussed in Chapter 5.

Assuming that an image represents the best quality that can practically be obtained, this section will deal with ways to suppress noise to improve the ability to visualize and demarcate for measurement the features which are present. The underlying assumptions in all of these methods are that the pixels in the image are much smaller than any of the important details, and that for most of the pixels present, their neighbors represent the same structure. Various averaging and comparison methods can be applied based on these assumptions.

These assumptions are very much the same as those inherent in classical image averaging, in which the assumption is that pixel readings at each location at different times represent the same structure in the viewed scene. This directly justifies averaging or integrating pixel readings over time to reduce random noise. When the signal varies in other ways described previously, other methods such as median filtering (discussed in the following paragraphs) can be used. These methods are directly analogous to the spatial comparisons discussed here except that they utilize the time sequence of measurements at each location.

Important differences exist between noise reduction by the use of frame averaging to combine many sequential readouts from a camera, and the use of a camera that can integrate the charge internally before it is read out. The latter mode is employed in astronomy, fluorescence microscopy, and other applications to very faint images, and is sometimes called "staring" mode

since the camera is simply open to the incident light. The two methods might seem to be equivalent, since both add together the incoming signal over some period of time. It is important, however, to understand that there are two quite different sources of noise to be considered.

In a camera used in staring mode, the electrons collected in each transistor on the charge coupled device (CCD) array include those produced by incoming photons and some from the dark current in the device itself. This current is strongly influenced by thermal noise, which can move electrons around at room temperature. Cooling the chip, either by a few tens of degrees with a Peltier cooler or by hundreds of degrees with liquid nitrogen or even liquid helium, can reduce this thermal noise dramatically. An infrared camera must be cooled more than a visible light camera, because the light photons themselves have less energy and so the production of a signal electron takes less energy, so that more dark current would be present at room temperature.

Cameras intended for staring application or long time exposures often specify the operating time needed to half-fill the dynamic range of the chip with dark current. For an inexpensive Peltier-cooled camera, the useful operating time may be several minutes. For a high quality device used for professional astronomy and cooled to much lower temperatures, it may be tens of hours. Collecting an image in staring mode for that length of time would raise the dark level to a medium grey, and any real signal would be superimposed on that background. The production of thermal electrons is a statistical process, therefore, not all pixels will have the same background level. Fluctuations in the background thus represent one type of noise in the image that may be dominant for these types of applications to very dim images. In most cases, this source of noise is small compared to the readout noise.

All cameras have some readout noise. In the typical CCD camera, the electrons from each transistor must be transferred across each line in order to be read out as a voltage to the computer. In a CCD camera, more transfers are needed to shift the electrons from one side of the image to the amplifier and so the resulting noise is greater on one side of the image than on the other. In an interline transfer camera, a separate row of transistors adjacent to each detector is used instead, somewhat reducing the readout noise. Of course, additional sources of noise from the other associated electronics (clock signals, wiring from the camera to the digitizer, pickup of electrical signals from the computer itself, and so forth) may degrade the signal even more. But even if those other sources of noise are minimized by careful design and proper wiring, there is an irreducible amount of noise superimposed on the signal each time the camera image is read out, amplified and digitized.

This noise is generally random, and hence is as likely to reduce as to increase the brightness of any particular pixel. **Figure 7** shows an example of two successive video frames acquired using a good quality digital camera and normal lighting. The images look essentially identical except for the motion of the clock pendulum. However, subtracting one frame from the other (and expanding the contrast) shows the pixel noise present in the images. With digital cameras, as discussed in Chapter 1, the normal procedure is to acquire a single image using an appropriate exposure time, and then transfer the digitized values to the computer. However, some camera designs that are not cooled may acquire several shorter exposures and average the results in the computer.

With video cameras, averaging many frames together causes the random noise from readout to partially cancel while the signal continues to add up. The small well size of the transistors on the very tiny chip make long exposures impractical The consequence is that frame averaging can reduce the relative magnitude of this source of noise in the image, but cannot eliminate it. **Figure 8** shows two video frames of the clock, acquired in dim light with high amplifier gain. The individual images show visible noise, which is even more apparent in the difference image.

For a very dim image, the optimum situation would be to integrate the signal within the camera chip, but without allowing any single pixel to reach saturation. Then reading the data out once will



Figure 7. *Two images of the same view, acquired from a digital camera, and the difference between them (pixels enlarged and contrast expanded to show detail).*



Figure 8. *Two images of the same view, acquired as sequential video frames, and the difference between them (contrast expanded to show detail).*

produce the minimum readout noise and the best image. For a brighter image, or for a camera than can only function at video rates, it may be acceptable to average together a sufficient number of frames to reduce the pixel variations due to random readout noise. This integration may be done in the frame grabber or in the computer program. The frame grabber is generally restricted by the amount of on-board memory, but can often collect every frame with no loss of data. The computer program is more flexible, but the time required to add the frames together may result in discarding some of the video frames, so that the total acquisition takes longer (which may not be a problem if the image does not vary with time). Of course, in fluorescence microscopy bleaching may occur and it may be desirable to collect all of the photons as rapidly as possible, either by frame averaging, or, for a very dim image, by using staring mode if the camera is capable of this.

Figure 9 shows a comparison of two SEM images taken at different scan rates. The fast scan image collects few electrons per pixel and so has a high noise level that obscures details in the

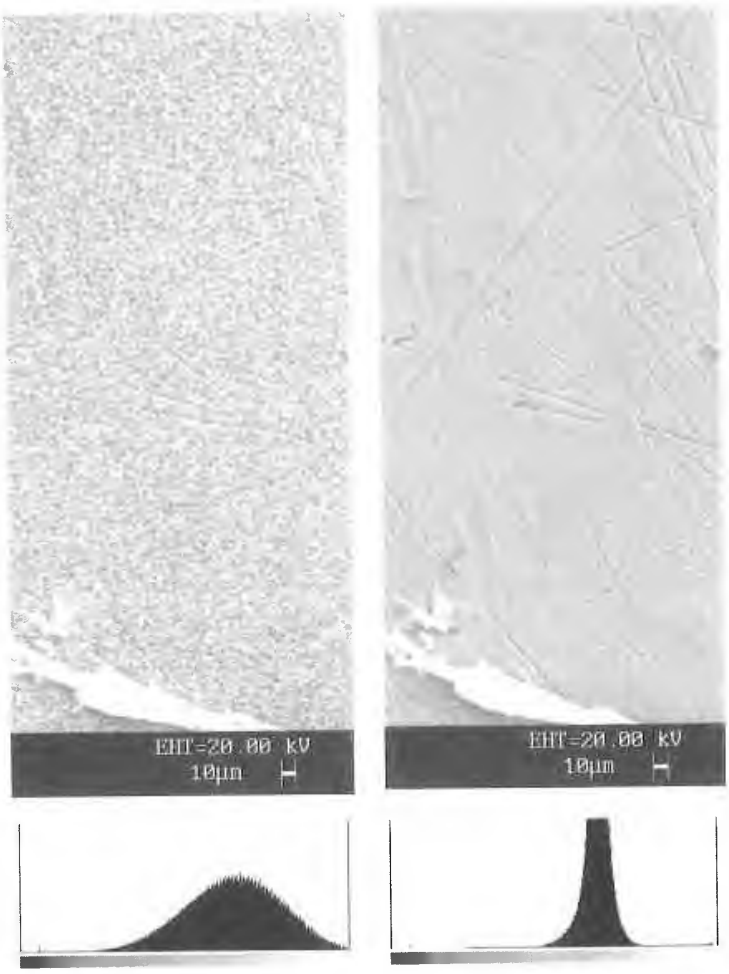


Figure 9. SEM images of a scratched metal surface: *(a) 1-second scan; (b) histogram of (a); (c) 20 second scan; (d) histogram of (c).*

image. Slowing the scan rate down from one second to 20 seconds increases the amount of signal and reduces the noise. The histograms show that the variation of brightness within the uniform region is reduced, which is why the visibility of detail is improved.

Neighborhood averaging

The simplest form of spatial averaging is simply to add together the pixel brightness values in each small region of the image, divide by the number of pixels in the neighborhood, and use the resulting value to construct a new image. **Figure 10** shows that this essentially produces an image with a smaller number of pixels. The block size in **Figure 3b** is 4×4, so that sixteen pixel values are added. The noise in this image is random due to the counting statistics of the small number of photons, so the improvement in image quality or signal-to-noise ratio is just the square root of 16, or a factor of 4; however, the image lateral resolution is seriously impacted and the small structures in the image can no longer be separately discerned.

The more common way to accomplish neighborhood averaging is to replace each pixel with the average of itself and its neighbors. This is often described as a "kernel" operation, since implementation can be generalized as the sum of the pixel values in the region multiplied by a set of

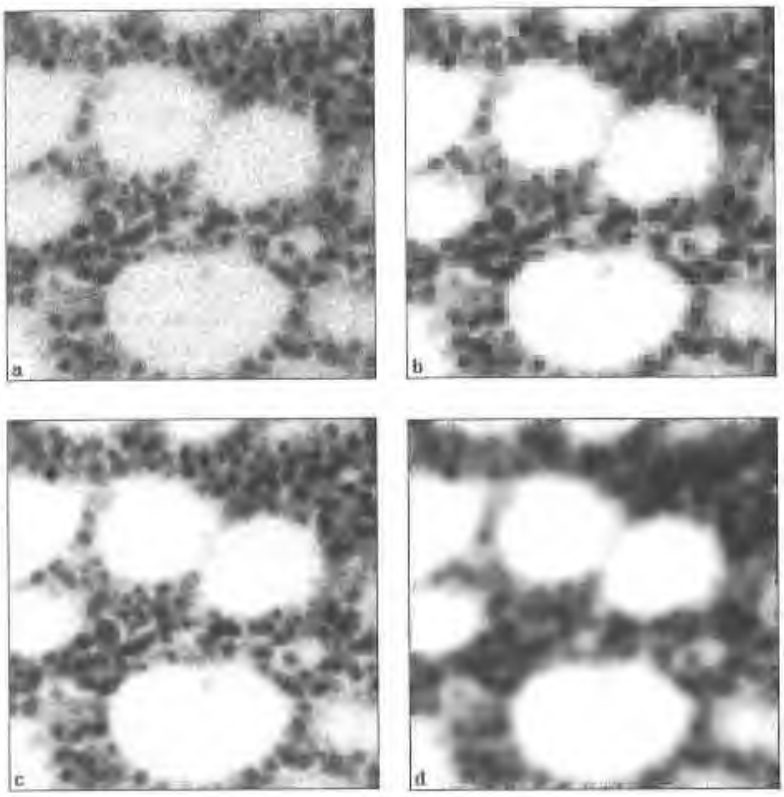


Figure 10. Smoothing by averaging: *(a) noisy original image (fluorescence light microscopy of bone marrow); (b) each 4 × 4 block of pixels averaged; (c) each pixel replaced by the average of itself and its eight neighbors in a 3 × 3 square block; (d) each pixel replaced by the average of itself and its 120 neighbors in an 11 × 11 square.*

integer weights. The process is also called a "convolution" because it corresponds to a procedure performed in Fourier space (discussed in Chapter 5).

$$P^{*}_{x,y} \frac{\sum_{i,j=-m}^{+m} W_{i,j} \cdot P_{x+i,y+j}}{\sum_{i,j=-m}^{+m} W_{i,j}} \qquad (1)$$

Equation 1 shows the calculation performed over a square of dimension $2m + 1$, which is odd. The neighborhood sizes thus range from 3 × 3, upward to 5 × 5, 7 × 7, etc. It is possible to use non-square regions; for larger neighborhoods an approximation of a circle is preferable to a square but somewhat more difficult to implement. The array of weights W for a simple neighbor averaging contains only 1s, and for a 3 × 3 region could be written as:

1	1	1
1	1	1
1	1	1

where we understand the convention that these coefficients are to be multiplied by pixels that surround the central pixel, the total normalized by dividing by the sum of weights, and the value written to the location of the central pixel to form a new image.

As a matter of practicality, because storage space is never unlimited, it is common to write the new image back into the same memory as the original; however, when this is done it is important to use the original pixel values for the summation, and not those new ones which have already been calculated for some of the neighbors. This requires keeping a copy of a few lines of the image during the process.

Neighborhood operations including kernel multiplication are usually applied symmetrically around each pixel. This creates a problem for pixels nearer to the edge of the image than the half-width of the neighborhood. Various approaches are used to deal with this problem, including designing special asymmetrical kernels or rules along edges or in corners, assuming that the image edges are mirrors, so that each line of pixels within the image is duplicated beyond it, or assuming that the image wraps around so that the left edge and right edge, and the top and bottom edges, are continuous. In the examples shown here, an even simpler approach has been used: the processing is restricted to that portion of the image where no edge conflicts arise. This leaves lines of unprocessed pixels along the edges of the images, equal in width to half the dimension of the neighborhood. None of these approaches is entirely satisfactory, and in general most processing operations sacrifice some pixels from the image edges.

Figure 10 shows the effect of smoothing using a 3×3 neighborhood average and also an 11×11 neighborhood size. The noise reduction is much greater with the larger region, but is accompanied by a significant blurring of the feature edges.

The amount of blurring can be reduced and more control exerted over the neighborhood averaging procedure by using weight values that are not 1. For example, the values

1	2	1
2	4	2
1	2	1

have several attractive characteristics. First, the central 4 which multiplies the original pixel contents is the largest factor, causing the central pixel to dominate the average and reducing blurring. The values of 2 for the four orthogonally touching neighbors and 1 for the four diagonally touching neighbors acknowledge the fact that the diagonal pixels are in fact farther away from the center of the neighborhood (actually by the factor $\sqrt{2}$). Finally, these weights have a total value of 16, which is a power of two and consequently very easy to divide in a computer implementation of this averaging procedure.

Similar sets of ad hoc weight values can be devised in various sizes. Integers are commonly used to speed implementation, and storage requirements are modest. We will see in Chapter 5, in the context of processing images in frequency space, that these kernels can be analyzed quite efficiently in that domain to understand their smoothing properties. It turns out from that analysis that one of the very useful "shapes" for a weight kernel is that of a Gaussian. This is a set of integers that approximate the profile of a Gaussian function along any row, column or diagonal through the center. It is characterized by a standard deviation, expressed in terms of pixel dimensions. The size of the kernel is generally made large enough that adding another row of terms would insert negligibly small numbers, ideally zeroes into the array, but of course some zero values will be present anyway in the corners since they are farther from the central pixel.

Choosing the actual integers is something of an art, at least for the larger kernels, since the goal is to approximate the smooth analytical curve of the Gaussian, although the total of the weights must usually be kept smaller than some practical limit to facilitate the computer arithmetic (Russ, 1995d). Several Gaussian kernels are shown below, with standard deviations that increase in geometric proportions from less than one pixel to many pixels. The standard deviation for these kernels is the

radius (in pixels) containing 68% of the integrated magnitude of the coefficients, or the volume under the surface if the kernel is pictured as a 3D plot of the integer values. This is a two-dimensional generalization of the usual definition of standard deviation; for a one-dimensional Gaussian distribution, 68% of the area under the curve lies within ± one standard deviation.

Of course, the kernel size increases with the standard deviation as well; the total width of the kernel is at least six times the standard deviation. For the largest examples illustrated below, only the upper left quadrant of the symmetrical array is shown. Repeated application of a small kernel, or the sequential application of two or more kernels, is equivalent to the single application of a larger one, which can be constructed as the convolution of the two kernels (applying the weighting factors from one kernel to those in the other as if they were pixel values, and summing and adding to generate a new, larger array of weights).

σ = 0.391 pixels (3 × 3)

1	4	1
4	12	4
1	4	1

σ = 0.625 pixels (5 × 5)

1	2	3	2	1
2	7	11	7	2
3	11	17	11	3
2	7	11	7	2
1	2	3	2	1

σ = 1.0 pixels (9 × 9)

0	0	1	1	1	1	1	0	0
0	1	2	3	3	3	2	1	0
1	2	3	6	7	6	3	2	1
1	3	6	9	11	9	6	3	1
1	3	7	11	12	11	7	3	1
1	3	6	9	11	9	6	3	1
1	2	3	6	7	6	3	2	1
0	1	2	3	3	3	2	1	0
0	0	1	1	1	1	1	0	0

σ = 1.6 pixels (11 × 11)

1	1	1	2	2	2	2	2	1	1	1
1	2	2	3	4	4	4	3	2	2	1
1	2	4	5	6	7	6	5	4	2	1
2	3	5	7	8	9	8	7	5	3	2
2	4	6	8	10	11	10	8	6	4	2
2	4	7	9	11	12	11	9	7	4	2
2	4	6	8	10	11	10	8	6	4	2
2	3	5	7	8	9	8	7	5	3	2
1	2	4	5	6	7	6	5	4	2	1
1	2	2	3	4	4	4	3	2	2	1
1	1	1	2	2	2	2	2	1	1	1

σ = 2.56 pixels (15 × 15)

2	2	3	4	5	5	6	6	6	5	5	4	3	2	2
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
6	8	11	13	16	18	19	20	19	18	16	13	11	8	6
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
2	2	3	4	5	5	6	6	6	5	5	4	3	2	2

σ =4.096 pixels (21 × 21)

5	6	7	8	9	10	11	12	13	13	13	13	13	12	11	10	9	8	7	6	5
6	7	9	10	11	12	14	15	15	16	16	16	15	15	14	12	11	10	9	7	6
7	9	10	12	13	15	16	17	18	18	19	18	18	17	16	15	13	12	10	9	7
8	10	12	13	15	17	18	20	21	21	21	21	21	20	18	17	15	13	12	10	8
9	11	13	15	17	19	21	22	23	24	24	24	23	22	21	19	17	15	13	11	9
10	12	15	17	19	21	23	25	26	27	27	27	26	25	23	21	19	17	15	12	10
11	14	16	18	21	23	25	27	28	29	29	29	28	27	25	23	21	18	16	14	11
12	15	17	20	22	25	27	29	30	31	31	31	30	29	27	25	22	20	17	15	12
13	15	18	21	23	26	28	30	32	33	33	33	32	30	28	26	23	21	18	15	13
13	16	18	21	24	27	29	31	33	34	34	34	33	31	29	27	24	21	18	16	13
13	16	19	21	24	27	29	31	33	34	34	34	33	31	29	27	24	21	19	16	13
13	16	18	21	24	27	29	31	33	34	34	34	33	31	29	27	24	21	18	16	13
13	15	18	21	23	26	28	30	32	33	33	33	32	30	28	26	23	21	18	15	13
12	15	17	20	22	25	27	29	30	31	31	31	30	29	27	25	22	20	17	15	12
11	14	16	18	21	23	25	27	28	29	29	29	28	27	25	23	21	18	16	14	11
10	12	15	17	19	21	23	25	26	27	27	27	26	25	23	21	19	17	15	12	10
9	11	13	15	17	19	21	22	23	24	24	24	23	22	21	19	17	15	13	11	9
8	10	12	13	15	17	18	20	21	21	21	21	21	20	18	17	15	13	12	10	8
7	9	10	12	13	15	16	17	18	18	19	18	18	17	16	15	13	12	10	9	7
6	7	9	10	11	12	14	15	15	16	16	16	15	15	14	12	11	10	9	7	6
5	6	7	8	9	10	11	12	13	13	13	13	13	12	11	10	9	8	7	6	5

σ = 6.536 pixels (29 × 29, upper left corner)

```
 7  8  9 10 10 11 12 13 13 14 14 15 15 15 15 ...
 8  9 10 11 11 12 13 14 15 15 16 16 16 17 17 ...
 9 10 11 12 13 13 14 15 16 17 17 18 18 18 18 ...
10 11 12 13 14 15 16 17 17 18 19 19 20 20 20 ...
10 11 13 14 15 16 17 18 19 20 20 21 21 21 21 ...
11 12 13 15 16 17 18 19 20 21 22 22 23 23 23 ...
12 13 14 16 17 18 19 20 21 22 23 24 24 24 24 ...
13 14 15 17 18 19 20 22 23 24 24 25 25 26 26 ...
13 15 16 17 19 20 21 23 24 25 26 26 27 27 27 ...
14 15 17 18 20 21 22 24 25 26 27 27 28 28 28 ...
14 16 17 19 20 22 23 24 26 27 28 28 29 29 29 ...
15 16 18 19 21 22 24 25 26 27 28 29 30 30 30 ...
15 16 18 20 21 23 24 25 27 28 29 30 30 30 31 ...
15 17 18 20 21 23 24 26 27 28 29 30 30 31 31 ...
15 17 18 20 21 23 24 26 27 28 29 30 31 31 31 ...
...
```

σ = 10.486 pixels (43 × 43, upper left corner)

```
 6  6  6  7  7  7  8  8  8  9  9  9  9 10 10 10 10 10 10 10 11 11 ...
 6  6  7  7  7  8  8  8  9  9  9 10 10 10 10 11 11 11 11 11 11 11 ...
 6  7  7  7  8  8  9  9  9 10 10 10 11 11 11 11 11 12 12 12 12 12 ...
 7  7  7  8  8  9  9  9 10 10 10 11 11 11 12 12 12 12 12 12 12 13 ...
 7  7  8  8  9  9  9 10 10 11 11 11 12 12 12 12 13 13 13 13 13 13 ...
 7  8  8  9  9 10 10 10 11 11 12 12 12 13 13 13 13 13 14 14 14 14 ...
 8  8  9  9  9 10 10 11 11 12 12 12 13 13 13 14 14 14 14 14 14 14 ...
 8  8  9  9 10 10 11 11 12 12 13 13 13 14 14 14 15 15 15 15 15 15 ...
 8  9  9 10 10 11 11 12 12 13 13 14 14 14 15 15 15 15 15 16 16 16 ...
 9  9 10 10 11 11 12 12 13 13 14 14 14 15 15 15 16 16 16 16 16 16 ...
 9  9 10 10 11 12 12 13 13 14 14 15 15 15 16 16 16 16 17 17 17 17 ...
 9 10 10 11 11 12 12 13 14 14 15 15 15 16 16 16 17 17 17 17 17 17 ...
 9 10 11 11 12 12 13 13 14 14 15 15 16 16 17 17 17 17 18 18 18 18 ...
10 10 11 11 12 13 13 14 14 15 15 16 16 17 17 17 18 18 18 18 18 18 ...
10 10 11 12 12 13 13 14 15 15 16 16 17 17 17 18 18 18 18 19 19 19 ...
10 11 11 12 12 13 14 14 15 15 16 16 17 17 18 18 18 19 19 19 19 19 ...
10 11 11 12 13 13 14 15 15 16 16 17 17 18 18 18 19 19 19 19 19 19 ...
10 11 12 12 13 13 14 15 15 16 16 17 17 18 18 19 19 19 19 19 20 20 ...
10 11 12 12 13 14 14 15 15 16 17 17 18 18 18 19 19 19 19 20 20 20 ...
10 11 12 12 13 14 14 15 16 16 17 17 18 18 19 19 19 19 20 20 20 20 ...
11 11 12 12 13 14 14 15 16 16 17 17 18 18 19 19 19 20 20 20 20 20 ...
11 11 12 13 13 14 14 15 16 16 17 17 18 18 19 19 19 20 20 20 20 20 ...
...
```

σ = 16.777 pixels (55 × 55, upper left corner)

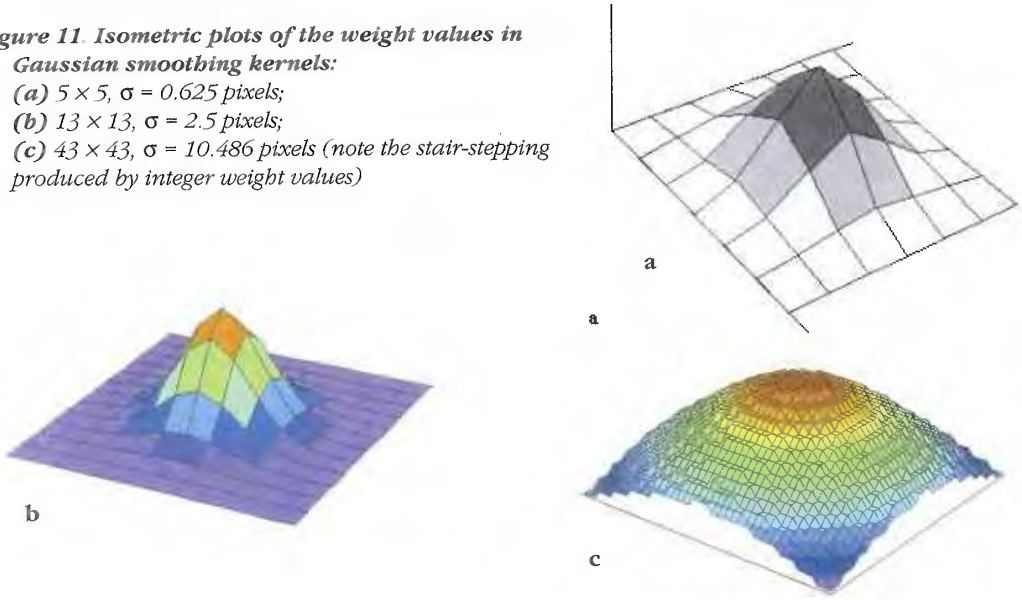
```
 5  5  5  5  5  5  5  6  6  6  6  6  6  6  6  7  7  7  7  7  7  7  7  7  7  7  7  ...
 5  5  5  5  5  6  6  6  6  6  6  6  6  6  7  7  7  7  7  7  7  7  7  7  7  7  7  7  ...
 5  5  5  5  6  6  6  6  6  6  6  6  6  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  ...
 5  5  5  6  6  6  6  6  6  6  7  7  7  7  7  7  7  7  7  7  7  8  8  8  8  8  8  8  ...
 5  5  6  6  6  6  6  6  6  7  7  7  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  ...
 5  6  6  6  6  6  6  6  7  7  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  8  8  ...
 5  6  6  6  6  6  6  7  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  8  8  8  8  ...
 6  6  6  6  6  6  7  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  ...
 6  6  6  6  6  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  9  9  9  9  9  9  9  ...
 6  6  6  6  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  9  9  9  9  9  9  9  9  ...
 6  6  6  7  7  7  7  7  7  8  8  8  8  8  8  8  8  8  8  9  9  9  9  9  9  9  9  9  ...
 6  6  6  7  7  7  7  7  8  8  8  8  8  8  8  8  9  9  9  9  9  9  9  9  9  9  9  9  ...
 6  6  7  7  7  7  7  7  8  8  8  8  8  8  9  9  9  9  9  9  9  9  9  9  9  9  9  9  ...
 6  7  7  7  7  7  7  8  8  8  8  8  8  9  9  9  9  9  9  9  9  9  9  9  9 10 10 10  ...
 6  7  7  7  7  7  8  8  8  8  8  8  9  9  9  9  9  9  9  9  9  9 10 10 10 10 10 10  ...
 7  7  7  7  7  7  8  8  8  8  8  8  9  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10  ...
 7  7  7  7  7  8  8  8  8  8  8  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10  ...
 7  7  7  7  7  8  8  8  8  8  8  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10  ...
 7  7  7  7  8  8  8  8  8  8  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10 10  ...
 7  7  7  7  8  8  8  8  8  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10 10 10  ...
 7  7  7  7  8  8  8  8  8  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10 10 10  ...
 7  7  7  8  8  8  8  8  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10 10 10 10  ...
 7  7  7  8  8  8  8  8  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10 10 10 10 11  ...
 7  7  7  8  8  8  8  8  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10 11 11 11 11  ...
 7  7  7  8  8  8  8  8  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 11 11 11 11 11  ...
 7  7  7  8  8  8  8  8  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10 11 11 11 11 11  ...
 7  7  7  8  8  8  8  8  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10 11 11 11 11 11  ...
 7  7  7  8  8  8  8  8  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10 11 11 11 11 11  ...
```

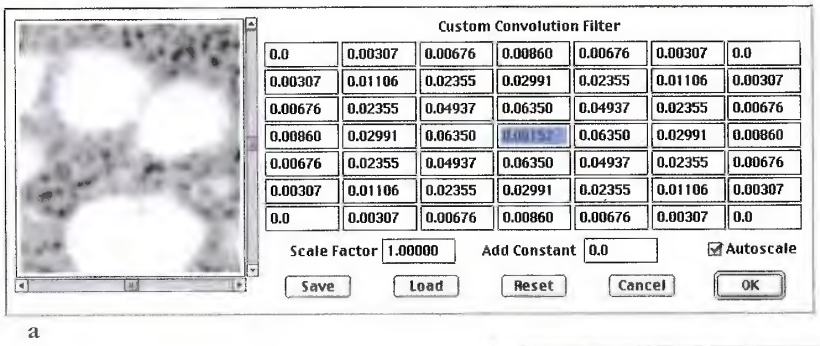
Figure 11 shows several Gaussian weight kernels plotted as isometric views. Notice that the quantization of the weights using integers produces some distortion. Some systems allow entering kernel weights as real numbers as shown in **Figure 12a**. Many systems generate Gaussian kernels as needed, when the user enters the standard deviation value (**Figure 12b**). The actual smoothing operation can be speeded up considerably in the case of Gaussian filters by separating the operation into two simpler ones. Instead of using the entire square array of weights, which for a 15 × 15 kernel would require 225 multiplications and additions, the filter can be separated into a vertical Gaussian blur with a linear array of weights (15 multiplications and additions) followed by a horizontal Gaussian blur (another 15 for a total of 30). The implementation of many convolution kernels can be separated in this way, but for purposes of understanding the algorithms it is probably better to consider the entire array of weights.

In subsequent sections on image processing, we will see other uses for kernels in which the weights are not symmetrical in magnitude and not all positive. The implementation of the kernel will remain the same, except that when negative weights are present the normalization is usually performed by division by the sum of the positive values only (because in these cases the sum of all the weights is usually zero). For the present our interest is restricted to smoothing of noise in images.

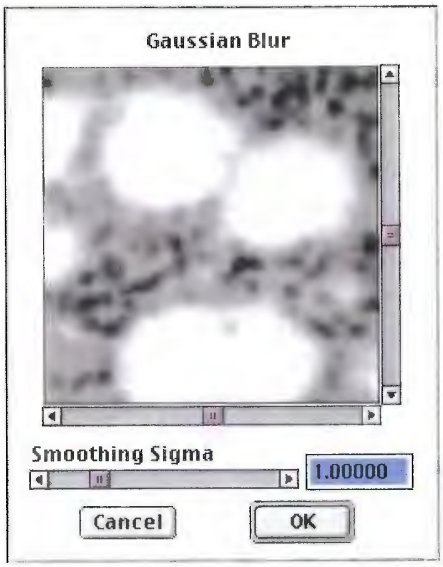
Figure 13 shows the same noisy image as **Figure 10**, along with an image of the same region using image averaging to reduce the statistical noise, as described in Chapter 1. The figure also shows an enlargement of a portion of the image in which the individual pixels can be discerned, as an aid to judging the pixel-to-pixel noise variations in uniform regions and the sharpness of boundaries

Figure 11. Isometric plots of the weight values in Gaussian smoothing kernels:
(a) *5 × 5, σ = 0.625 pixels;*
(b) *13 × 13, σ = 2.5 pixels;*
(c) *43 × 43, σ = 10.486 pixels (note the stair-stepping produced by integer weight values)*





a



b

Figure 12. Gaussian smoothing to reduce noise:
(a) *entering real values for a Gaussian filter with standard deviation = 1.0 pixels;*
(b) *specifying the standard deviation for automatic generation of the weights. The preview displays also expand the image contrast.*

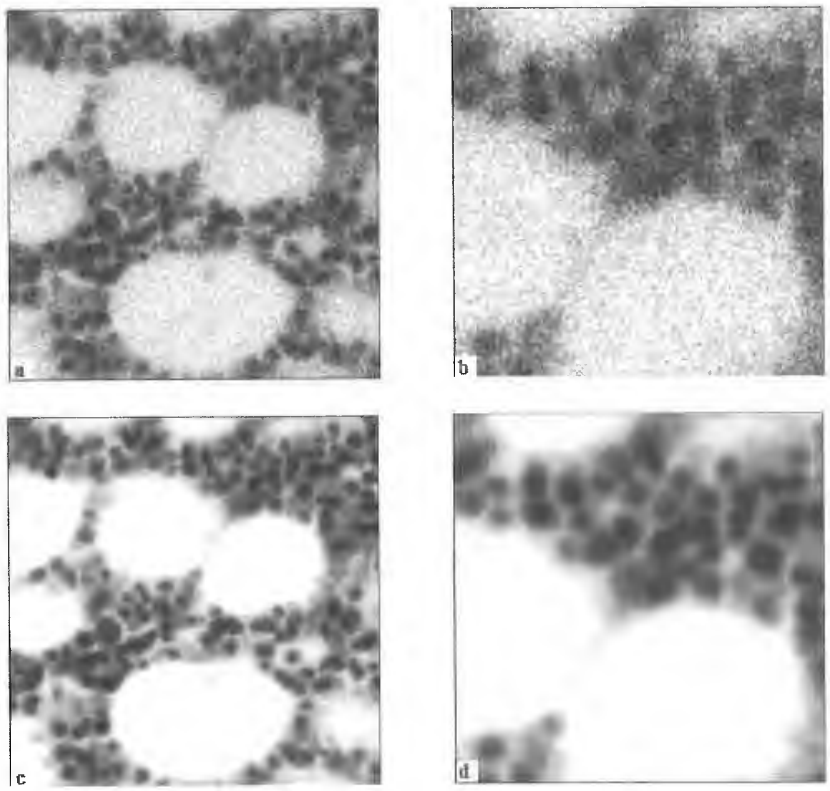


Figure 13. Reduction of noise by temporal averaging:
(a) *original image (same as Figure 10);*
(b) *enlarged to show individual pixel detail;*
(c) *averaging of 256 frames;*
(d) *enlarged to show individual pixel detail.*

between different structures. Applying smoothing with a Gaussian kernel with a standard deviation of 1.0 pixels produces the improvement in quality shown in **Figure 14**.

This type of averaging can reduce visible noise in the image, but it also blurs edges, displaces boundaries, and reduces contrast. It can even introduce artefacts when two nearby structures are averaged together in a way that creates an apparent feature between them. **Figure 15** shows an example in which the lines of the test pattern are blurred by the 11×11 averaging window causing false lines to appear between them.

In Chapter 4 on image enhancement, the use of kernels in which a ring of negative weight values surrounds a positive central peak will be shown. The purpose of this modification is to sharpen edges and avoid some of the blurring that is produced by smoothing out noise. The same method has long been used for the smoothing of one-dimensional signal profiles, such as x-ray diffraction patterns, spectra, or time-varying electronic signals. This is often performed using a Savitsky and Golay (1964) fitting procedure. Tables of coefficients published for this purpose (and intended for efficient application in dedicated computers) are designed to be used just as the weighting coefficients discussed previously, except that they operate in only one dimension. The process is equivalent to performing a least-squares fit of the data points to a polynomial. The smoothed profiles preserve the magnitude of steps while smoothing out noise. **Table 1** lists these coefficients for

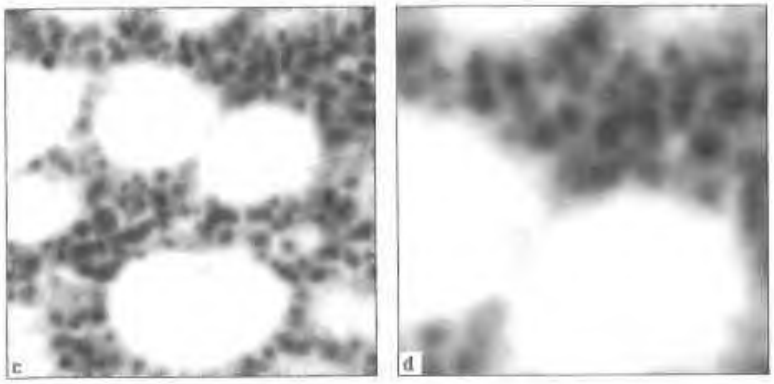


Figure 14. Reduction of noise by Gaussian smoothing (compare with Figure 13a, b): ***(a)*** *application of a Gaussian filter with standard deviation of 1 pixel;* ***(b)*** *enlarged to show individual pixel detail.*

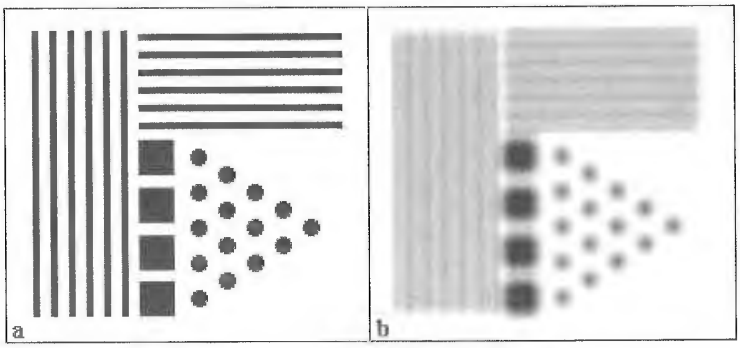


Figure 15. *Artefacts due to smoothing. Applying an 11 × 11 smoothing kernel to the test pattern in image* ***a*** *produces apparent lines between the original ones, as shown in image* ***b****.*

second (quadratic) and fourth (quartic) power polynomials, for fits extending over neighborhoods ranging from 5 to 19 points. These profiles are plotted in **Figure 16**.

This same method can be extended to two dimensions, of course (Edwards, 1982). This can be done either by first applying the coefficients in the horizontal direction and then in the vertical direction, or by constructing a full 2D kernel. **Figure 17** shows the application of a 7 × 7 Savitsky and Golay quadratic polynomial to smooth the image from **Figure 13**.

It is interesting to compare the results of this spatial-domain smoothing to that which can be accomplished in the frequency domain. As discussed in Chapter 5, multiplication of the frequency transform by a convolution function is equivalent to application of a kernel in the spatial domain. The most common noise-filtering method is to remove high frequency information, which represents pixel-to-pixel variations associated with noise. Such removal may be done by setting an aperture on the two-dimensional transform, eliminating higher frequencies, and retransforming. **Figure 18** shows the result of applying this technique to the same image as in **Figure 13**. A circular low-pass filter with radius of 35 pixels and a 10-pixel-wide cosine edge shape (the importance of these parameters is discussed in Chapter 5) was applied. The smoothing is similar to that accomplished in the spatial domain.

As a practical matter, for kernel sizes that are small, or ones that are separable into vertical and horizontal passes like the Gaussian, multiplication and addition of the pixel values in the spatial domain is a faster operation. For kernels that are larger than about 15 × 15 (the limit depends to some extent on the particulars of the individual computer, whether the image can be held entirely in memory or must be accessed from disk, whether multiple processors can be used, etc.) the frequency domain method will be faster; however, because both procedures are mathematically identical, it is generally easier to understand the process based on spatial domain kernels of weights.

Table 1. Savitsky and Golay fitting coefficients

Quadratic polynomial fit

5	7	9	11	13	15	17	19
0	0	0	0	0	0	0	-.0602
0	0	0	0	0	0	-.065	-.0226
0	0	0	0	0	-.0706	-.0186	.0106
0	0	0	0	-.0769	-.0118	.0217	.0394
0	0	0	-.0839	0	.038	.0557	.0637
0	0	-.0909	.021	.0629	.0787	.0836	.0836
0	-.0952	.0606	.1026	.1119	.1104	.1053	.0991
-.0857	.1429	.1688	.1608	.1469	.133	.1207	.1101
.3429	.2857	.2338	.1958	.1678	.1466	.13	.1168
.4857	.3333	.2554	.2075	.1748	.1511	.1331	.119
.3429	.2857	.2338	.1958	.1678	.1466	.13	.1168
-.0857	.1429	.1688	.1608	.1469	.133	.1207	.1101
0	-.0952	.0606	.1026	.1119	.1104	.1053	.0991
0	0	-.0909	.021	.0629	.0787	.0836	.0836
0	0	0	-.0839	0	.038	.0557	.0637
0	0	0	0	-.0769	-.0118	.0217	.0394
0	0	0	0	0	-.0706	-.0186	.0106
0	0	0	0	0	0	-.065	-.0226
0	0	0	0	0	0	0	-.0602

Quartic polynomial fit

5	7	9	11	13	15	17	19
0	0	0	0	0	0	.0464	-.0343
0	0	0	0	0	.0464	-.0464	-.0565
0	0	0	0	.0452	-.0619	-.0619	-.039
0	0	0	.042	-.0814	-.0636	-.0279	.0024
0	0	.035	-.1049	-.0658	-.0036	.0322	.0545
0	.0216	-.1282	-.0233	.0452	.0813	.0988	.1063
.25	-.1299	.0699	.1399	.1604	.1624	.1572	.1494
-.5	.3247	.3147	.2797	.2468	.2192	.1965	.1777
1.5	.5671	.4172	.3333	.2785	.2395	.2103	.1875
-.5	.3247	.3147	.2797	.2468	.2192	.1965	.1777
.25	-.1299	.0699	.1399	.1604	.1624	.1572	.1494
0	.0216	-.1282	-.0233	.0452	.0813	.0988	.1063
0	0	.035	-.1049	-.0658	-.0036	.0322	.0545
0	0	0	.042	-.0814	-.0636	-.0279	.0024
0	0	0	0	.0452	-.0619	-.0619	-.039
0	0	0	0	0	.0464	-.0464	-.0565
0	0	0	0	0	0	.0464	-.0343
0	0	0	0	0	0	0	.0458

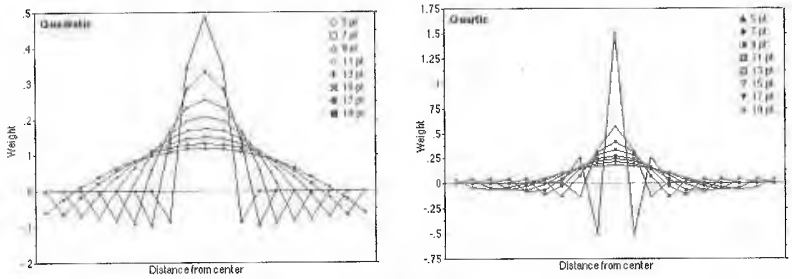


Figure 16. *Savitsky and Golay linear smoothing weights for least squares fitting to quadratic and quartic polynomials.*

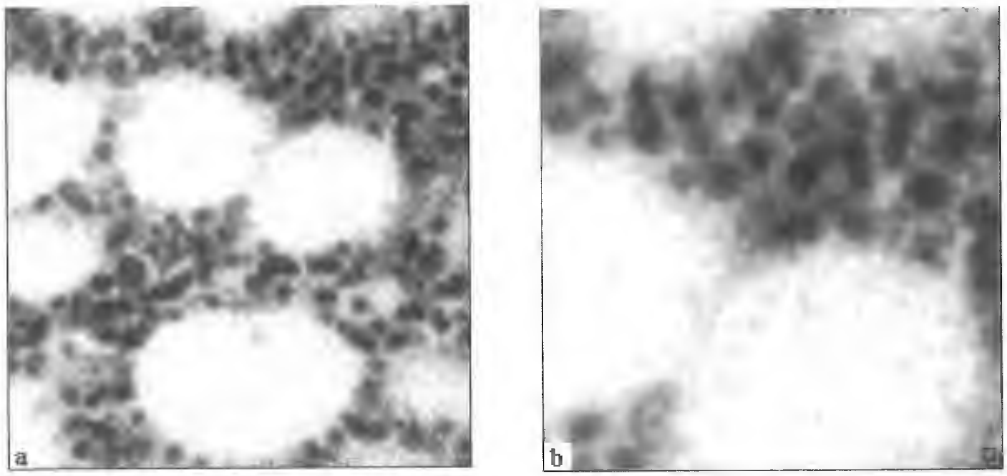


Figure 17. Smoothing with a 7-point wide, quadratic Savitsky and Golay fit:
(a) *same image as* ***Figure 13****, smoothed;*
(b) *enlarged to show pixel detail.*

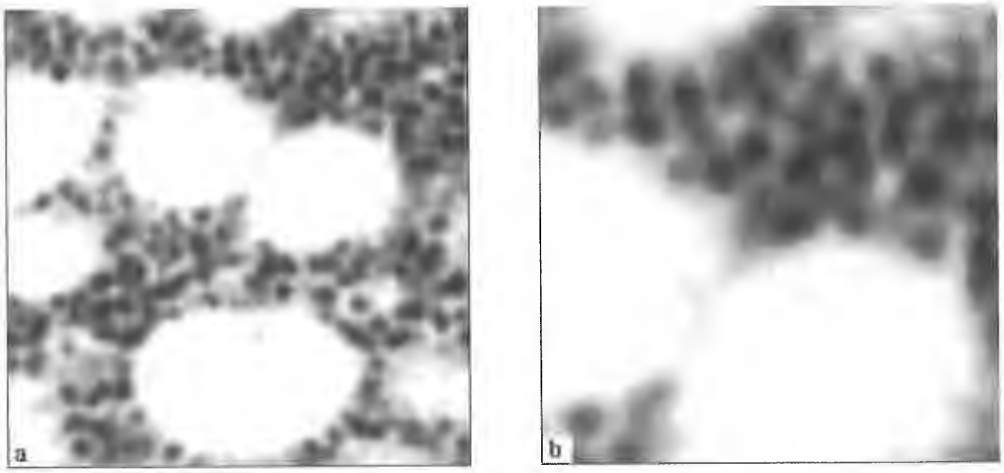


Figure 18. Smoothing in frequency space, using a circular low-pass filter with radius of 35 pixels and a 10-pixel wide cosine edge shape:
(a) *application to image in* ***Figure 13;***
(b) *enlargement to show pixel detail.*

Neighborhood ranking

Smoothing filters do reduce noise, but the underlying assumption is that all of the pixels in the neighborhood represent multiple samples of the same value, in other words, that they all belong to the same feature. Clearly, at edges and boundaries this is not true, and all of the smoothing filters shown above produce some blurring of edges, which is undesirable. The Gaussian filters produce the least edge blurring for a given amount of noise reduction, but cannot eliminate the blurring altogether.

The use of weighting kernels to average together pixels in a neighborhood is a convolution operation, which has a direct counterpart in frequency space image processing. It is a linear operation that uses all of the pixels in the neighborhood, and in which no information is lost from the original image. Other processing operations can be performed in neighborhoods in the spatial domain that also provide noise smoothing. These are not linear and do not utilize or preserve all of the original data.

The most widely used of these methods is based on ranking of the pixels in a neighborhood according to brightness. Then, for example, the median value in this ordered list can be used as the brightness value for the central pixel. As in the case of the kernel operations, this is used to produce a new image and only the original pixel values are used in the ranking for the neighborhood around each pixel.

The median filter is an excellent rejecter of certain kinds of noise, for instance "shot" or impulse noise in which individual pixels are corrupted or missing from the image. If a pixel is accidentally changed to an extreme value, it will be eliminated from the image and replaced by a "reasonable" value, the median value in the neighborhood. This type of noise occurs in CMOS cameras with "dead" transistors that have no output or always put out maximum signals, and in interference microscopes for points on a surface with a locally high slope that returns no light, for example.

Figure 19 shows an example of this type of noise. Ten percent of the pixels in the original image, selected randomly, are set to black, and another ten percent to white. This is a rather extreme amount of noise, but a median filter is able to remove the noise and replace the bad pixels with reasonable values while causing a minimal distortion or degradation of the image. Two different neighborhoods are used: a 3×3 square containing a total of nine pixels, and a 5×5 octagonal (approximately circular) region containing a total of 21 pixels. **Figure 20** shows several of the neighborhood regions often used for ranking. Of course, the computational effort required rises quickly with the number of values to be sorted, even using specialized methods which keep partial sets of the pixels ranked separately so that as the neighborhood is moved across the image, only a few additional pixel comparisons are needed.

Application of a median filter can also be used to reduce the type of random or speckle noise shown before in the context of averaging. **Figure 21** shows the same image as in **Figure 5**, with a 5×5 octagonal median filter applied. There are two principal advantages to the median filter as compared with multiplication by weights. First, the method does not reduce the brightness difference across steps, because the values available are only those present in the neighborhood region, not an average between those values. Second, median filtering does not shift boundaries as averaging may, depending on the relative magnitude of values present in the neighborhood. Overcoming these problems makes the median filter preferred both for visual examination and measurement of images (Huang, 1979, Yang and Huang, 1981).

Figure 22 shows a comparison of uniform averaging, Gaussian filtering, and median filtering applied to a noisy SEM image. The better preservation of edge sharpness by the median filter is apparent. Even with a very large neighborhood (**Figure 22e**) the edges do not shift position, but when the neighborhood radius is larger than the size of any features present, they are eliminated

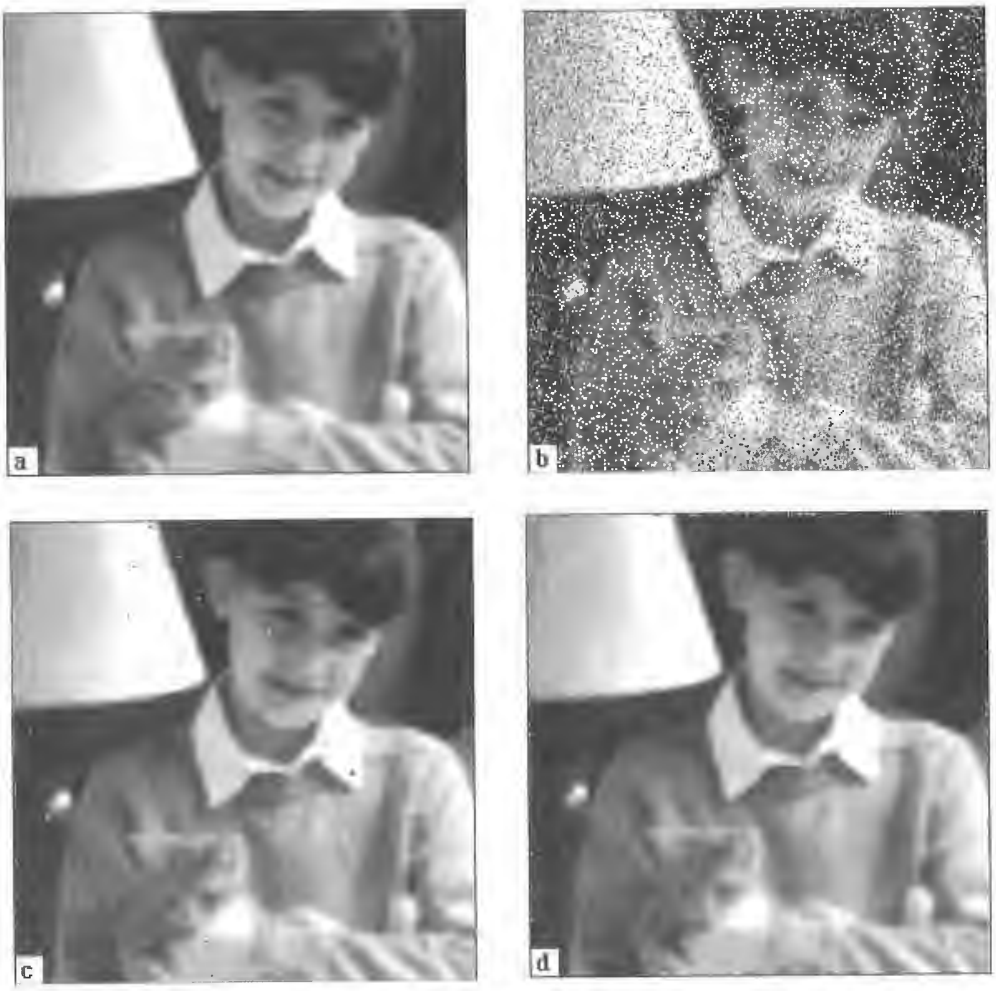


Figure 19. Removal of shot noise with a median filter:
(a) *original image;*
(b) *image* ***a*** *with 10% of the pixels randomly selected and set to black, and another 10% randomly selected and set to white;*
(c) *application of median filtering to image* ***b*** *using a 3 × 3 square region;*
(d) *application of median filtering to image* ***b*** *using a 5 × 5 octagonal region.*

and replaced by values from the surrounding background. This is an essential feature of the median filter, which can sometimes be used to advantage to remove dirt or other small, unwanted features from images. For noise reduction purposes, however, it is better to repeat the application of a small median than to use a large one.

Because of the minimal degradation to edges from median filtering, it is possible to apply the method repeatedly. **Figure 23** shows an example in which a 5 × 5 octagonal median filter was applied 12 times to an image. The fine detail is erased in this process, and large regions take on the same brightness values, although the edges remain in place and well defined. This type of leveling of brightness due to repetition of median filtering is sometimes described as contouring or posterization (but those terms also have other meanings that will be presented elsewhere).

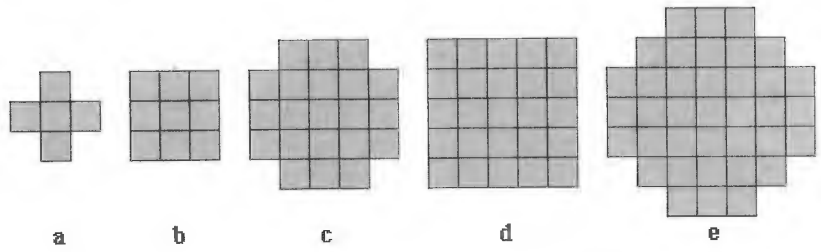


Figure 20. Neighborhood patterns used for median filtering: *(a) four nearest-neighbor cross; (b) 3 × 3 square containing nine pixels; (c) 5 × 5 octagonal region with 21 pixels; (d) 5 × 5 square containing 25 pixels; (e) 7 × 7 octagonal region containing 37 pixels.*

The concept of a median filter requires a ranking order for the pixels in the neighborhood, which for grey scale images is simply provided by the pixel value. Color images present a challenge to this idea (Sartor and Weeks, 2001; Comer and Delp 1999; Heijmans, 1994). Simple application of the ranking operations to the red, green and blue channels is rarely useful and does not generalize to HSI because hue is an angle that wraps around modulo 360°.

A color median filter can be devised by using as the definition of the median value that pixel whose color coordinates give the smallest sum-of-squares of distances to the other pixels in the neighborhood (Astolo et al., 1990; Oistämö and Neuvo, 1990; Russ, 1995b). The choice of the color space in which these coordinate distances are measured can also present a problem. As discussed earlier, HSI space is generally preferred for processing to RGB space, but there is no unique answer to the question of the relative scaling factors of these different spaces, or how to deal with the angular measure of hue values. **Figure 24** shows an example of an HSI color median filter.

The extension of median filtering (or rank-order filtering in general) from a simple ranking of scalar numbers (the grey scale pixel values) to vectors representing color space values opens the

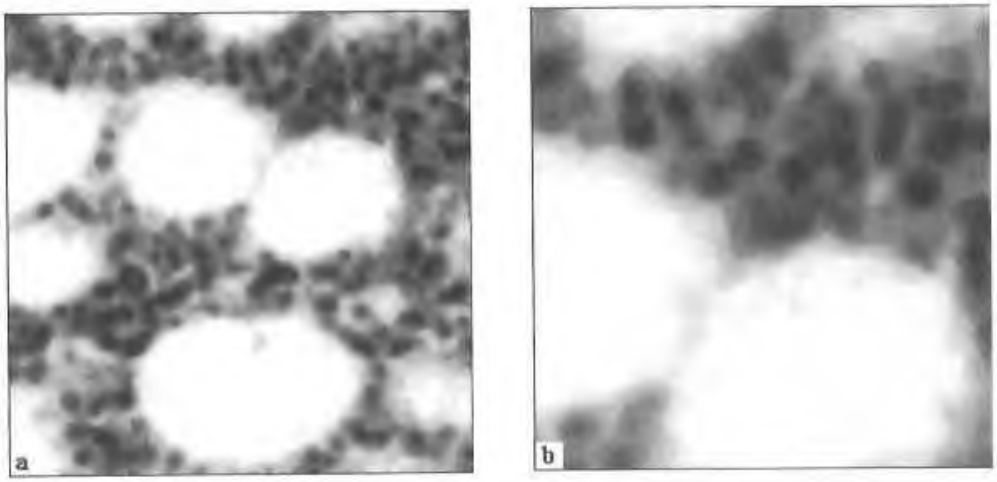
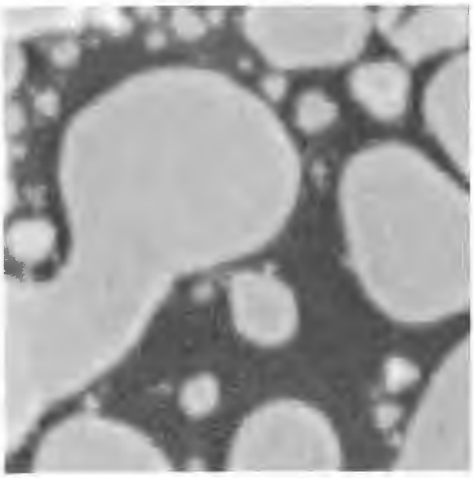
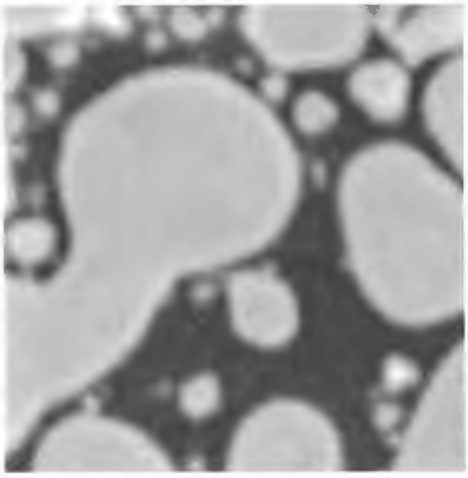


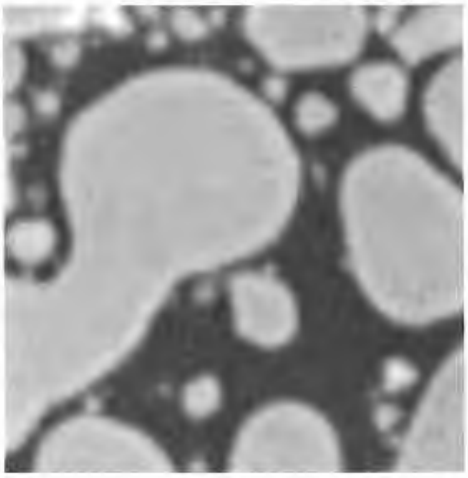
Figure 21. Smoothing with a median filter:
(a) *the same image as in* ***Figure 13*** *after application of a 5 × 5 octagonal median filter;* ***(b)*** *enlargement of image* ***a*** *to show individual pixels.*



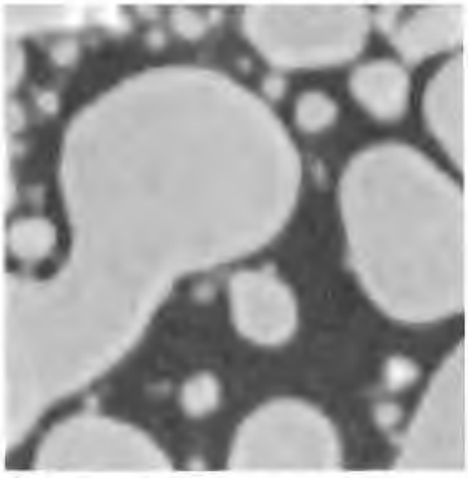
a



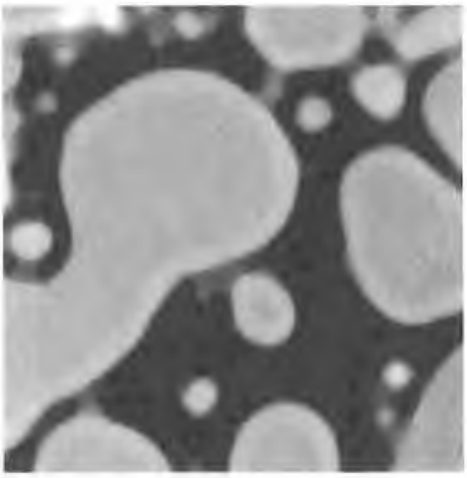
b



c



d



e

Figure 22. Comparison of noise reduction techniques (SEM image enlarged to show pixel detail):
(a) *original;*
(b) *5 × 5 uniform averaging;*
(c) *Gaussian filter, standard deviation = 1 pixel;*
(d) *median filter, 5× 5 octagonal region;*
(e) *median filter, radius = 15 pixels.*



Figure 23. Repeated application of a 5 × 5 octagonal median filter: *(a) original image; (b) after 12 applications — the fine details have been erased and textured regions leveled to a uniform shade of grey, but boundaries have not shifted.*

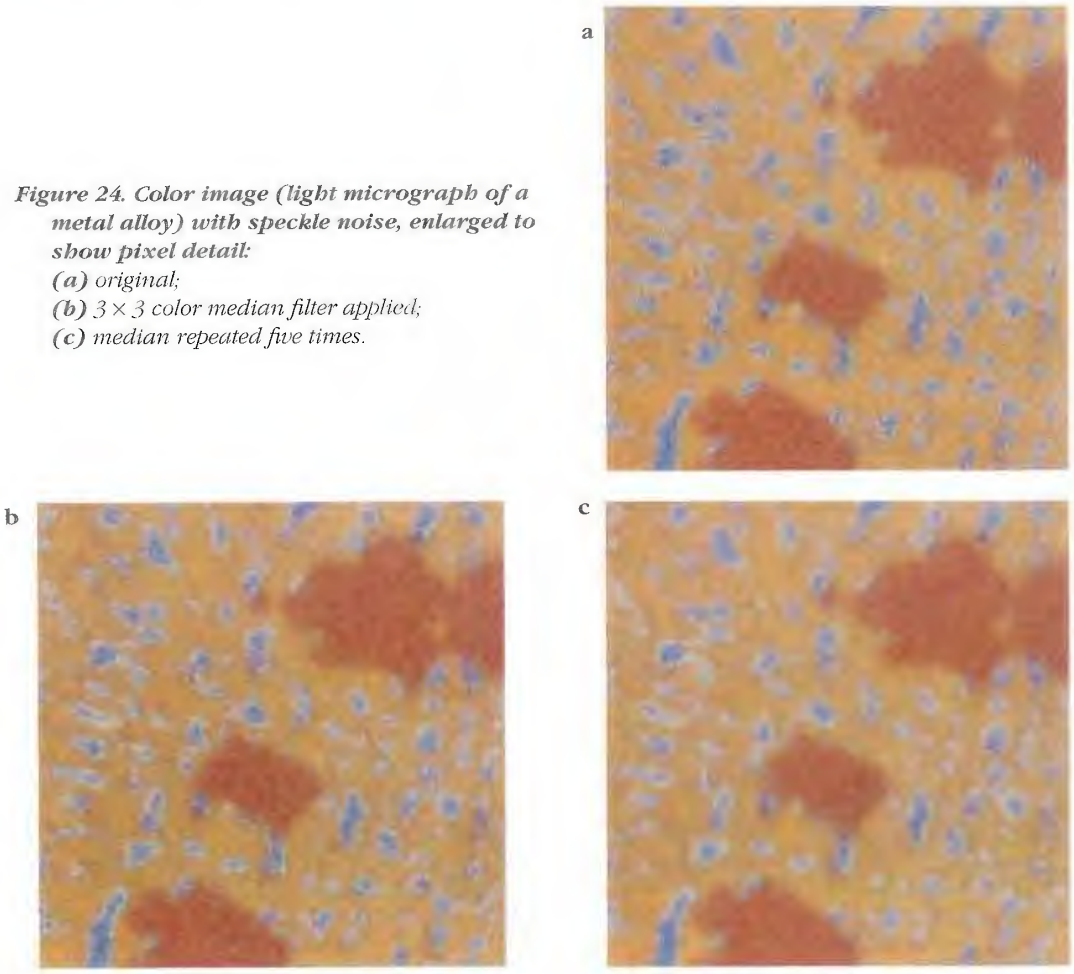


Figure 24. Color image (light micrograph of a metal alloy) with speckle noise, enlarged to show pixel detail:
(a) *original;*
(b) *3 × 3 color median filter applied;*
(c) *median repeated five times.*

door to more complex uses of vector ranking. Instead of the vector length, it is possible to construct rankings based on vector angles to remove vectors (colors) that point in atypical directions, or hybrid methods that combine direction and distance (Smolka et al., 2001). Vectors need not be restricted to three dimensions, so it is possible to combine many criteria into a total measure of pixel similarity. The performance of these methods is bought at the price of computational complexity, but it is possible in some cases to reconstruct images corrupted by up to 70% noise pixels with quite good accuracy.

Sharpening of edges can be accomplished even better with a mode filter (Davies, 1988). The mode of the distribution of brightness values in each neighborhood is, by definition, the most likely value; however, for a small neighborhood, the mode is poorly defined. An approximation to this value can be obtained with a truncated median filter. For any asymmetric distribution, such as would be obtained at most locations near but not precisely straddling an edge, the mode is the highest point, and the median lies closer to the mode than the mean value. This is illustrated in **Figure 25**. The truncated median technique consists of discarding a few values from the neighborhood so that the median value of the remaining pixels is shifted toward the mode. In the example shown in **Figure 26**, this is done for a 3 × 3 neighborhood by skipping the two pixels whose brightness values are most different from the mean, ranking the remaining seven values, and assigning the median to the central pixel.

Another modification to the median filter is used to overcome its tendency to erase lines which are narrower than the half-width of the neighborhood, and to round corners. The so-called hybrid median, or corner-preserving median, is actually a three-step ranking operation (Nieminen et al., 1987). In a 5 × 5 pixel neighborhood, pixels may be ranked in two different groups as shown in **Figure 27**. The median values of the 45-degree neighbors forming an "X" and the 90-degree neighbors forming a "+" (both groups include the central pixel) are compared to the central pixel and the median value of that set is then saved as the new pixel value. As shown in **Figure 28**, this method preserves lines and corners which are erased or rounded off by the conventional median. In a larger neighborhood, more orientations can be employed; four directions can be used in a 5 × 5 region, producing four values that can be ranked along with the original central pixel.

If the hybrid median filter is applied repeatedly, it can also produce posterization. Because the details of lines and corners are preserved by the hybrid median, the shapes of regions are not smoothed as they are with the conventional median, although the brightness values across steps are still sharpened and posterized, as illustrated in **Figure 29**.

The fact that the hybrid median involves multiple ranking operations, first within each of the groups of pixels and then to compare those medians to the central pixel, does not impose a computational penalty. Each of the ranking operations is for a much smaller number of values than used in a square or octagonal region of the same size. For example, the 5-pixel wide neighborhood

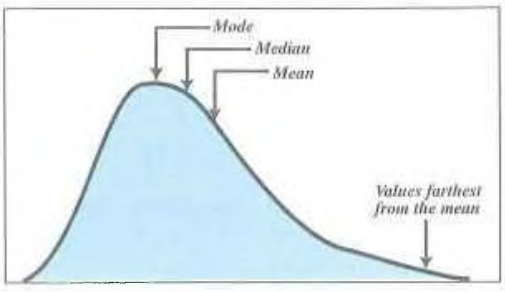


Figure 25. *Schematic diagram of an asymmetric histogram distribution of brightness values, showing the relationship between the mode, median and mean. The truncated median filter works by discarding the values in the distribution which are farthest from the mean, and then using the median of the remainder as an estimate for the mode. For a symmetrical distribution values are discarded from both ends and the median value does not change (it is already equal to the mode).*



Figure 26. Application of the truncated median filter to posterize the image from Figure 23a:
(a) *one application of the 3 × 3 truncated median;*
(b) *difference between* ***Figure 26a*** *and a conventional 3 × 3 median filter, showing the difference in values along edges;*
(c) *12 applications of the truncated median filter.*

used in the examples contains either 25 (in the square neighborhood) or 21 pixels (in the octagonal neighborhood) which must be ranked in the traditional method. In the hybrid method, each of the groups contains only 9 pixels, and the final comparison involves only three or five values. Even with the additional logic and manipulation of values, the hybrid method is about as fast as the conventional median.

Posterizing an image, or reducing the number of grey levels so that regions become uniform in grey value and edges between regions become abrupt, falls more into the category of enhancement than correcting defects, but is included here as a side effect of median filtering. Other methods can produce this effect. One that is related to the median is the extremum filter, which replaces each pixel value with either the minimum or maximum value in the neighborhood, whichever is closer to the mean value. **Figure 30** shows this operator applied to the image from **Figure 19**. This filter is not edge-preserving and may shift boundaries. When the extremum filter is iterated, it is sometimes called a "toggle" filter (Arce et al., 2000).

Sometimes it is useful to construct a neighborhood that is not round, but has a specific shape based on independent knowledge about the nature of the image. A common defect in video images is the appearance of horizontal lines that indicate variations in signal between the two interlaced fields. A similar defect occurs in atomic force microscope images due to DC signal offsets.

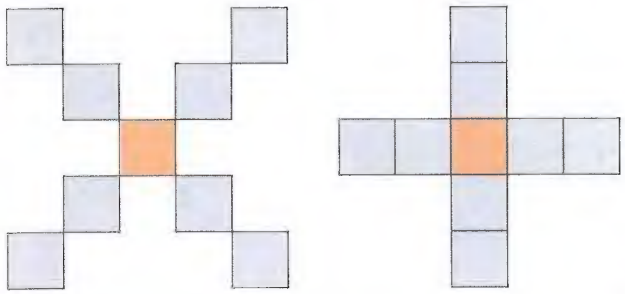


Figure 27. *Diagram of neighborhood pixels used in the hybrid median filter. Both groups include the central pixel (color) and are ranked separately. The median of each group, and the central pixel, are then ranked again to select the overall median value.*

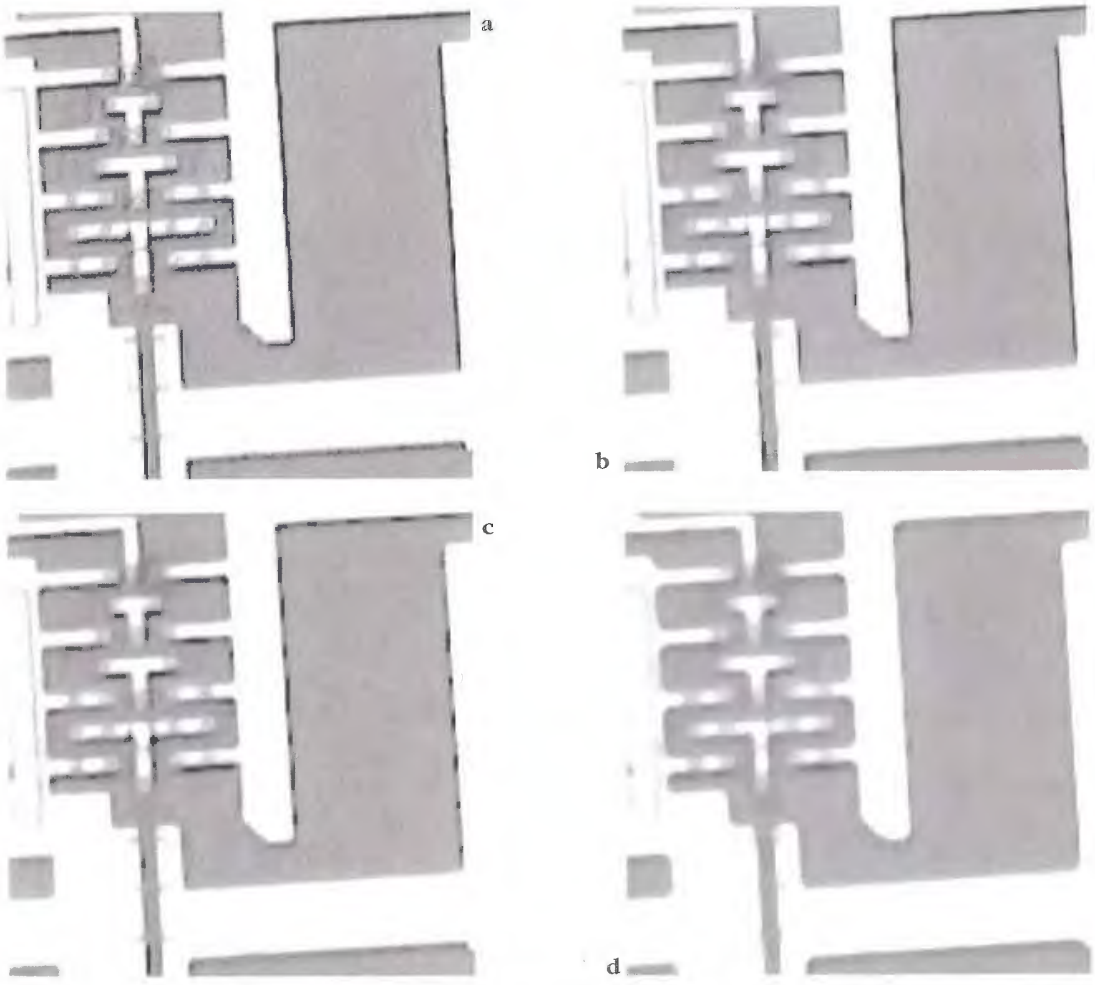
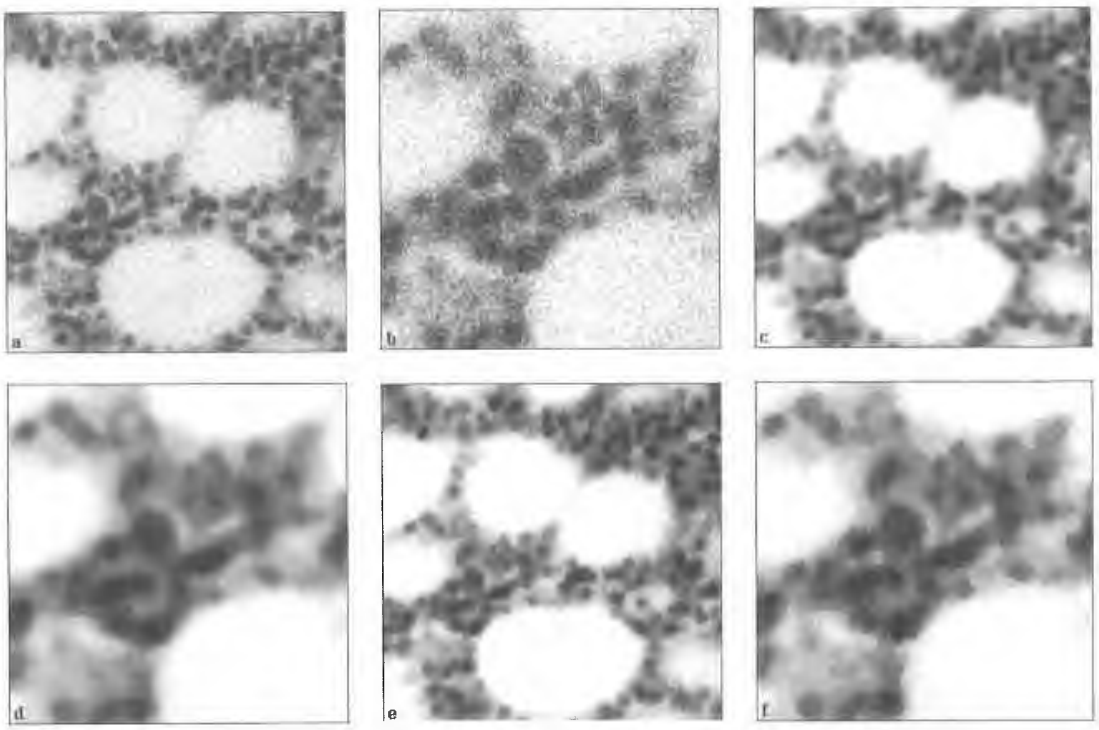


Figure 28. Application of the hybrid median filter to a light microscope image of an integrated circuit, showing the improved retention of lines and corners:
(a) *original image with noise;*
(b) *application of the 5 × 5 hybrid median filter;*
(c) *application of a conventional 3 ×3 median, which does not remove all of the noise but still degrades corners and edges somewhat;*
(d) *application of a conventional 5 × 5 octagonal median filter, showing its greater rounding of corners and elimination of edges.*

Figure 29. Repeated application of the hybrid median filter to the noisy image in Figure 10:
(a) *original image;*
(b) *zoomed portion of* ***Figure 29a*** *showing individual pixels;*
(c) *repeated application of a conventional 5-pixel wide median filter;*
(d) *zoomed portion of* ***Figure 29c*** *showing individual pixels;*
(e) *repeated application of the hybrid 5-pixel wide median filter;*
(f) *zoomed portion of* ***Figure 29e*** *showing individual pixels.*
Notice that the brightness values are posterized and smoothed and edge contrast is sharpened, but the shapes of features and edges are not smoothed.



Movie film often exhibits vertical scratches resulting from wear and tear and rubbing against transport devices. If the direction of the lines or scratches is known, a neighborhood may be constructed to remove it.

Figure 31 shows an example of video line noise in a poor quality surveillance tape. This is a common defect that can arise from dirty recording heads, for example. A median neighborhood that is vertical but only one pixel wide can eliminate much of the line noise. Similarly, **Figure 32** shows horizontal streaks in a low voltage SEM image in which the scan rate was too high for the amplifier time constant, which blurs information along the scan line direction. Again, a median filter applied in a vertical stripe corrects the problem.

In most simple cases the neighborhood used in a ranking operation such as a median filter is circular with a radius selected on the basis of the size of the smallest features to be retained. For computational simplicity, a square neighborhood is sometimes used, which can introduce some directional bias into the results. It is also possible to use an adaptive neighborhood, one that includes

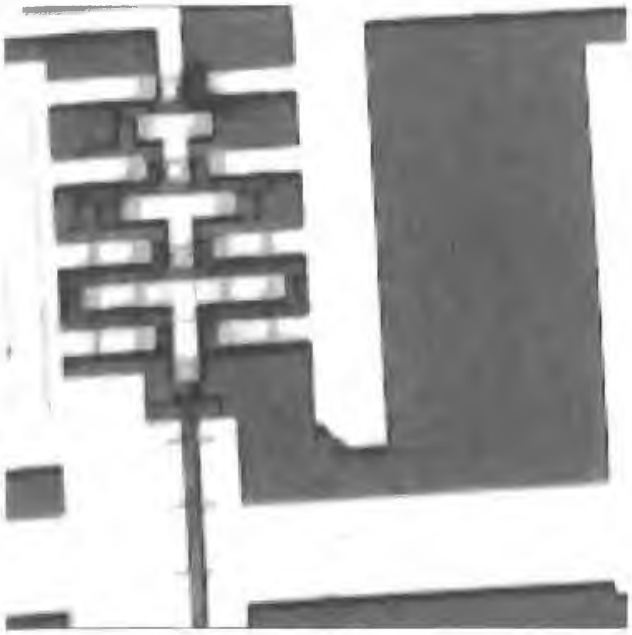
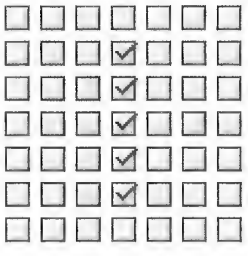


***Figure 30**. Posterization of the image from **Figure 28**, produced by applying a 3 × 3 extremum filter which replaces each pixel value with either the minimum or maximum value in the neighborhood, whichever is closer to the mean value.*



a

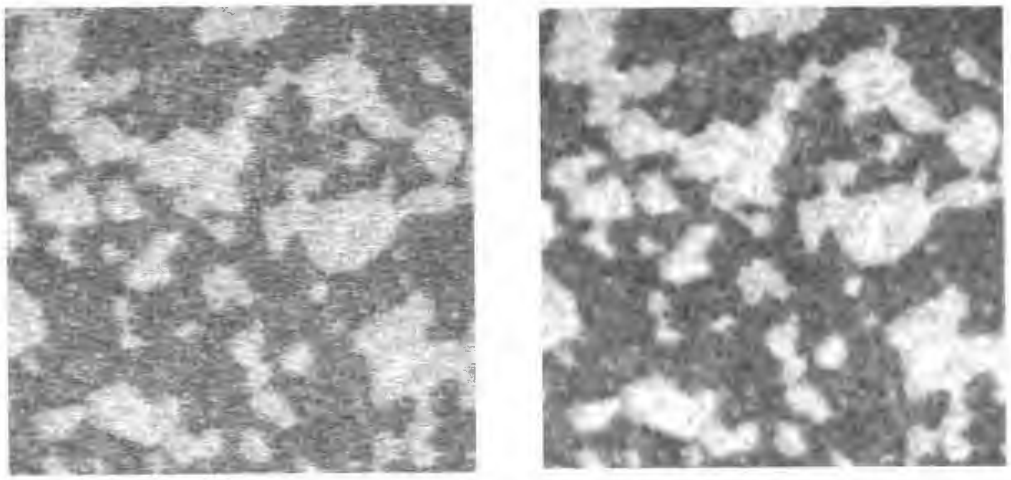


b

***Figure 31**. Surveillance video image with horizontal scan line noise **(a)**, a custom neighborhood to perform median filtering only in a vertical direction **(b)**, and the result **(c)**.*



c



a b

Figure 32. *Low-voltage SEM image of a thin film, with horizontal streaking* ***(a)****, and the result of applying a vertical median filter* ***(b)****.*

pixels based on their value and position. In general, the goal is to select for ranking a set of N (a small odd number) pixel values. There are several possible methods, all of which start with a larger neighborhood of candidate pixels which are then included in the ranking if they meet additional criteria, which may be used in combination:

1. Pixels where the difference from the central pixel is less than some adjustable threshold
2. The N pixels within the larger neighborhood that are closest in value to the central pixel
3. The N pixels within the larger neighborhood that are closest in value to the central pixel and are contiguous with it and with each other (Kober et al., 2001)

It is also possible to weight the pixels according to their distance from the central pixel, by entering nearby pixel values more than once into the list to be ranked. Generally, at the expense of computational complexity, these methods produce good rejection of impulsive noise (as does the conventional median) and improved rejection of additive random noise.

Other neighborhood noise-reduction methods

A modification to the simple averaging of neighborhood values that attempts to achieve some of the advantages of the median filter is the so-called Olympic filter. The name comes from the system of scoring used in some events in the Olympic games, in which the highest and lowest scores are discarded and the remainder averaged. The same thing is done with the pixel values in the neighborhood. By discarding the extreme values, shot noise is rejected. Then the average of the remaining pixel values is used as the new brightness.

Figure 33 shows an application of this method to the image from **Figure 13**, containing Gaussian noise (random intensity variations or speckle). Because it still causes blurring of edges and still requires sorting of the brightness values, this method is generally inferior to the others discussed and is not often used. **Figure 34** shows an application to the shot noise introduced in **Figure 19**. The performance is quite poor: the features are blurred and yet the noise is not all removed.

Other versions of modified or conditional smoothing omit some of the pixel values in the moving neighborhood from the smoothing operation. The justification is always based on the omitted pixels being different from those that are used, so that they presumably belong to a different region.

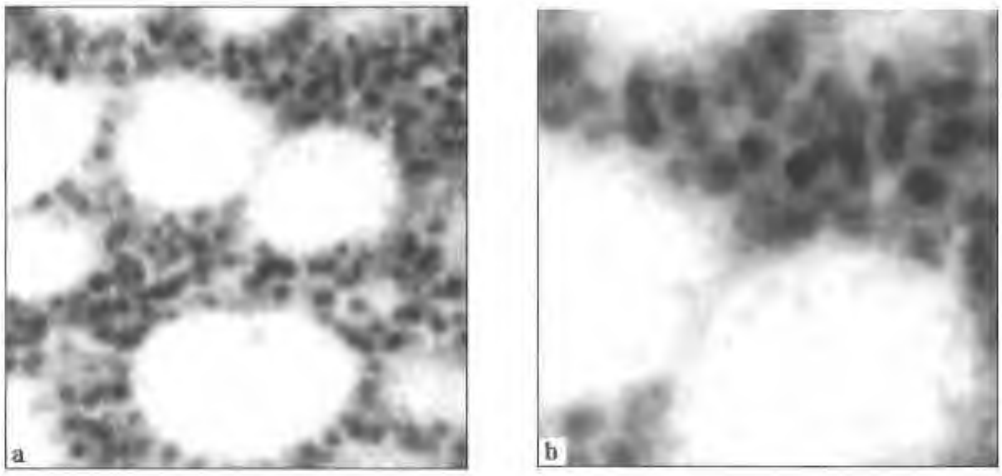


Figure 33. Application of an Olympic filter to Gaussian noise. The four brightest and four darkest pixels in each 5 × 5 neighborhood are ignored and the remaining 17 averaged to produce a new image: (a) *application to the image in* ***Figure 10****;* ***(b)*** *enlargement to show pixel detail.*

Most of the techniques take into account both the location of the pixel in the region and the difference in the value from the original central pixel. For example, fitting a function to the pixels may be used to detect a sharp change in the slope and to omit pixels that lie beyond the edge ("kriging"). **Figure 35** shows one example of such a conditional smoothing process. It does reduce noise, but is not obviously superior to a simple median. All of these methods make some assumptions about the nature of the image and of the noise present, and the better the assumptions fit the actual image, the better the result.



Figure 34. Application of Olympic filter to shot noise. The original image is the same as in Figure 13: (a) *the two brightest and two darkest pixels in each 3 × 3 neighborhood are ignored and the remaining five averaged;* ***(b)*** *the four brightest and four darkest pixels in each 5 × 5 neighborhood are ignored and the remaining 17 averaged.*



Figure 35. Application of conditional smoothing to a noisy image (Figure 19):
(a) *original;*
(b) *conditional smoothing that estimates boundary locations and performs smoothing only using pixels within each region;*
(c) *median filter (5-pixel wide octagonal neighborhood).*

Other, more complicated combinations of operations are used for very specific types of images. For instance, synthetic aperture radar (SAR) images contain speckle noise which varies in a known way with the image brightness. To remove the noise, the brightness of each pixel is compared to the average value of a local neighborhood. If it exceeds it by an amount calculated from the average and the standard deviation, then it is replaced by a weighted average value. Using some coefficients determined by experiment, the method is reported (Nathan and Curlander, 1990) to perform better at improving signal-to-noise than a simple median filter. This is a good example of an ad hoc processing method based on knowledge of the characteristics of the signal and the noise in a particular situation. In general, any filtering method that chooses between several algorithms or modifies its algorithm based on the actual contents of the image or the neighborhood is called an adaptive filter (Mastin, 1985).

Noise is often modeled as a Gaussian additive function, and noise reduction methods are often tested by adding Gaussian noise to an image, but in fact various noise sources have very widely differing characteristics. Noise that arises from photon or particle counting is generally Poisson, which becomes Gaussian for large numbers. The distribution of exposed grains in film is also approximately Gaussian. The effect of electronic circuitry in cameras and amplifiers can be either additive or multiplicative, and generally affects dark areas differently from bright ones. Noise resulting from light scattering in the atmosphere or from surfaces is generally multiplicative rather than additive. Speckle interferometry encountered in radar imaging is more complex, since it interacts through shifts in phase and polarization. It also tends to have a spatial correlation, which makes it more difficult to model. The removal of speckle by acquiring many short duration images is beyond the scope of this text.

Another way of filtering by ranking is to use the maximum and minimum brightness rather than the median. **Figure 36** shows the results of a two-step operation. First, the brightest pixel value in each region (a 5×5 octagonal neighborhood) was used to replace the original pixel values. Then in a second transformation, the darkest pixel value in each region was selected. This type of combined operation requires two full passes through the image, and during each pass only the previous pixel brightness values are used to derive the new ones.

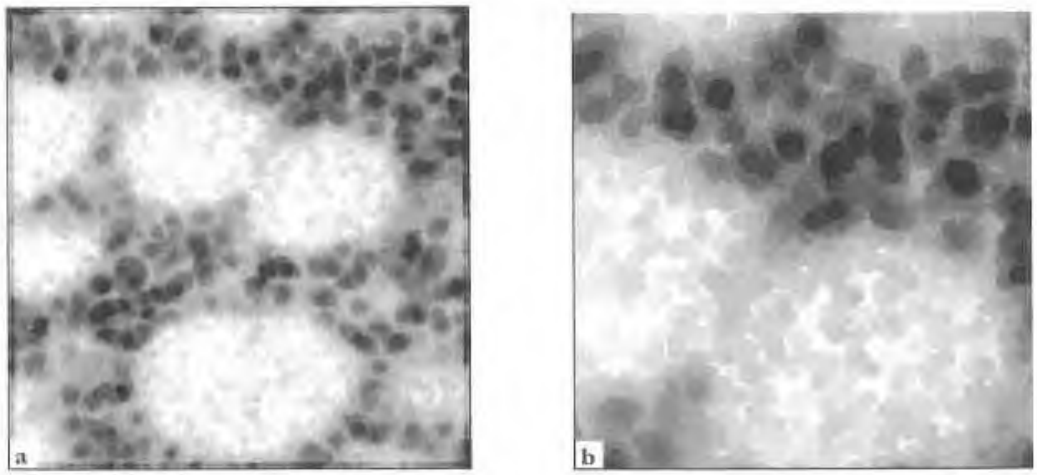


Figure 36. *Grey-scale opening, or erosion and dilation. Two separate ranking operations are performed on the original image from* ***Figure 5****. First each pixel value is replaced with the brightest value in the neighborhood (5 × 5 octagonal); then using this image, each pixel value is replaced by the darkest value in the same size neighborhood:* ***(a)*** *result;* ***(b)*** *enlarged to show pixel detail.*

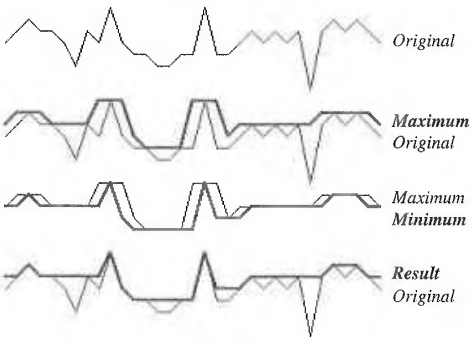


Figure 37. Schematic diagram of the operation of grey-scale erosion and dilation in one dimension, showing (starting from the top): *the original profile with result of first (maximum) pass, producing a new profile through brightest points; the second step in which a new profile passes through the darkest (minimum) points in the result from step 1; comparison of the final result to the original profile, showing rejection of noise and dark spikes but retention of bright features.*

For reasons that will be discussed in more detail in Chapter 7, this type of operation is often described as a grey-scale erosion and dilation, by analogy to the erosion and dilation steps which are performed on binary images. The sequence is also called an opening, again by analogy to operations on binary images. By adjusting the sizes of the neighborhoods (which need not be the same) used in the two separate passes, to locate the brightest and then the darkest values, other processing effects are obtained. These are described in Chapter 4.

As a method for removing noise, this technique would seem to be the antithesis of a median filter, which discards extreme values. It may be helpful to visualize the image as a surface for which the brightness represents an elevation. **Figure 37** shows a one-dimensional representation of such a situation. In the first pass, the brightest values in each region are used to construct a new profile which follows the "tops of the trees." In the second pass, the darkest values in each region are used to bring the profile back down to those points which were large enough to survive the first one, giving a new profile that ignores the dark noise spikes in the original while retaining the bright features. Applying the same operations in the other order (erosion followed by dilation) would have removed bright noise while retaining dark features.

Another method that uses ranking, but in two different size neighborhoods, is useful for locating and removing noise. It is often called a "top hat" or "rolling ball" filter. Imagine the brightness values of the pixels to represent the elevation of a surface. A top hat filter consists of a flat disk that rests on the surface, and a central crown of a smaller diameter, as shown in **Figure 38**. This filter is centered on each pixel in the image, with the brim "resting" on the surface. Any pixels that "stick up" through the crown of the hat are considered to be noise, and replaced. The replacement value may be either the mean or the median value of the pixels covered by the brim of the hat.



Figure 38. *Diagram of a top hat filter. The brim (red) rests on the "surface" that corresponds to the brightness of pixels in the image. Any pixel with a brightness value able to rise through the crown of the hat is detected and can be replaced with the mean or median of the outer neighborhood. The inner and outer radius of the brim and the height of the crown are all adjustable parameters.*

The implementation of this filter uses two neighborhoods, one a round (or approximately round) region corresponding to the inside or crown of the hat, and a second annular neighborhood surrounding it that corresponds to the brim. In each, the maximum (brightest) value is found. If the difference between the brightest pixel in the interior region and the outside exceeds a threshold (the height of the hat) then the pixel value is replaced with the mean or median of the outside region.

An adaptation of this same logic to a trench-shaped neighborhood (a line of pixels forming the inner region, with parallel lines on both sides forming the outer region) can be used to effectively find and remove scratches, provided they have a known direction. This technique is sometimes used to remove scratches when digitizing film.

If the sense of the brightness values is reversed and darkest pixels are the target of the filter, it is usually called a rolling ball filter. The symbolism is that a sphere rolling on the surface will be unable to touch the bottom of the pit represented by the dark noise pixel. **Figure 39** shows an example. The dust particles on the slide are all smaller than 9 pixels across, so a filter consisting of an inner circle with a radius of 4 pixels and an outer one with a radius of 6 pixels forms the top hat. The dust particles are removed, but because the bug itself is too large to fit inside the inner circle, there is no large difference between the darkest values in the two neighborhoods and the pixels are not altered.

In the next chapter, we will consider the use of this same filter for a different purpose. Instead of removing extreme points as noise, the same operation can be used to find points of interest that are smaller than the crown of the hat and brighter or darker than the local neighborhood. In that case, the points that do *not* protrude through the crown are suppressed, and only those that do are kept. **Figure 39c** also illustrates this property of the top hat. Using the same settings for neighborhood size, but keeping the extreme values and suppressing those that do not have a difference greater than the top hat threshold reveals just the dust particles on the slide.

Defect removal, maximum entropy, and maximum likelihood

As shown, the rolling ball/top hat filter can be used to remove small defects by replacing the pixels with a value taken from the surrounding neighborhood. A more general approach to this type of replacement is to interpolate the replacement values. Real images often contain discrete localized defects, either originating in the specimen itself or in the camera (e.g., dirt on the lens). If the defect is large and irregular in shape, smoothing may leave some of the pixel values from the defect remaining in the resulting image, and it may be impractical to use a neighborhood for rank-based filtering that is large enough to encompass it. In many cases, better results can be obtained by filling in the region of the defect using the adjacent values. In the example shown in **Figure 40**, a hole in a leaf has been manually circled and the pixels replaced by linear interpolation between the edge pixels. The algorithm is similar in effect to stretching a membrane across an irregular opening with an edge of variable height. The elastic membrane forms a lowest-energy surface. Calculating the pixel values within the region is accomplished in the same way.

Instead of manual outlining, regions may be selected by any of the thresholding techniques discussed in Chapter 6. These include specifying a range of brightness or color values that discriminate the defects from their surroundings, or using a region-growing method in which selecting one point in the defect allows the software to include all touching similar pixels. In **Figure 41**, this method was used to select the yellow flower. Then the region was enlarged by two pixels (using dilation, a method discussed in Chapter 7), and the background interpolated from the edge pixels of the region. This approach can be quite useful for removing manually selected features from images. Subtracting the modified image from the original produces an image of just the removed feature, as shown.

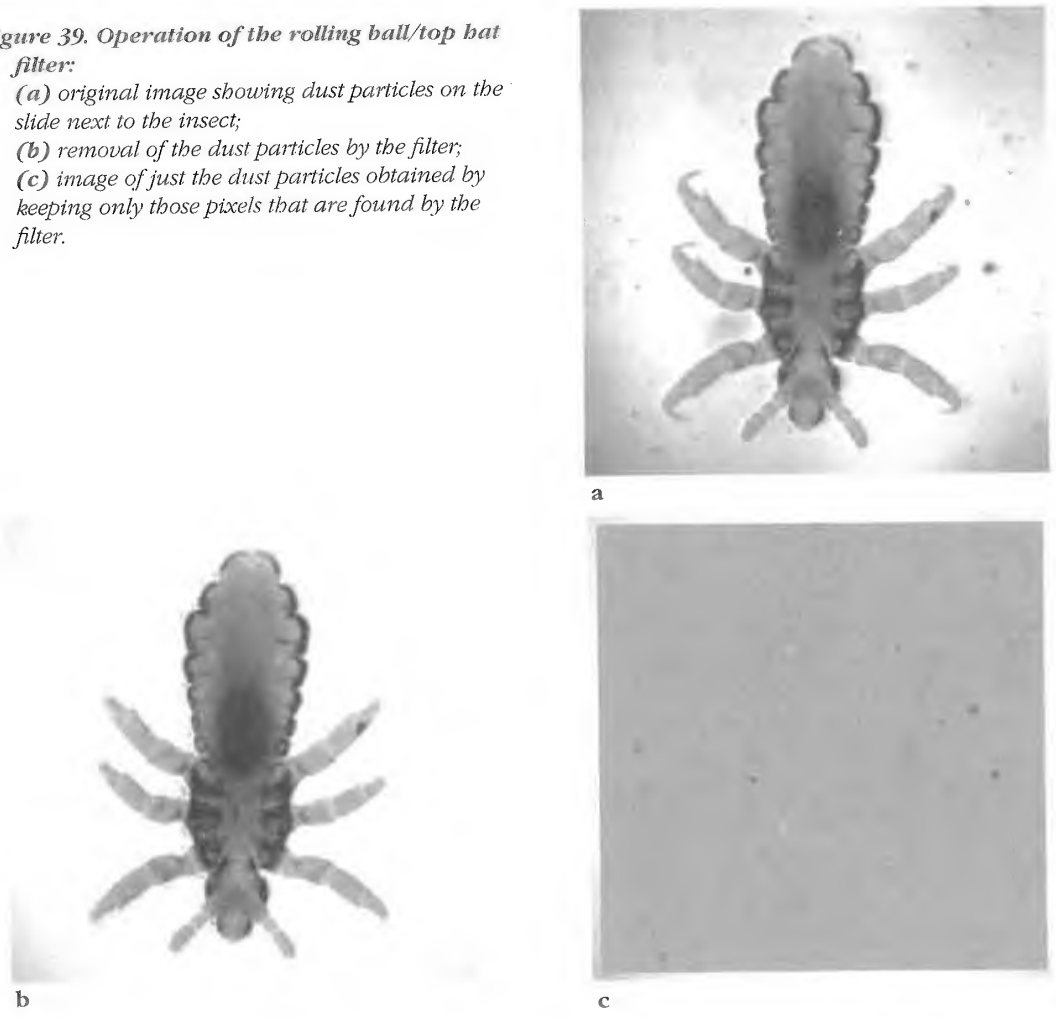


Figure 39. Operation of the rolling ball/top hat filter:
(a) original image showing dust particles on the slide next to the insect;
(b) removal of the dust particles by the filter;
(c) image of just the dust particles obtained by keeping only those pixels that are found by the filter.

Images may contain other defects, such as blur due to motion or out-of-focus optics, in addition to noise. The "inverse filtering" methods described in Chapter 5 under the topic of deconvolution are quite noise sensitive. When the noise that is present can itself be characterized, it may be possible to use a maximum entropy approach to remove the artefacts.

Maximum entropy methods represent a computer-intensive approach to removing artefacts such as noise or blur from images, based on some foreknowledge about the image contents and some assumptions about the nature of the degradation that is to be removed and the image restored (Skilling, 1986; Frieden, 1988). The conventional description of the method is to imagine an image containing N pixels, that has been formed by a total of M photons (where usually $M>>N$). The number of photons in any single pixel (i.e., the pixel's brightness value) is P_i where i is the pixel brightness value.

The measured image has normalized pixel brightness values $p_i=\Sigma P_i/M$ that only approximate the "true" image that would be collected if there were no artefacts. The p_i values are just the image histogram. The method used to approach this ideal image is to alter pixel brightness values to maximize the entropy in the image, subject to some constraints. The justification for this method is given in terms of statistical probability and Bayes' theorem, and will not be derived here. In some cases this method produces dramatic improvements in image quality.

The "entropy" of the brightness pattern is given in a formal sense by calculating the number of ways that the pattern could have been formed by rearrangement, or $S = M!/p_1!\ p_2!, \ldots p_N!$ where ! indicates factorial. For large values of M, this reduces by Stirling's approximation to the more familiar $S = -\sum p_i \log p_i$. This is the same calculation of entropy used in statistical mechanics. In the particular case of taking the log to the base 2, the entropy of the image is the number of bits per pixel needed to represent the image, according to information theory.

The entropy in the image would be maximized in an absolute sense simply by setting the brightness of each pixel to the average brightness. Clearly, this is not the "right" solution. It is the application of constraints that produce usable results. The most common constraint for images containing noise is based on a chi-squared statistic, calculated as $\chi^2 = 1/\sigma^2 \sum (p_i - p_i')^2$. In this expression the p_i values are the original pixel values and the p_i' are the altered brightness values, and σ is the standard deviation of the values. An upper limit can be set on the value of χ^2 allowed in the calculation of a new set of p_i' values to maximize the entropy. A typical (but essentially arbitrary) limit for χ^2 is N, the number of pixels in the array.

This constraint is not the only possible choice. A sum of the absolute value of differences, or some other weighting rule, could also be chosen. This is not quite enough information to produce an optimal image, and so other constraints may be added. One is that the totals of the p_i and p_i' values must be equal. Bryan and Skilling (1980) also require, for instance, that the distribution of the $p_i - p_i'$ values corresponds to the expected noise characteristics of the imaging source (e.g., a Poisson or Gaussian distribution for simple counting statistics). And, of course, we must be careful to include such seemingly "obvious" constraints as non-negativity (no pixel can collect fewer than zero photons). Jaynes (1985) makes the point that there is practically always a significant amount of real knowledge about the image which can be used as constraints, but which is assumed to be so obvious that it is ignored.

An iterative solution for the values of p_i' produces a new image with the desired smoothness and noise characteristics which are often improved from the original image. Other formulations of the maximum entropy approach may compare one iteration of the image to the next, by calculating not the total entropy but the cross entropy, $-\sum p_i \log (p_i/q_i)$ where q_i is the previous image brightness value for the same pixel, or the modeled brightness for a theoretical image. In this formulation, the cross entropy is to be minimized. The basic principle remains the same.

Maximum likelihood reconstruction of an image is another related approach in which the assumption is made that the boundaries between regions should be sharp. In most real images, pixels that straddle any boundary average values from both sides according the exact placement of the edge with respect to the finite size of the pixel. This leads to intermediate brightness or color values that appear to blur the boundary. Reclassifying the pixels to the brightness of one region or the other sharpens the boundaries.

One of the most common statistical approaches to the classification measures the variance of pixels in many subregions of a neighborhood around each pixel as shown in **Figure 42**. Whichever region has the lowest variance is taken to represent the region to which the central pixel should belong, and it is assigned the mean value from that subregion (Kuwahara et al., 1976). **Figure 43** shows an application of the method.

When applied to halftoned images that have been digitized with a scanner, this filter can eliminate the halftone patterns as shown in **Figure 44**. Also note that for real objects such as the flowers and leaves shown, in which the edges of objects are naturally shaded, the filter converts the appearance to that of a painting or cartoon.

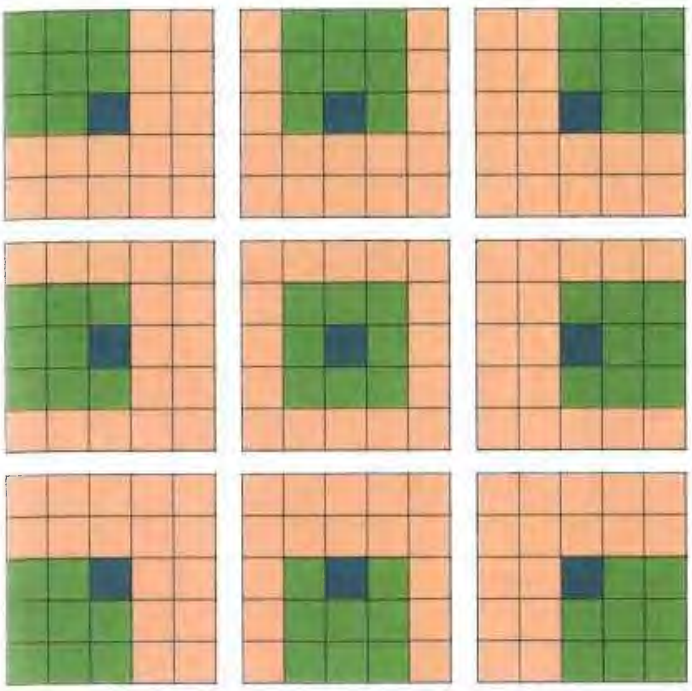


Figure 42. *Nine 3 × 3 subregions (green) within a 5 × 5 neighborhood. The Kuwahara filter assigns the mean value of whichever region has the smallest variance to the central pixel (dark).*

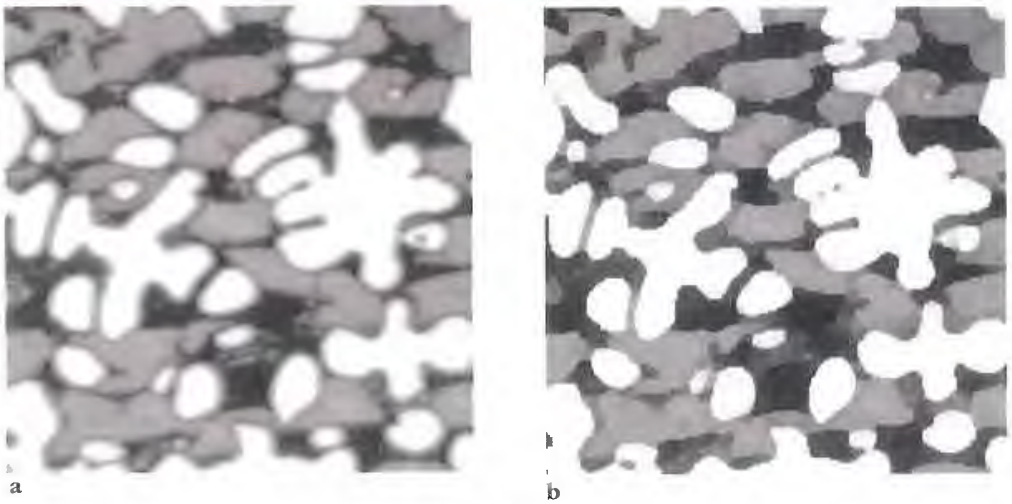


Figure 43. Application of a maximum likelihood filter: *(a) original image (enlarged to show pixel detail); (b) filter applied, resulting in sharp region boundaries.*

Nonuniform illumination

The most straightforward strategy for image analysis uses the brightness of regions in the image as a means of identification: it is assumed that the same type of feature will have the same brightness (or color, in a color image) wherever it appears in the field of view. If this brightness is different from that of other features, or can be made so by appropriate image processing as discussed in Chapter 4, then it can be used to discriminate the features for counting,



a b

Figure 44. Application of the maximum likelihood filter to a scanned image with halftone: *(a) original; (b) filter applied.*

measurement or identification. Even if there are a few other types of objects that cannot be distinguished on the basis of brightness or color alone, subsequent measurements may suffice to select the ones of interest.

This approach is not without pitfalls, which are discussed further in Chapter 7 in conjunction with converting grey scale images to binary (black-and-white) images. Other approaches are available such as region growing or split-and-merge that do not have such stringent requirements for uniformity of illumination. Simple brightness thresholding, when it can be used, is by far the simplest and fastest method to isolate features in an image, and it is important to consider the problems of shading of images.

When irregular surfaces are viewed, the amount of light scattered to the viewer or camera from each region is a function of the orientation of the surface with respect to the source of light and the viewer, even if the surface material and finish is uniform. In fact, this principle can be used to estimate the surface orientation, using a technique known as shape-from-shading. Human vision seems to apply these rules very easily and rapidly because we are not generally confused by images of real-world objects. We will not pursue those methods here, however.

Most of the images we really want to process are essentially two-dimensional. Whether they come from light or electron microscopes, or satellite images, the variation in surface elevation is usually small compared to the lateral dimensions, giving rise to what is often called "two-and-one-half D." This is not always the case of course (consider a woven fabric examined in the SEM), but we will treat such problems as exceptions to the general rule and recognize that more elaborate processing may be needed.

Even surfaces of low relief need not be flat, a simple example being the curvature of the earth as viewed from a weather satellite. This will produce a shading across the field of view, as will illumination of a macroscopic or microscopic surface from one side. Even elaborate collections of lights, or ring lights, can only approximate uniform illumination of the scene.

For transmission imaging, the uniformity of the light source with a condenser lens system can be made quite good, but it is easy for these systems to get out of perfect alignment and produce

shading as well. Finally, it was mentioned in Chapter 1 that lenses or the cameras themselves may cause vignetting, in which the corners of the image are darker than the center because the light is partially absorbed.

Most of these defects can be minimized by careful setup of the imaging conditions, or if they cannot be eliminated altogether, can be assumed to be constant over some period of time. This assumption allows correction by image processing. In many instances, it is possible to acquire a "background" image in which a uniform reference surface or specimen is inserted in place of the actual samples to be viewed, and the light intensity recorded. This image can then be used to "level" the subsequent images. The process is often called "background subtraction" but, in many cases, this is a misnomer. If the image acquisition device is logarithmic with a gamma of 1.0, then subtraction of the background image point-by-point from each acquired image is correct. If the camera or sensor is linear, then the correct procedure is to divide the acquired image by the background. For other sensor response functions, there is no simple correct arithmetic method, and the calibrated response must first be determined and applied to convert the measured signal to a linear or logarithmic space.

In the process of subtracting (or dividing) one image by another, some of the dynamic range of the original data will be lost. The greater the variation in background brightness, the less the remaining variation from that level can be recorded in the image and will be left after the leveling process. This loss and the inevitable increase in statistical noise that results from subtracting one signal from another argue that all practical steps should be taken first to make illumination uniform before acquiring the images, before resorting to processing methods.

Figure 83 in Chapter 1 showed an example of leveling in which the background illumination function could be acquired separately. This acquisition is most commonly done by removing the specimen from the light path, for instance a slide, and storing an image representing the variation. This image can then be used for leveling, although in many cases, it is not practical to remove the specimen, or its presence is a contributor to the brightness variation. This includes situations in which the specimen thickness varies and so the overall absorption of light is affected. Another case is that in which the surface being examined is not perfectly flat, which causes incident light or SEM images to show a varying background.

Figure 45 shows an example of the latter type. The SEM image shows particles on a substrate, which because of the geometry of the surface and of the SEM chamber causes a portion of the image to appear darker than the rest. The human eye is quite tolerant of this kind of gradual brightness variation, and so it is sometimes helpful to apply pseudo-color lookup tables to reveal the shading present in the image. Moving the specimen or changing the magnification alters the pattern of light and dark, in which case it is necessary to perform the correction using the image itself. Fortunately, in many of these situations the variation of background brightness is a smooth and well behaved function of location and can be approximated by simple functions such as polynomials. The leveling in the figure was performed using this method.

Figure 46 shows a situation in which the brightness variation can be extracted directly from the original image. By separating the red, green, and blue planes from the image it is seen that the green image contains the important detail: the blood vessels in the retina of the eye, which are difficult to view because of the overall brightness variation. The red image has nearly the same variation so the ratio of the green to red levels the varying background and makes the vessels more uniform in contrast.

Another situation in which direct measurement of two images allows dividing to correct for nonuniform brightness arises in the transmission electron microscope. It is common, particularly for inorganic samples in materials science, for the specimens to vary in thickness. Computing the ratio of the conventional brightfield image to the zero-loss electrons isolated by a spectrometer attached to the microscope gives a direct measure of the variation in specimen thickness, which can subsequently be used to quantify concentration and structural measurements.

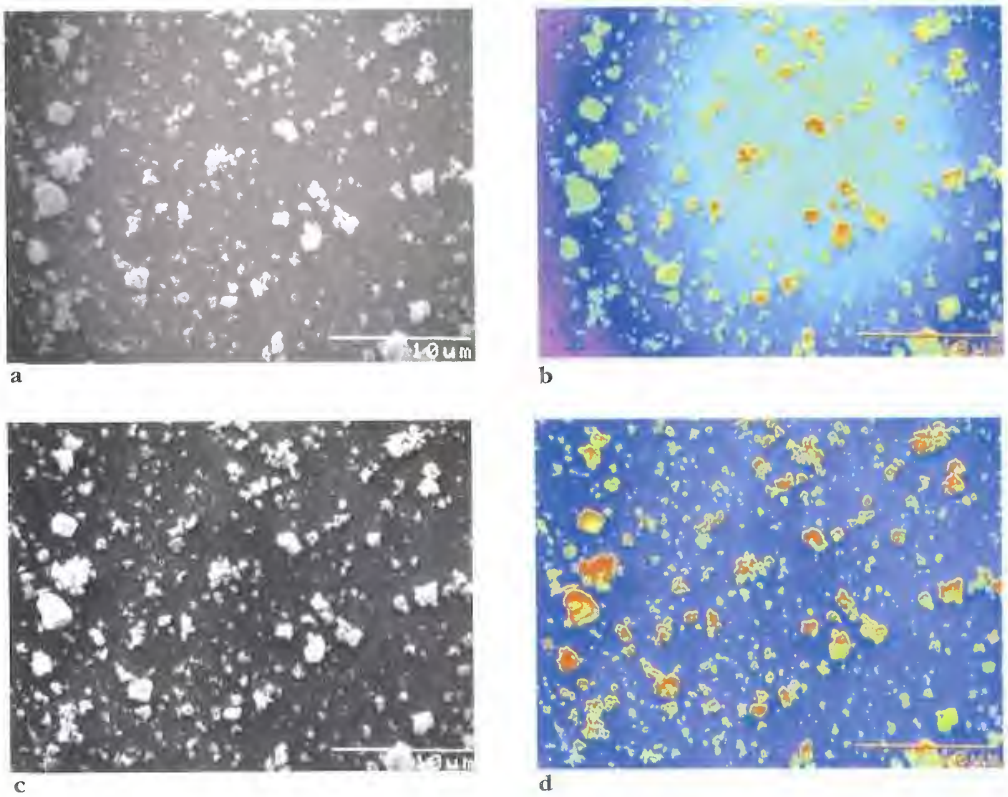


Figure 45. Leveling image contrast:
(a) *SEM image of particulates on a substrate;*
(b) *pseudo-color applied to a to make the shading more visually evident;*
(c) *image leveled by polynomial fitting;*
(d) *pseudo-color applied to* ***c****.*

Fitting a background function

By selecting a number of points in the image, a list of brightness values and locations can be acquired. These can then used to perform least-squares fitting of a function B(*x,y*) that approximates the background, and can be subtracted (or divided) just as a physically acquired background image would be. When the user marks these points, for instance by using a pointing device such as a mouse, trackball or light pen, it is important to select locations that should all have the same brightness and are well distributed across the image. Locating many points in one small region and few or none in other parts of the image requires the function to extrapolate the polynomial, and can introduce significant errors. For a second-order polynomial, the functional form of the fitted background is

$$B(x,y) = a0 + a_1 \cdot x + a_2 \cdot y + a_3 \cdot x^2 + a_4 \cdot y^2 + a_5 \cdot xy \qquad (2)$$

This polynomial has six fitted constants, and so in principle could be fitted with only that number of marked points. Likewise, a third-order polynomial would have ten coefficients; however, in order to get a good fit and diminish sensitivity to minor fluctuations in individual pixels, and to have

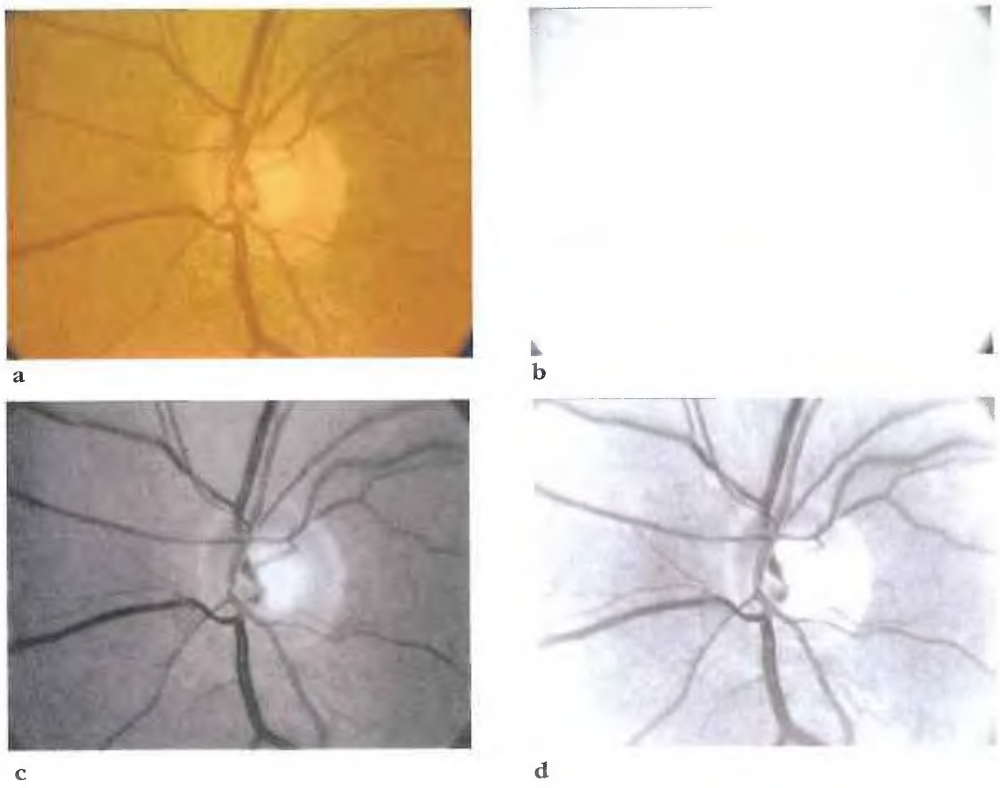


Figure 46. Leveling with a measured background:
(a) *retina image from an opthalmoscope, showing shading that makes the capillaries difficult to discern;*
(b) *red channel contains no detail but does show the shading;*
(c) *green channel has most of the contrast in the image;*
(d) *ratio of green to red shows more uniform contrast (original image courtesy George Mansoor, University of Connecticut Health Care).*

enough points to sample the entire image area properly, it is usual to require several times this minimum number of points.

Figure 47 shows an example in which a background region can be selected in this way. Points around the periphery of the stamp should all be the same in brightness but vary because of nonuniform illumination. Selecting that region, fitting a polynomial function to it, and subtracting it (because this is a digitized photograph, subtraction is appropriate), levels the resulting image of the background and also the stamp. Notice that one effect of leveling is to make the peaks in the image histogram much narrower, since all of the pixels in the same region have more nearly the same value. Comparison of this result with **Figure 41c** shows that the edge interpolation method is somewhat inferior (note the color shift in the flower) because it provides only linear interpolation and uses fewer pixel values to define the background brightness.

In some cases, it is practical to automatically locate the points for background fitting. Automatic leveling is easiest when there is a distinct structure or phase present that is well distributed throughout the image area and contains the darkest (or lightest) pixels present. In that case, the image can be subdivided into a grid of smaller squares or rectangles, the darkest (or lightest) pixels in each subregion located, and these points used for the fitting.

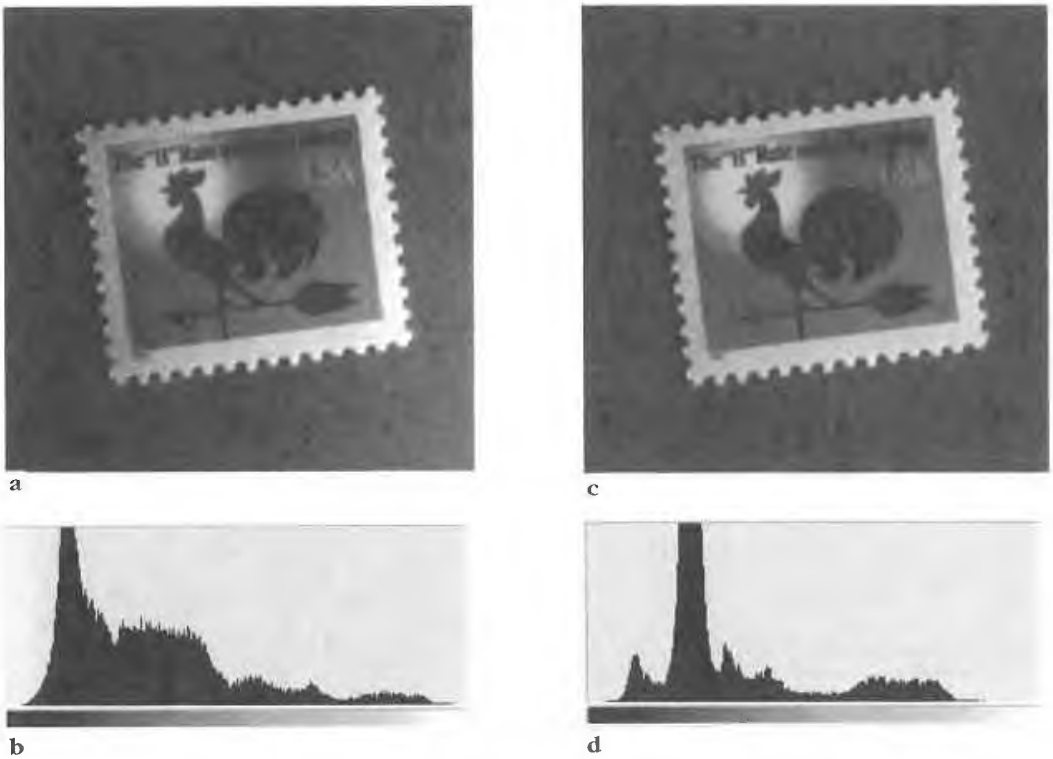
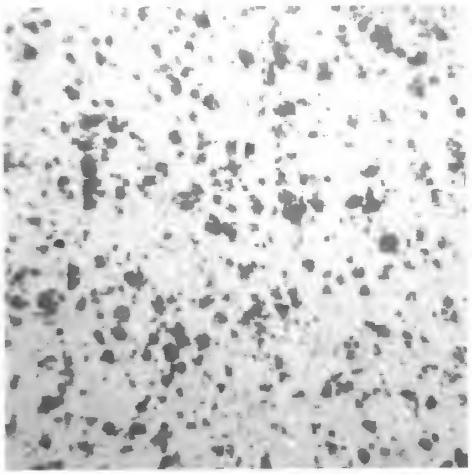


Figure 47. Correction of nonuniform lighting by selecting a background region:
(a) *original image;*
(b) *histogram of* ***a****;*
(c) *result of subtracting a polynomial fit to the pixel values in the periphery of the stamp;*
(d) *histogram of* ***c****.*

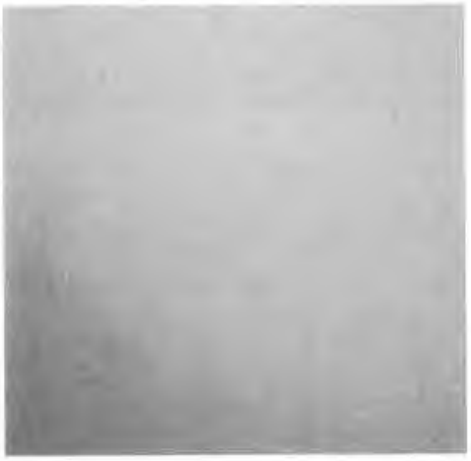
Figure 48 shows an example in which the specimen (a polished ceramic) has an overall variation in brightness due to curvature of the surface. The brightest pixels in each region of the specimen represent the matrix, and so should all be the same. Subdividing the image into a 9 × 9 grid and locating the brightest pixel in each square gives a total of 81 points, which are then used to calculate a second-order polynomial (six coefficients) by least-squares. The fitting routine in this case reported a fitting error (rms value) of less than two brightness values out of the total 0–255 range for pixels in the image.

Figure 48b shows the calculated brightness using the B(x,y) function, and **Figure 48c** shows the result after subtracting the background from the original, pixel by pixel, to level the image. This leveling removes the variation in background brightness and permits setting brightness thresholds to delineate the pores for measurement, as discussed in Chapter 6.

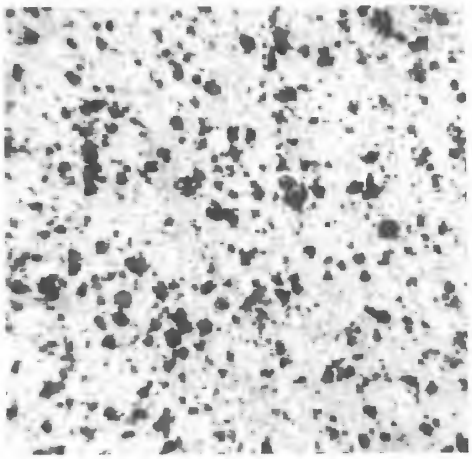
This approach to automatic leveling can of course be applied to either a light or a dark background. By simultaneously applying it to both the lightest and darkest pixels in each region of the image, it is possible to stretch the contrast of the image as a function of location, as shown schematically in **Figure 49** using a line profile of brightness as an example. This autocontrast method works particularly well when the image loses contrast due to nonuniform illumination or varying sample thickness. **Figure 50** shows an example.



a



b



c

Figure 48. Automatic leveling of nonuniform illumination:
(a) *Reflection light microscope image of ceramic specimen, with nonuniform background brightness due to a non-planar surface;*
(b) *background function calculated as a polynomial fit to the brightest point in each of 81 squares (a 9 × 9 grid);*
(c) *leveled image after subtracting* ***b*** *from* ***a****.*

Another approach sometimes used to remove gradual variation in overall brightness employs the frequency transforms discussed in Chapter 5. It assumes that the background variation in the image is a low-frequency signal, and can be separated in frequency space from the higher frequencies that define the features present. If this assumption is justified, and the frequencies corresponding to the background can be identified, then they can be removed by a simple filter in the frequency space representation.

Figure 51 shows an example of this approach. The brightness variation in the original image is due to off-centered illumination in the microscope. Transforming the image into frequency space with a 2D FFT (as discussed in Chapter 5), reducing the magnitude of the first four frequency components by filtering the frequency space image, and retransforming, produces the image shown in the figure.

This method is not entirely successful for this image. The edges of the image show significant variations present because the frequency transform attempts to match the left and right edges and the top and bottom edges. In addition, the brightness of dark and light regions throughout the image which have the same appearance and would be expected to properly have the same brightness show considerable local variations because the brightness variation is a function of the local details, including the actual brightness values and the shapes of the features. There are few practical situations in which leveling can be satisfactorily performed by this method.

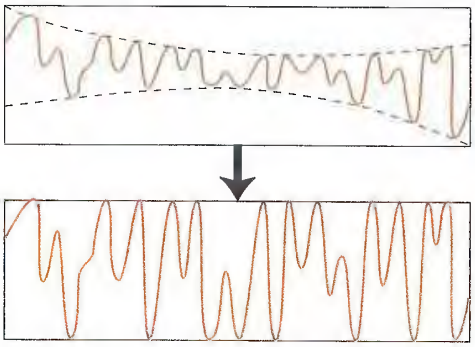


Figure 49. *Schematic diagram of automatic contrast adjustment by fitting polynomials to both the brightest and darkest values across an image and stretching the brightness values between those limits.*

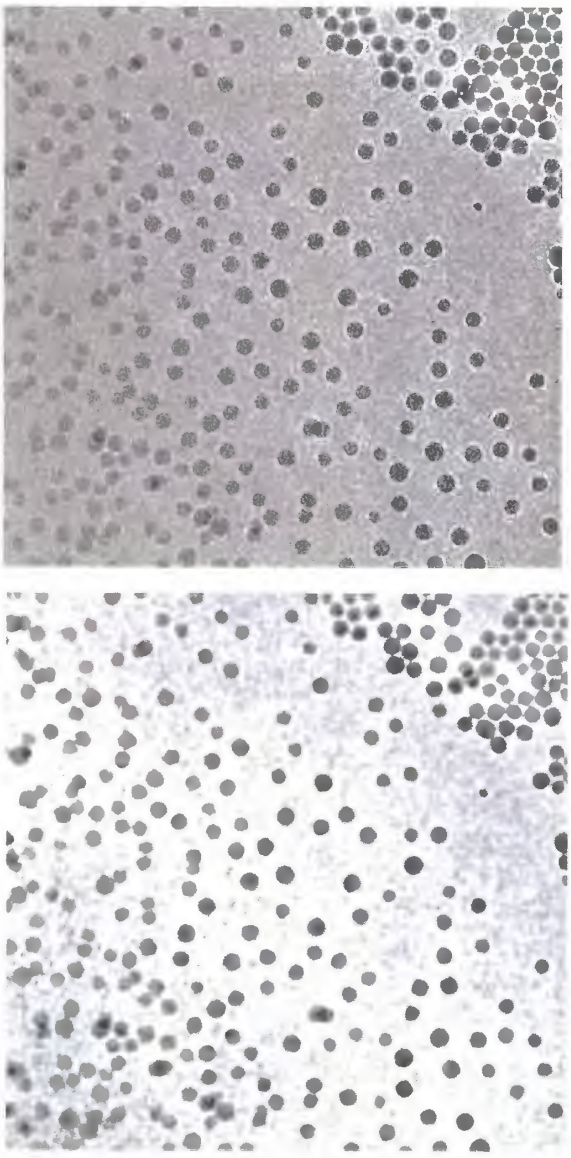


Figure 50. TEM image of latex particles: ***(a)*** *original, with varying contrast due to changing sample thickness;* ***(b)*** *after application of automatic contrast by fitting polynomial functions to light and dark pixel values (and applying a median filter to reduce random pixel noise).*

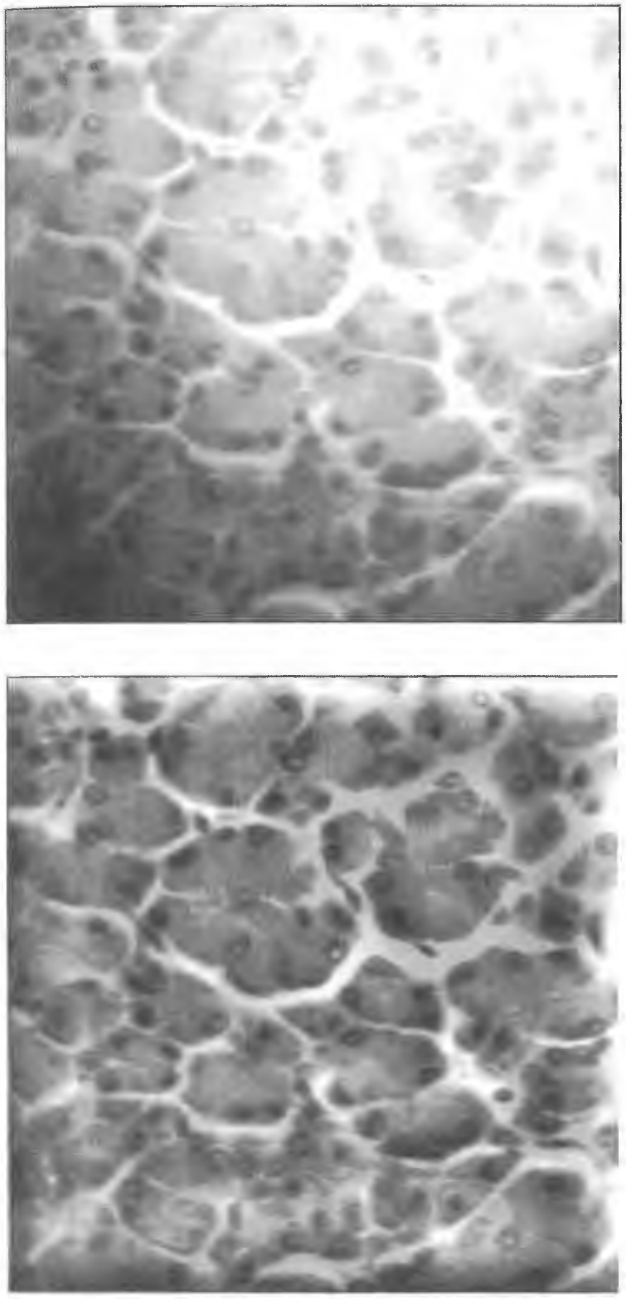


Figure 51. Leveling of image brightness and contrast by removal of low-frequency terms in 2D FFT: *(a) original image, showing nonuniform illumination; (b) attempt to level the brightness by reducing the magnitude to zero of the lowest four frequency components. Notice that in addition to the variations near the edge, the brightness of similar structures is not constant throughout the image.*

Rank leveling

When the background varies more abruptly than can be fit to simple functions, another approach can be used. This method is especially useful when the surface is irregular, such as details on a fracture surface examined in the SEM, or a thin section with a fold viewed in the TEM. The assumption behind this method is that features of interest are limited in size and smaller than the scale of background variations, and that the background is everywhere either lighter than or darker than the features. Both requirements are often met in practical situations.

Rank neighborhood operations (such as the median filter) were discussed previously in this chapter, and are also used in Chapter 4. The basic idea behind neighborhood ranking operations is to compare each pixel to its neighbors or combine the pixel values in some small region, minimally the eight touching pixels in a 3×3 square. This operation is performed for each pixel in the image, and a new image is produced as a result. In many practical implementations, the new image replaces the original image with only a temporary requirement for additional storage; these implementation considerations are discussed in Chapter 4.

For our present purposes, the neighborhood comparison works as follows: for each pixel, examine the pixels in a 3×3 square, a 5×5 octagon, or other similar small region. If the background is known to be darker than the features, find the darkest pixel in each neighborhood and replace the value of the original pixel with that darker brightness value. For the case of a background lighter than the features, the brightest pixel in the neighborhood is used instead. These operations are sometimes called grey-scale erosion and dilation, by analogy to the morphological processing discussed in Chapter 7. The result of applying this operation to the entire image is to shrink the features by the radius of the neighborhood region, and to extend the local background brightness values into the area previously covered by features.

Figure 52 illustrates this procedure for an image of rice grains on a dark and uneven background. A neighborhood is used here which consists of 21 pixels in an octagonal 5×5 pattern centered on each pixel in the image. The darkest pixel value in that region replaces the original central pixel. This operation is repeated for every pixel in the image, always using the original image pixels and not the new ones from application of the procedure to other pixels. After this procedure is complete, the rice grains are reduced in size as shown in **Figure 52b**. Repeating the operation continues to shrink the grains and to extend the background based on the local background brightness.

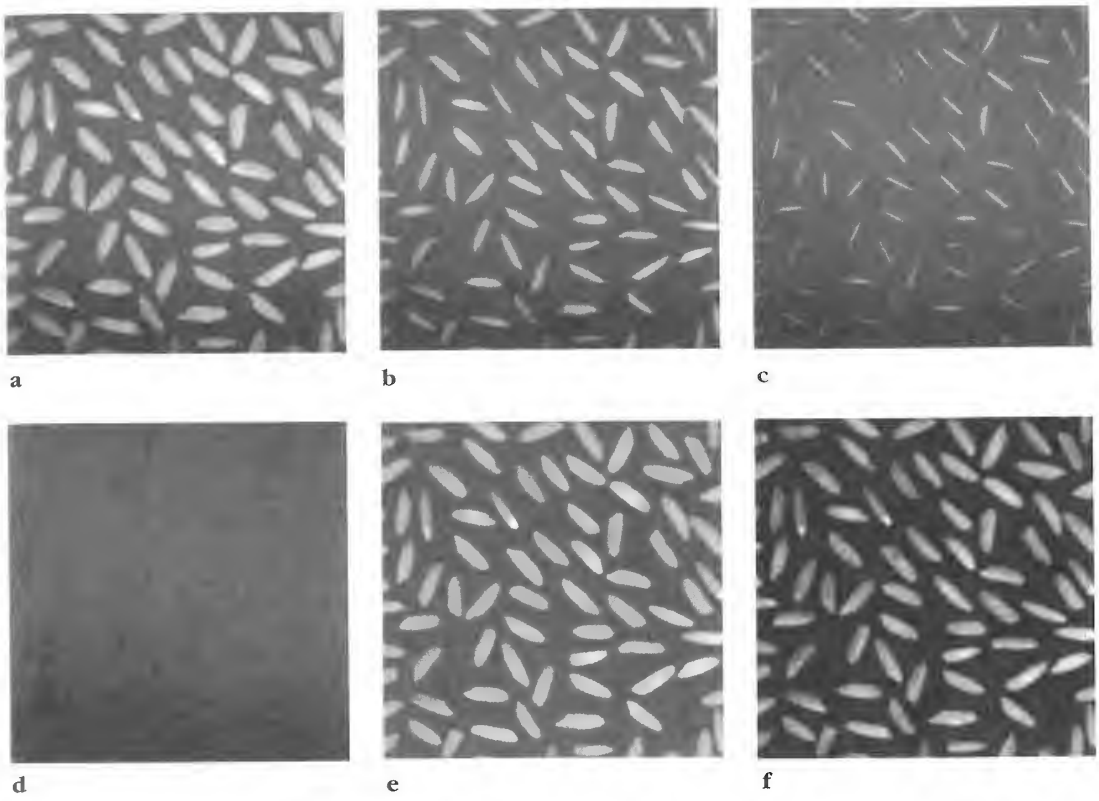
After four repetitions (**Figure 52d**), the rice grains have been removed. This removal is possible because the maximum width of any grain is not larger than four times the width of the 5-pixel-wide neighborhood used for the ranking. Knowing how many times to apply this operation depends upon knowing the width (smallest dimension) of the largest features present, or simply watching the progress of the operation and repeating until the features are removed. In some cases, this can be judged from the disappearance of a peak from the image histogram. The background produced by this method has the large-scale variation present in the original image, and subtracting it produces a leveled image (**Figure 52f**) that clearly defines the features and allows them to be separated from the background by thresholding.

This method is particularly suitable for the quite irregular background brightness variations that occur when looking at details on fracture surfaces. **Figure 53** shows two SEM images in which local ridges and variations produce bright lines on a dark but irregular background. The background in each case was estimated by applying of the ranking operation keeping the darkest pixel in a 7-pixel-wide octagonal neighborhood, followed by smoothing with a Gaussian kernel with a standard deviation of 1.0 pixels. Subtracting this background from the original makes the markings, fatigue striations in one image and brittle quasi-cleavage marks in the other, more visible by removing the overall variation due to surface orientation.

This method is also useful for examination of particles and other surface decorations on freeze-fractured cell walls in biological specimens, examination of surface roughness on irregular particles or pollen, and other similar problems. In some cases it can be used to enhance the visibility of dislocations in transmission electron microscope (TEM) images of materials, which appear as dark lines in different grains whose overall brightness varies due to lattice orientation.

Figure 54 shows an example in which both leveling techniques are used to complement each other. First a ranked background is created by replacing the dark axons with the locally lighter

Figure 52. Constructing a background image with a rank operation:
(a) *an image of rice grains with nonuniform illumination;*
(b) *each pixel replaced with the darkest neighboring pixel in an octagonal 5 ×5 neighborhood;*
(c) *another repetition of the "darkest neighbor" or grey-scale erosion operation;*
(d) *after four repetitions, only the background remains;*
(e) *result of subtracting* ***d*** *from* ***a****;*
(f) *the leveled result with contrast expanded.*



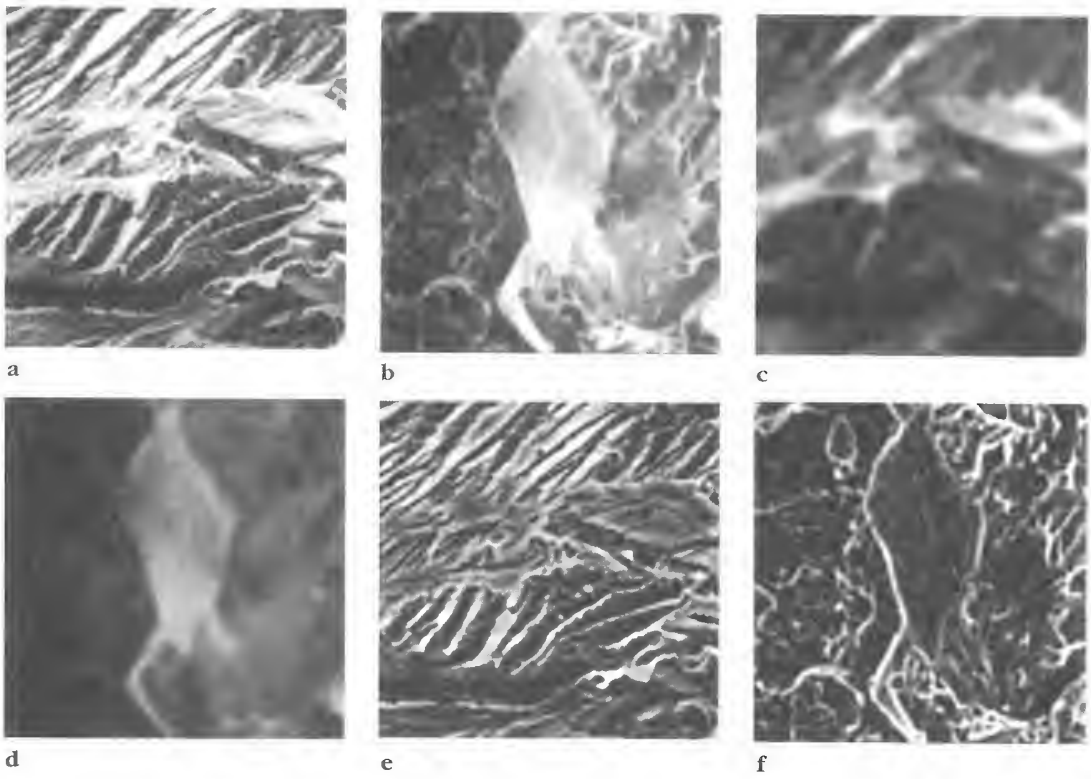
background which is divided into the original image. Then a polynomial fit to the brightest and darkest pixel values is used to locally expand the contrast, which helps to reveal even the small dark axon lines.

The ability to level brightness variations by subtracting a background image, whether obtained by measurement, mathematical fitting, or image processing, is not a cost-free process. Subtraction uses up part of the dynamic range, or grey scale, of the image. **Figure 55** shows an example. The original image has a shading variation that can be fit rather well by a quadratic function, but this has a range of about half of the total 256 grey levels. After the function has been subtracted, the leveled image does not have enough remaining brightness range to show detail in the dark areas of some features. This clipping may interfere with further analysis of the image.

Color shading

Color images present significant problems for shading correction. A typical example of a situation in which this arises is aerial or satellite imagery, in which the irregularity of the ground or the

Figure 53. Application of rank leveling to SEM images:
(a, b) *original images of metal fractures showing local brightness variations due to surface orientation;*
(c, d) *background image produced as described in the text;*
(e, f) *leveled result after subtraction of the background from the original.*



curvature of the planet surface produces an overall shading. In some cases, this affects only the intensity in the image and leaves the color information unaffected; but, depending on the camera response, possible texturing or specular reflection from the surface, atmospheric absorption, and other details of the imaging, it is more common to find that there are also color shifts between the areas in the image that have different illumination.

The same methods used above for grey scale images can be applied to the intensity plane from a color image. In some instances, the same leveling strategies can be applied to the hue plane as shown in **Figure 56**. This is slightly more complicated than leveling brightness since the hue values "warp around" modulo 360°; the simplest solution is to fit the polynomial twice, once with the origin at red and once at cyan, and use whichever gives the better fit. Leveling is practically never useful for application directly to the red, green and blue planes. When (mis)used in this way the operations produce color shifts in pixels that alter the image so that it can not be successfully thresholded, and in most cases does not even "look" right.

Chapter 1 showed an example of another approach. Although the various color planes have each been altered by the effects of geometry and other factors, to a first approximation, the effect is the same across the spectrum of colors. In that case, it is appropriate to use ratios of one color plane to another to level the effects of nonuniform surface orientation or illumination. Filtering the color image in different wavelengths and then dividing one by another cancels out some of the nonuniformity and produces a leveled image in which similar features located in different areas have the same final appearance.

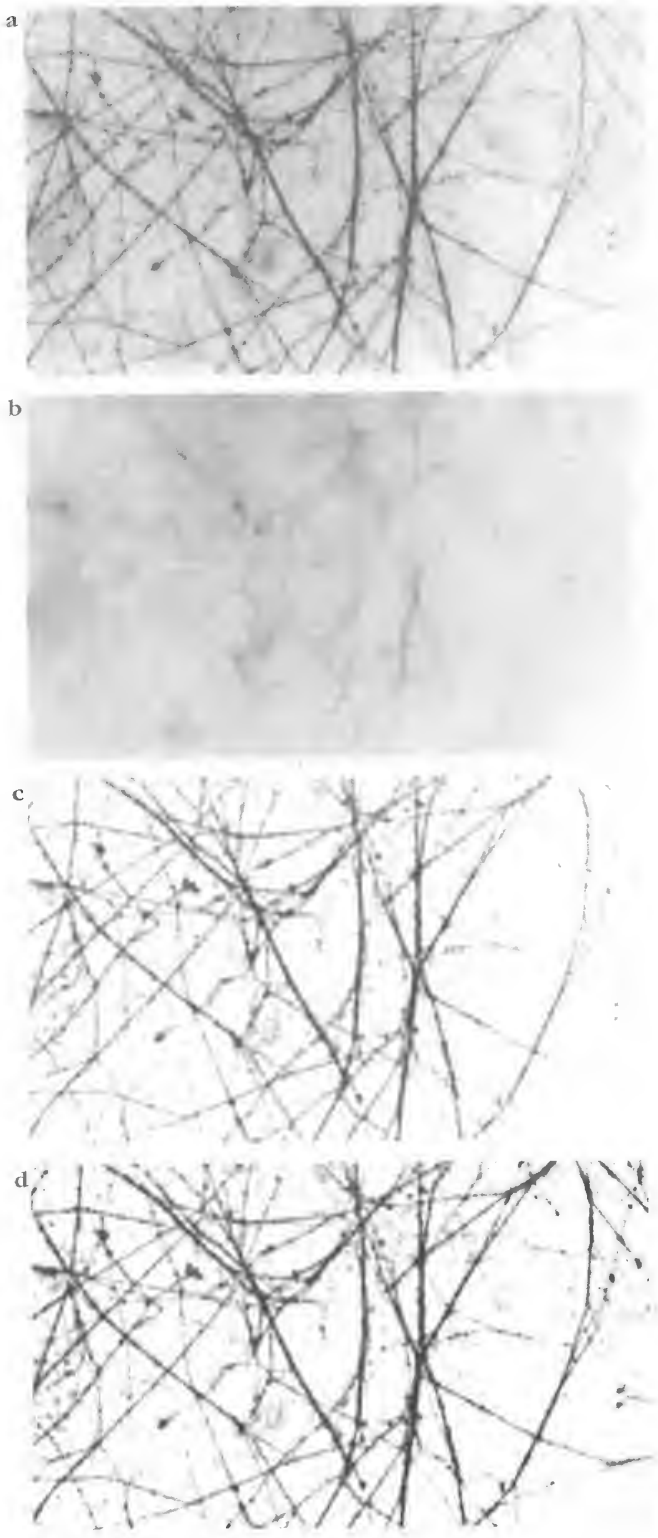


Figure 54. *Combined rank and polynomial methods:*
(a) *original TEM image of axons;*
(b) *background produced by replacing dark pixels with their brightest neighbors within a 5-pixel radius;*
(c) *result of dividing* ***a*** *by* ***b****;*
(d) *contrast expanded between automatically fit polynomial limits.*

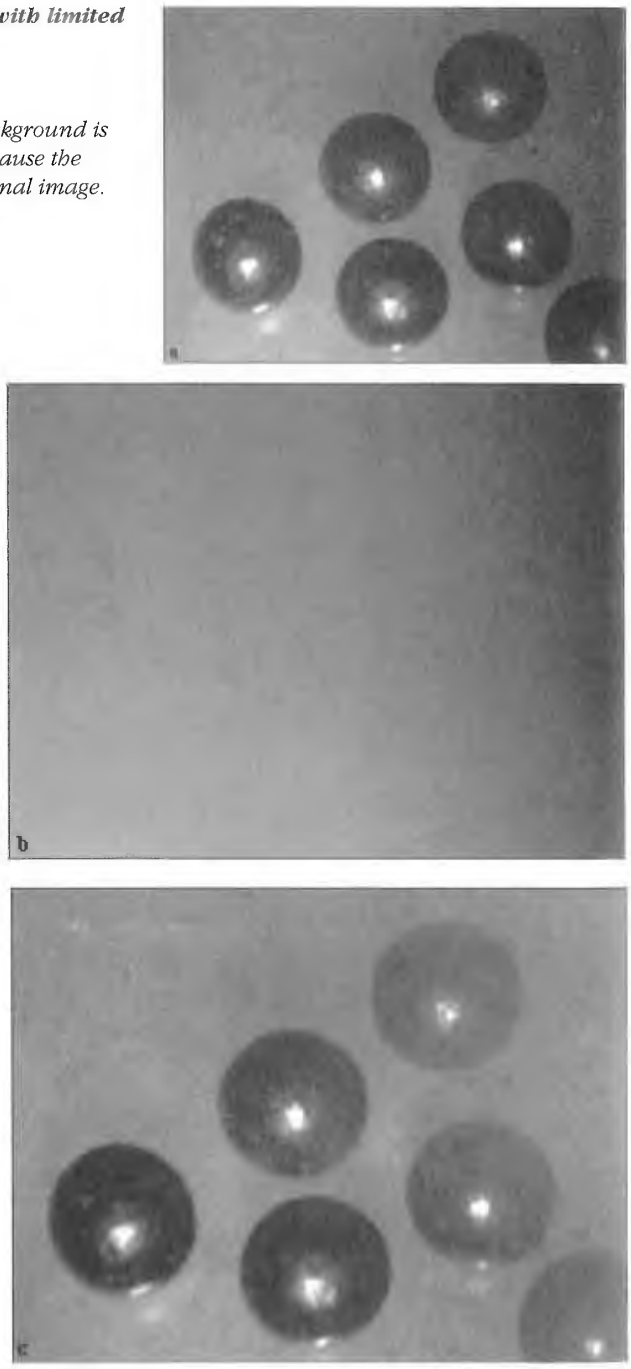
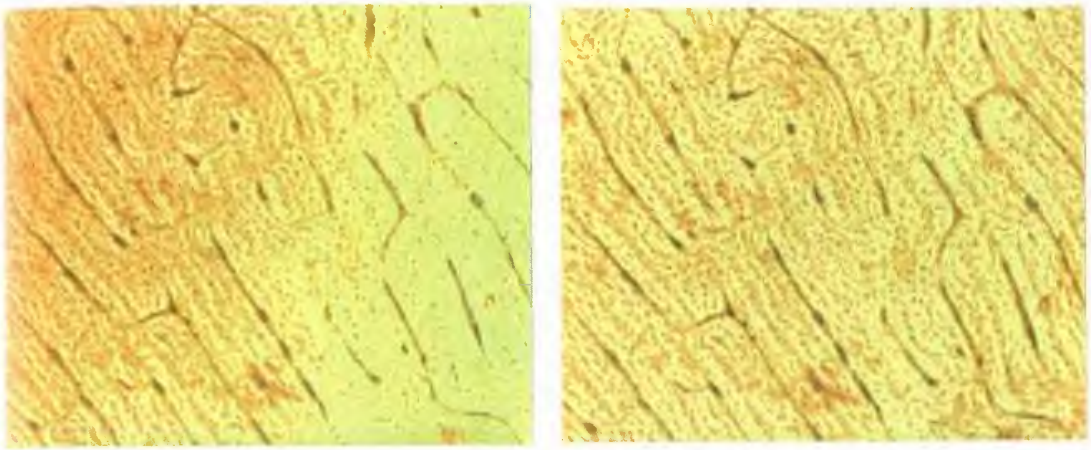


Figure 55. Effect of leveling on an image with limited grey-scale range:
(a) *original image;*
(b) *fitted polynomial background;*
(c) *subtracting image* ***b*** *from* ***a*** *— the background is uniform, but the dark features are not because the original pixels were fully black in the original image.*

Figure 57 shows an example using a satellite image of the entire earth. The limb darkening around the edges of the globe is due primarily to viewing angle, and secondarily to the effects of atmospheric absorption. Histogram equalization (discussed in Chapter 4) of the individual color planes increases the contrast between features and improves the visibility of structures, but the resulting colors are not "real" and do not have any intrinsic meaning or use. Notice in particular the pink colors that appear in clouds, and the green in the oceans and along cloud edges. Separating the image into



a b

Figure 56. Leveling hue: ***(a)*** *light micrograph of bone, showing nonuniform color due to misalignment of the bulb in the illuminator;* ***(b)*** *after polynomial leveling of the hue values.*

separate color planes and ratioing them reveals fine details and levels the overall contrast range. It does not produce an image of the globe in which pixels near the limb have their colors "corrected" to be similar to those in the center of the field of view.

The rank-based leveling method can also be applied to color images. In the example of **Figure 58**, a red dye has been injected to reveal blood vessels but the background varies, making the smaller vessels and those at the periphery difficult to detect. Removing the blood vessels by replacing each pixel with the color values from its neighbor with the least red (color erosion) followed by the complementary operation in which each pixel is replaced by its neighbor with the greatest red value (color dilation) produces a background. Subtracting this background image from the original produces optimal contrast for the blood vessels.

Color correction of images is a very rich and complicated field. Generally, it requires detailed knowledge of the light source and the camera response, which must be obtained with calibration standards. Accurate colorimetry also requires extreme stability of all components. Colorimetry goes well beyond the capabilities of general-purpose image processing and analysis systems, and is not considered in detail here; however, there is a more routine interest in making some adjustments to color images to permit comparisons between regions or images, or to provide some control over consistency in the printing of color images.

These adjustments are generally accomplished with standardization, at least of a relative kind. For instance, most printers (as discussed in Chapter 2) cannot reproduce the full gamut of colors displayed on the screen or captured by a camera. In order to calibrate the relationship between the display and the input/output devices, some systems create a test pattern that is printed on the system printer, redigitized from the printed output, and compared to the original. This comparison allows the system to construct an internal correction matrix to adjust the colors so that the display and reproduction of colors will be more consistent. It also allows flagging those colors that cannot be printed accurately with a given printer.

Correction for a shift in colors, due for instance to a known change in lighting conditions, is also possible. Acquiring an image from a test card under each of several conditions allows the computer to build a matrix of corrections that may be expressed internally as either RGB or HSI components. These are multiplied by the incoming signal to approximately correct for the change in lighting.

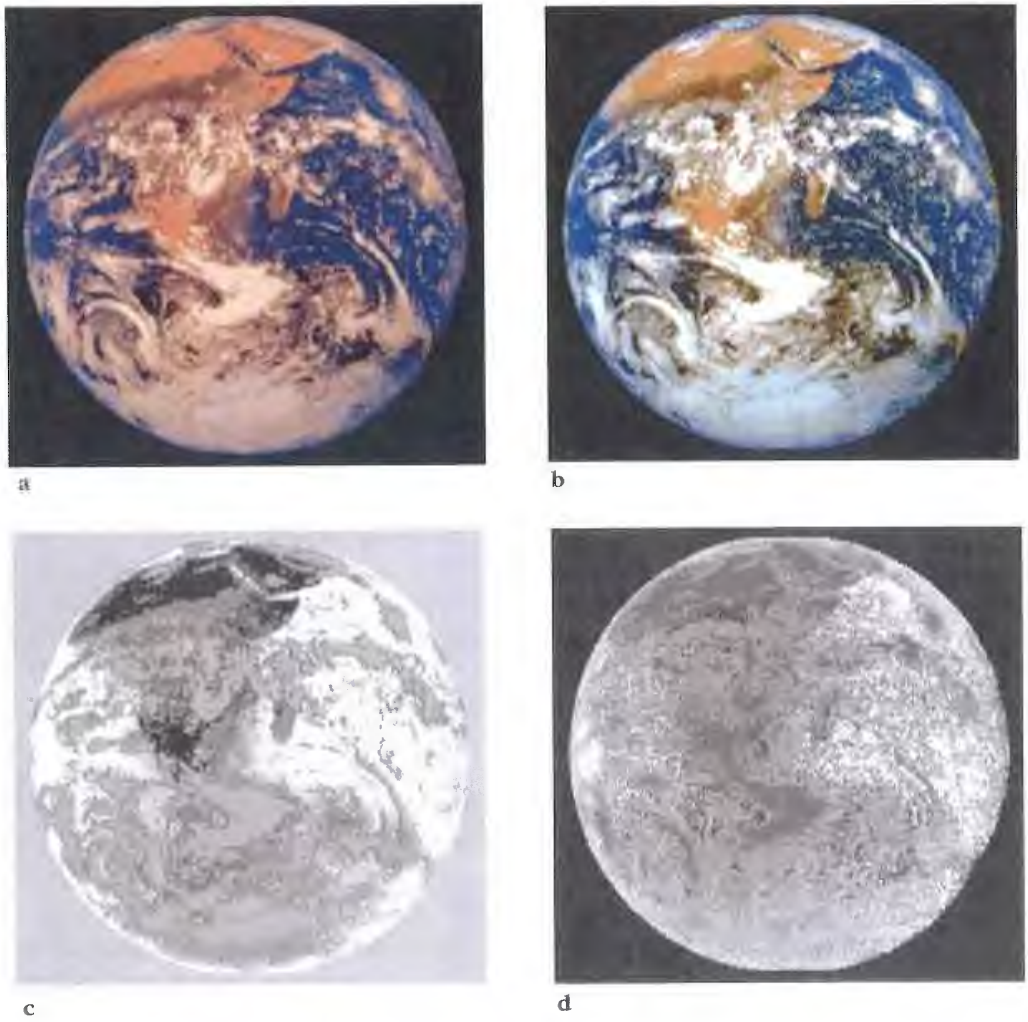


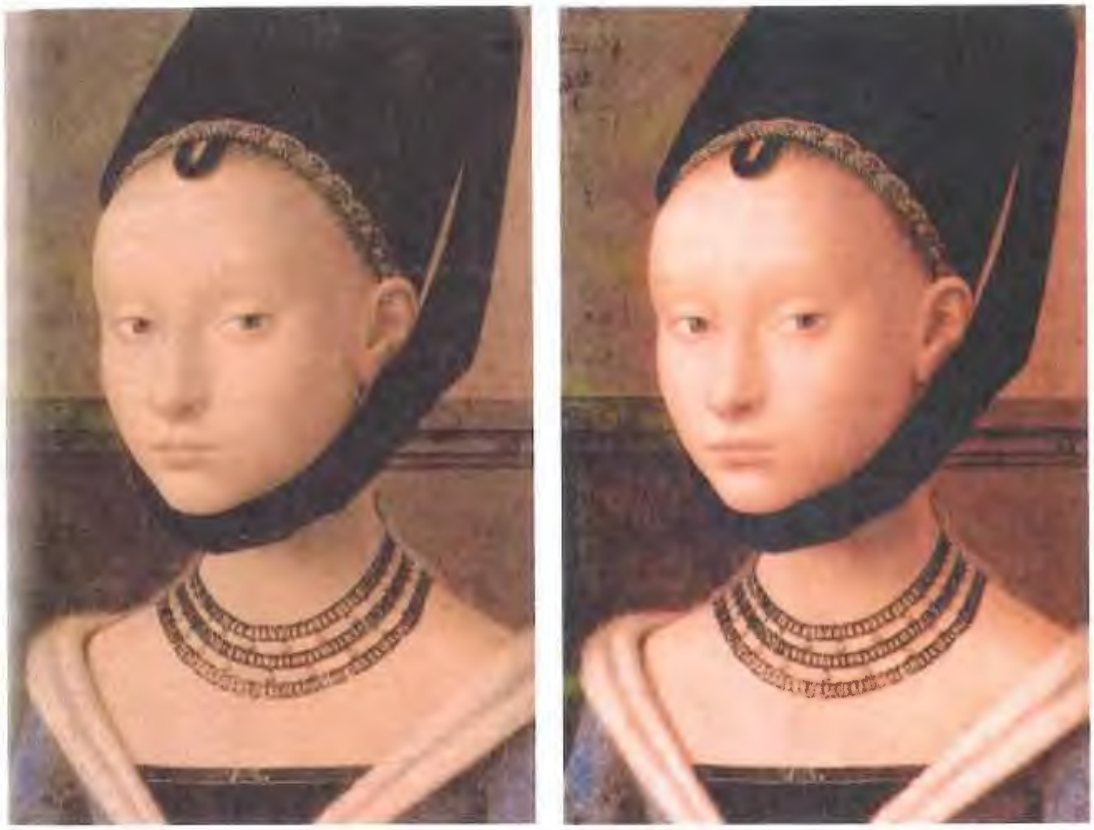
Figure 57. Satellite image of the earth:
(a) *original;*
(b) *with contrast increased by histogram equalization of each color plane;*
(c) *ratio of the image filtered at 450 nm (blue) to that at 650 nm (orange);*
(d) *ratio of the image filtered at 700 nm (red) to that at 550 (green).*

One example of the use of this type of correction is to compensate for the yellowing of varnish and other coatings applied to artworks. **Figure 59** shows an example. The painting is "Portrait of a Lady," painted in 1470 by Petrus Christus, a Flemish Renaissance painter (Gemäldegalerie der Staatlichen Museen, Berlin-Dahlem). The painting shows several defects. The two most obvious are the network of fine cracks that have formed in the paint, and the yellowing and fading of the pigments. Measurements on similar pigments freshly made and applied provide color samples that allow compensation for the color change. The application of a median filter allows filling in the cracks without blurring edges, much as the pixel noise was removed from the grey scale images shown previously. The result is a restoration of the appearance of the original painting.



a b c

Figure 58. Example of rank leveling in a color image as described in the text *(image courtesy Kathy Spencer, The Scripps Research Institute, La Jolla, CA):*
(a) *original image;*
(b) *background produced by color ranking;*
(c) ***a*** *minus* ***b****.*



a b

Figure 59. Portrait of a Lady, painted in 1470 by Petrus Christus:
(a) *present appearance, showing cracks and fading of the colors;*
(b) *application of a median filter to fill in the dark cracks, and color compensation to adjust the colors.*

Non-planar views

Computer graphics is much concerned with methods for displaying the surfaces of three-dimensional objects. Some of these methods will be used in Chapter 12 to display representations of three-dimensional structures obtained from a series of 2D image slices, or from direct 3D imaging methods such as tomography.

One particular use of computer graphics which most of us take for granted can be seen each evening on the local news program. Most TV stations in the U.S. have a weather forecast that uses satellite images from the GOES satellite. These pictures show the U.S. as it appears from latitude 0, longitude 108 W (the satellite is shifted to 98 W in summertime to get a better view of hurricanes developing in the south Atlantic), at a geosynchronous elevation of about 22,000 miles.

This image shows cloud patterns, and a series of images taken during the day shows movement of storms and other weather systems. In these images, the coastlines, Great Lakes, and a few other topographic features are evident, but may be partially obscured by clouds. Given the average citizen's geographical knowledge, that picture would not help most viewers to recognize their location. So computer graphics are used to superimpose political outlines, such as the state borders and perhaps other information such as cities, to assist the viewer. Most U.S. TV stations have heavy investments in computer graphics for advertising, news, etc., but they rarely generate these lines themselves, instead obtaining the images with the lines already present from a company which specializes in that niche market.

How are these lines generated? This is not simply a matter of overlaying a conventional map, say a Mercator's projection as used in the school classroom, over the satellite image. The curvature of the earth and the foreshortening of the image need to be taken into account. **Figure 60** shows a weather satellite image of North America that is clearly foreshortened at the top, and also shows noticeable curvature from west to east across the width of the country.

The coordinates, in latitude and longitude, of points on the earth's surface are used to calculate a perspective view of the roughly spherical globe as it is seen from the satellite. Because the viewpoint is constant, this is a one-time calculation, which nevertheless needs to be done for a great many points to construct good outline maps for superposition. The calculation can be visualized as shown in the diagram of **Figure 61**.



Figure 60. *GOES-7 image of North America with political boundary lines superimposed. The dark area just west of Baja California is the shadow of the moon, during the eclipse of June 11, 1991 (image courtesy National Environmental Satellite Data and Information Service).*

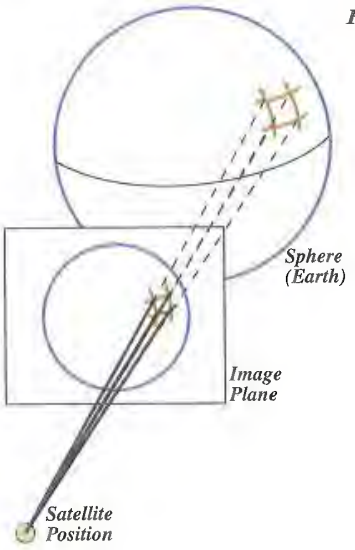


Figure 61. *Diagram of satellite imaging. As in any perspective geometry, the "flat" image is formed by projecting view lines from the three-dimensional object to the viewpoint and constructing the image from the points at which they intersect the image plane.*

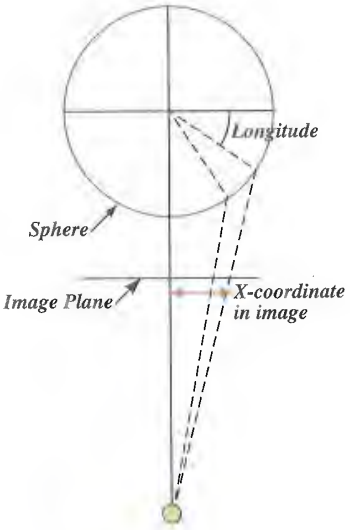


Figure 62. *Simple trigonometry can be used to calculate the location of points in the image plane from the longitude of the point on the earth, and the location of the satellite. This is the view from the North Pole; a similar view from the equator gives the y-coordinate.*

The location of a point on the spherical earth (specified by its latitude and longitude) is used to determine the intersection of a view line to the satellite with a flat image plane, inserted in front of the sphere. This calculation requires only simple trigonometry as indicated in **Figure 62**. The coordinates of the points in that plane are the location of the point in the viewed image. As shown, a square on the ground is viewed as a skewed trapezoid, and if the square is large enough its sides are noticeably curved.

Computer graphics

Computer graphics is similarly used to construct perspective drawings of three-dimensional objects so that they can be viewed on the computer screen, for instance in CAD (computer aided design) programs. The subject goes far beyond our needs here; the interested reader should refer

to standard texts such as Foley and Van Dam (1984) or Hearn and Baker (1986). The display process is the same as that just described, with the addition of perspective control that allows the user to adjust the apparent distance of the camera or viewpoint so as to control the degree of foreshortening that occurs (equivalent to choosing a long or short focal length lens for a camera; the short focal length lens produces more distortion in the image).

Ignoring perspective distortion for the moment (i.e., using a telephoto lens), we can represent the translation of a point in three dimensions by matrix multiplication of its x, y, z coordinates by a set of values that describe rotation and translation. This is simpler to examine in more detail in two dimensions, since our main interest here is with two-dimensional images. Consider a point with Cartesian X,Y coordinates and how it moves when we shift it or rotate it and the other points in the object with respect to the coordinate system.

From simple geometry, we know that a translation of an object simply adds offsets to X and Y, to produce

$$\begin{aligned} X' &= X + \Delta X \\ Y' &= Y + \Delta Y \end{aligned} \tag{3}$$

while stretching the object requires multiplicative coefficients which may not be the same

$$\begin{aligned} X' &= \alpha X \\ Y' &= \beta Y \end{aligned} \tag{4}$$

and rotation of an object by the angle ϑ introduces an interdependence between the original X and Y coordinates of the form

$$\begin{aligned} X' &= X + Y \sin \vartheta \\ Y' &= Y - X \cos \vartheta \end{aligned} \tag{5}$$

In general, the notation for two-dimensional translations is most commonly written using so-called homogeneous coordinates and matrix notation. The coordinates X,Y are combined in a vector along with an arbitrary constant 1 to allow the translation values to be incorporated into the matrix math, producing the result

$$[X'\ Y'\ 1] = [X\ Y\ 1] \cdot \begin{vmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{vmatrix} \tag{6}$$

which multiplies out to

$$\begin{aligned} X' &= aX + cY + e \\ Y' &= bX + dY + f \end{aligned} \tag{7}$$

By comparing this matrix form to the examples above, we see that the e and f terms are the translational shift values. The diagonal values a and d are the stretching coefficients, while b and c are the sine and cosine terms involved in rotation. When a series of transformations is combined, including rotation, translation and stretching, a series of matrices is produced that can be multiplied

together. When this happens, for instance to produce rotation about some point other than the origin, or to combine nonuniform stretching with rotation, the individual terms are combined in ways that complicate their simple interpretation; however, only the same six coefficients are needed.

If only these terms are used, we cannot produce curvature or twisting of the objects. By introducing higher-order terms, more complex stretching and twisting of figures is possible. This would produce a more complex equation of the form

$$X' = a_1 + a_2X + a_3Y + a_4XY + a_5X^2 + a_6Y^2 + \ldots \tag{8}$$

and a similar relationship for Y'. There is no fundamental reason to limit this polynomial expansion to any particular maximum power, except that as the complexity grows the number of coefficients needed rises (and the difficulty of obtaining them), and the mathematical precision needed to apply the transformation increases. It is unusual to have terms beyond second power, which can handle most commonly encountered cases of distortion and even approximate the curvature produced by looking at a spherical surface, at least over small ranges of angles.

Of course, some surface mappings are better handled by other functions. The standard Mercator's projection of the spherical earth onto a cylinder (**Figure 63**) sends the poles to infinity and greatly magnifies areas at high latitudes. It would require many polynomial terms to approximate it, but since the actual geometry of the mapping is known, it is easy to use the cosecant function that efficiently performs the transformation.

Geometrical distortion

Now we must examine what to do with these mathematical operations. Images are frequently obtained that are not of flat surfaces viewed normally. The example of the satellite image used above is one obvious case. So is viewing surfaces in the SEM, in which the specimen surface is often tilted to increase the contrast in the detected image. If the surface is not flat, different regions may be tilted at arbitrary angles, or continuous curvature may be present. This distortion becomes important if we want to perform any measurements or comparisons within or between images. Many airborne cameras and radars introduce a predictable distortion (which is therefore correctable) due

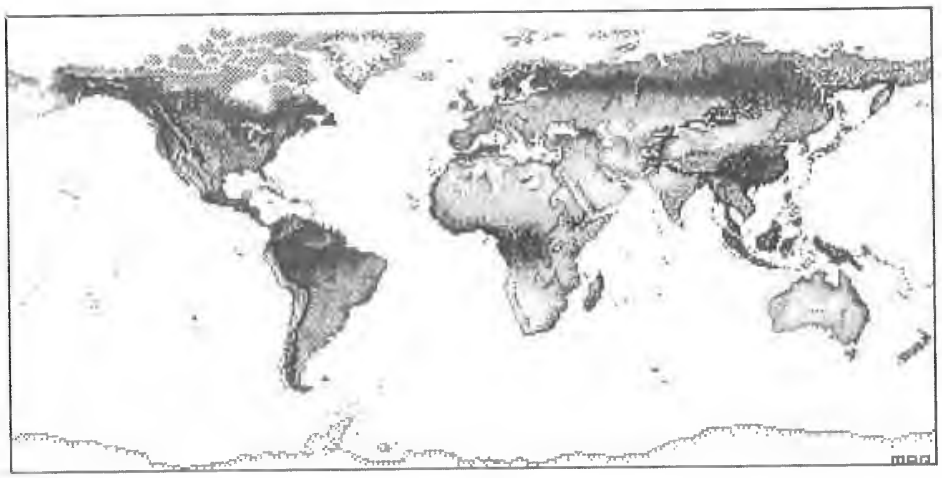


Figure 63. *The standard Mercator's projection of the earth used in maps projects the points on the sphere onto a cylinder, producing distortion at high latitudes.*

to the use of a moving or sideways scan pattern, or by imaging a single line onto continuously moving film. In all of these cases, knowing the distortion is the key to correcting it.

This situation does not commonly arise with light microscopy because the depth of field of the optics is so low that surfaces must be flat and normal to the optical axis to remain in focus. On the other hand, other imaging technologies, frequently encounter non-ideal surfaces or viewing conditions.

Making maps from aerial or satellite images is one application (Thompson, 1966). Of course, there is no perfect projection of a spherical surface onto a flat one, so various approximations and useful conventions are employed. In each case, however, there is a known relationship between the coordinates on the globe and those on the map which can be expressed mathematically. But what about the image? If the viewpoint is exactly known, as for the case of the weather satellite, or can be calculated for the moment of exposure, as for the case of a space probe passing by a planet, then the same kind of mathematical relationship can be determined.

This procedure is usually impractical for aerial photographs, as the plane position is not that precisely controlled. The alternative is to locate a few reference points in the image whose locations on the globe or the map are known, and use them to determine the equations relating position in the image to location on the map. This technique is generally known as image warping or rubber sheeting, and while the equations are the same as those used in computer graphics, the techniques for determining the coefficients are quite different.

We have seen that a pair of equations calculating X', Y' coordinates for a transformed view from original coordinates X,Y may include constants, linear terms in X and Y, plus higher order terms such as XY, X^2, etc. Adding more terms of higher order makes it possible to introduce more complex distortions in the transformation. If the problem is simply one of rotation, only linear terms are needed, and a constraint on the coefficients can be introduced to preserve angles. In terms of the simple matrix shown in **Equation 6**, this would require that the two stretching coefficients a and d must be equal. That means that only a few constants are needed, and they can be determined by locating a few known reference points and setting up simultaneous equations.

More elaborate stretching to align images with each other or with a map requires correspondingly more terms and more points. In electron microscopy, the great depth of field permits acquiring pictures of samples which are locally flat but oriented at an angle to the point of view, producing distortion that is essentially trapezoidal as shown in **Figure 64**. The portion of the surface that is closest to the lens is magnified more than regions farther away, and distances are foreshortened in the direction of tilt. In order to measure and compare features on these surfaces, or even to properly apply image processing methods (which generally assume that the neighbor pixels in various directions are at equal distances from the center), it may be necessary to transform this image to correct the distortion. Since the exact tilt angle and working distance may not be known, a method that uses only reference points within the image itself will be needed.

All that is required here is the ability to identify four points whose real X,Y coordinates on the surface are known, and whose image coordinates X',Y' can be measured. Then the following equations are written

$$X = a_1 + a_2X' + a_3Y' + a_4X'Y' \tag{9}$$

$$Y = b_1 + b_2X' + b_3Y' + b_4X'Y'$$

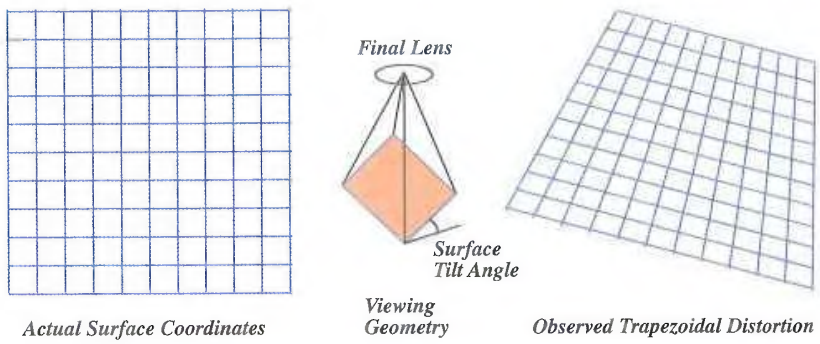


Figure 64. *Trapezoidal distortion commonly encountered in the electron microscope when observing a tilted surface.*

for each of the four sets of coordinates. This allows solving for the constants a_i and b_i. Of course, if more points are available, then they can be used to obtain a least-squares solution that minimizes the effect of the inevitable small errors in measurement of coordinates in the image.

By limiting the equation to those terms needed to accomplish the rotation and stretching involved in the trapezoidal distortion which we expect to be present, we minimize the number of points needed for the fit. More than the three terms shown above in **Equation 7** are required, because angles are not preserved in this kind of foreshortening. But using the fewest possible is preferred to a general equation involving many higher order terms, both in terms of the efficiency of the calculation (number of reference points) and the precision of the coefficients.

Likewise, if we know that the distortion in the image is that produced by viewing a globe, the appropriate sine and cosine terms can be used in the fitting equations. Of course, if we have no independent knowledge about the shape of the surface or the kind of imaging distortion, then other methods such as stereoscopy represent the only practical approach to determining the geometry.

The chief problem that arises with such perspective correction is for features that do not lie on the tilted planar surface, but extend above or below it. In the example of **Figure 65**, the front of the building is corrected to show the correct angles and dimensions for the structure, but the light pole that extends out over the street in front appears to be at an angle to the building when in fact it is exactly perpendicular.

Alignment

Another very common situation is the alignment of serial section images. In this case, there may be no "ground truth" to align to, but only relative alignment between the successive slices. The alignment is performed either by using features within the image that can be recognized in successive slices, or by introducing fiducial marks such as holes drilled through the specimen block or fibers inserted into it before the sections are cut. The points may be located manually by the user, or automatically by the imaging system, although the latter method works best for artificial markings such as holes, and somewhat poorly when trying to use details within the images which match only imperfectly from one section to the next. Relative alignment is discussed in more detail in Chapter 12.

Serial sections cut with a microtome from a block of embedded biological material commonly are foreshortened in the cutting direction by 5 to 15%, due to compression of the block by the knife.



Figure 65. *Correcting trapezoidal distortion due to a non-perpendicular view. Slight distortion remains due to the use of a wide angle lens, and as noted in the text, features out of the plane of the building are incorrectly depicted.*

Then they are rotated arbitrarily before they are viewed. The result is a need for an alignment equation of the form

$$X = a_1 + a_2X' + a_3Y' \tag{10}$$

with only three constants (and a similar equation for Y). Hence, locating three reference points that are common to two sequential images allows one to be rotated and stretched to align with the other. **Figure 66** shows an image in which three points have been marked with vectors to indicate their movement to perform this alignment, and the resulting transformation of the image by stretching.

This kind of warping can be performed to align images with other images, as in serial section reconstruction, or to align images along their edges to permit assembling them as a mosaic (Milgram, 1975). Alignment of side-by-side sections of a mosaic is often attempted with SEM images but fails because of the trapezoidal distortion discussed previously. The result is that features along the

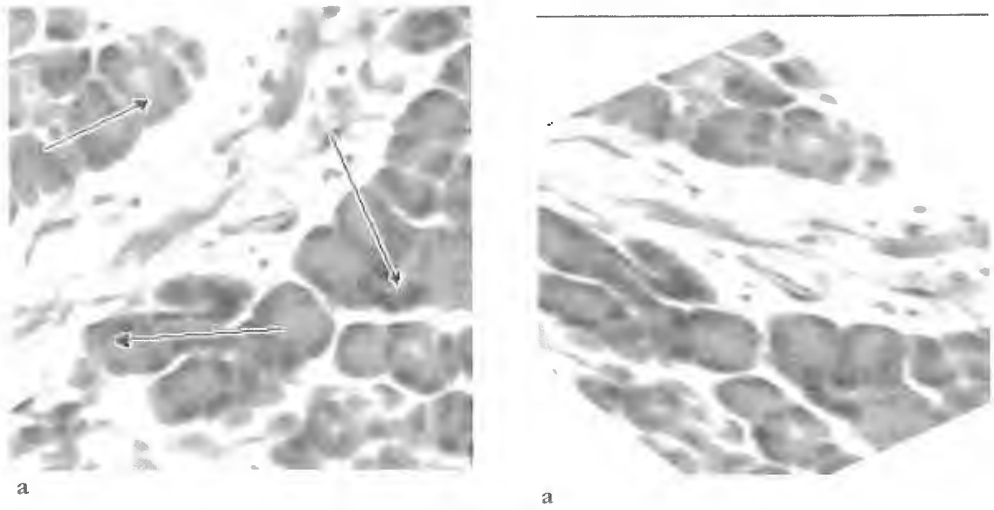


Figure 66. An image with three displacement vectors (a), *and the rotation and stretching transformation which they produce* ***(b)****. Notice that portions of the original image are lost and some areas of the transformed image have no information.*



Figure 67. *Mosaic image of the Valles Marineris on Mars, assembled from satellite images.*

image boundaries do not quite line up, and the mosaic is imperfect. Using rubber sheeting can correct this defect.

Such correction is routinely done for satellite and space probe pictures. **Figure 67** shows an example of a mosaic image constructed from multiple images taken from orbit of the surface of Mars. Boundaries between images are visible because of brightness differences due to variations in illumination or exposure, but the features line up well across the seams. Recently, interest has increased in creating mosaics from separate images to generate panoramic views, and in

Figure 68. *Six images that overlap to provide a 3 × 2 array covering a larger scene.*

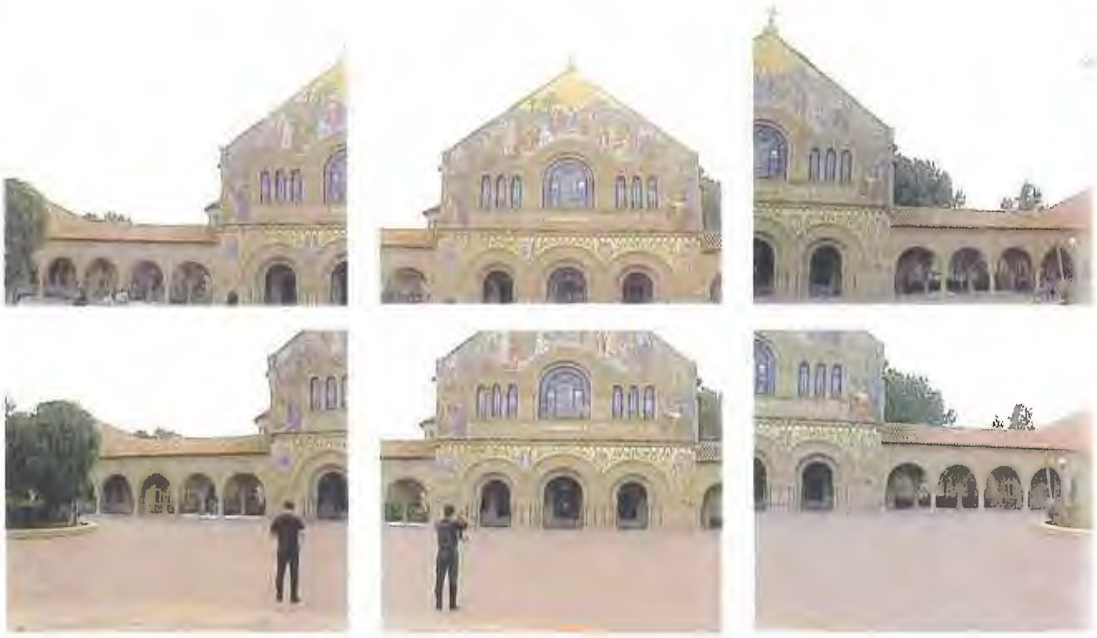




Figure 69. *The mosaic produced from the images in* ***Figure 68*** *(example and software from Enroute Imaging, Palo Alto, CA).*

overcoming the limited resolution (as compared with film) of consumer digital cameras. **Figure 68** shows six individual images taken from a single location (by rotating the camera on a tripod) that form a 3 × 2 array with considerable overlap. **Figure 69** shows the resulting mosaic which forms a complete image of the building. Note that the images have been distorted to fit together smoothly, and that their edges do not form straight lines. This image is pleasing visually but not of much use for measurement purposes.



a

b

Figure 70.
(a) *Large mosaic image assembled from eight individual images, each 1600 × 1200 pixels from a digital camera, as discussed in the text;* ***(b)*** *detail of fit between two image tiles.*

For scientific imaging purposes, the use of a microscope stage or other specimen positioning device that can shift the sample being imaged with reasonable precision while the camera remains fixed would seem to offer the possibility of acquiring images of unlimited size. More constraints are here to assist in the fitting together of the image. It is known, for example, that the images may be slightly rotated (the mechanism may have slight wobble or misalignment and the shifting cannot be counted on to provide exact edge-to-edge alignment), but they cannot be distorted (i.e., straight lines remain straight and angles are unchanged). Thus, the fitting together process can only shift and rotate the individual image tiles in their entirety.

If the overlap between tiles is between 10 and 20%, and the angular mismatch is no more than a few degrees, matching each of the tiles together can indeed produce large high-resolution mosaics as shown in **Figure 70**. The matching technique is based on cross-correlation (discussed in Chapter 5), and an iterative procedure was used to match all of the tiles together for a best fit. This is particularly effective for images acquired in the atomic force microscope, because the area covered by these devices tends to be rather small, and with the very high spatial resolution it is not feasible to design specimen shifting hardware that is absolutely precise (Condeco et al., 2000).

Interpolation

When images are being aligned, it is possible to write the equations either in terms of the coordinates in the original image as a function of the geometrically corrected one, or vice versa. In practice it is usually preferable to use the grid of x,y coordinates in the corrected image to calculate for

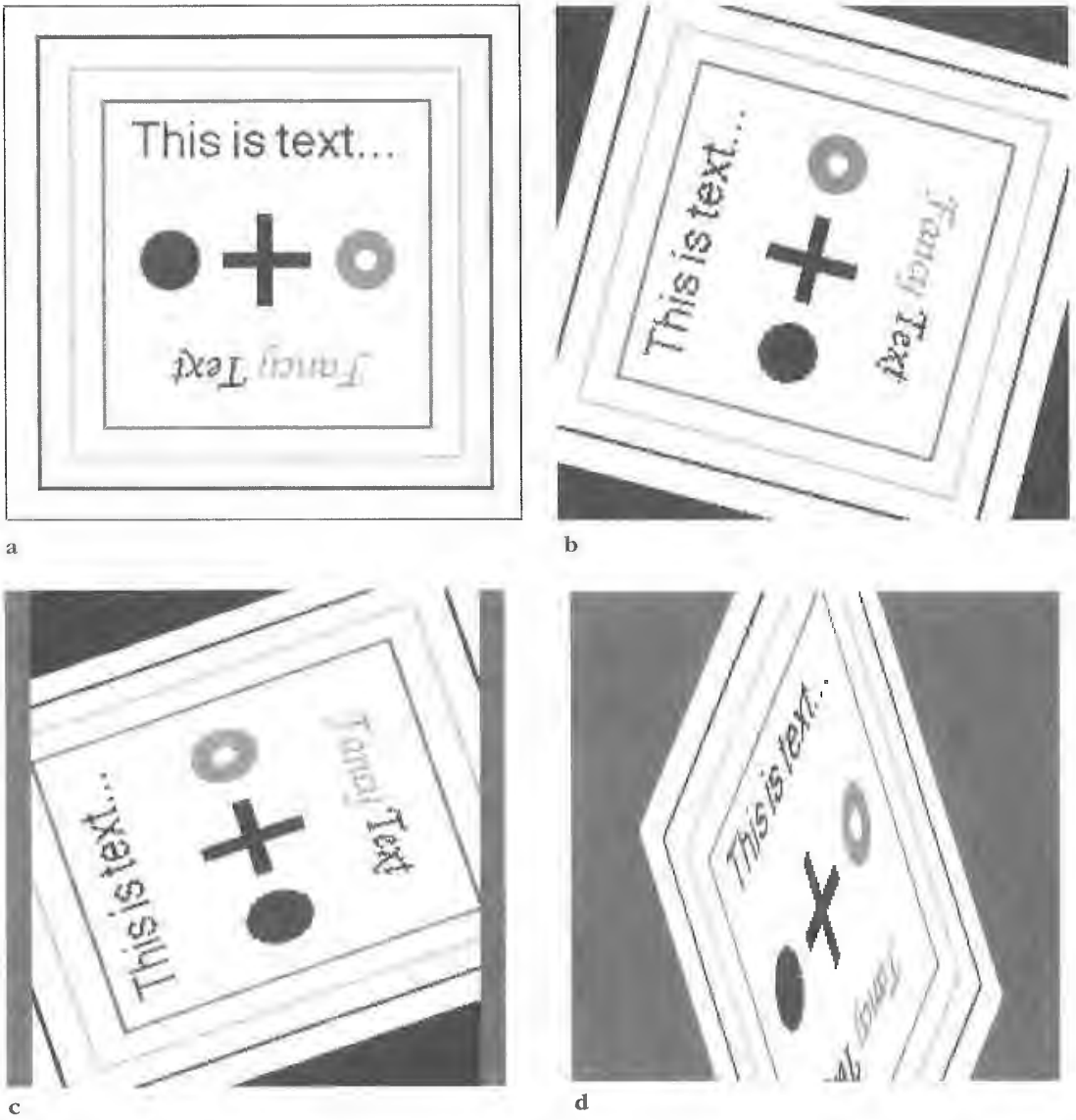


Figure 71. Rotation and stretching of a test image:
(a) *original;*
(b) *rotation only, no change in scale;*
(c) *rotation and uniform stretching while maintaining angles;*
(d) *general rotation and stretching in which angles may vary (lines remain straight).*

each the coordinates in the original image, and to perform the calculation in terms of actual pixel addresses.

Unfortunately, these calculated coordinates for the original location will only rarely be integers. This means that the location lies "between" the pixels in the original image. Several methods are used to deal with this problem. The simplest is to truncate the calculated values so that the frac-

tional part of the address is discarded and the pixel lying toward the origin of the coordinate system is used. Slightly better results are obtained by rounding the address values to select the nearest pixel, whose brightness is then copied to the transformed image array.

Either method introduces some error in location that can cause distortion of the transformed image. **Figure 71** shows examples using a test pattern in which the biasing of the lines and apparent variations in their width is evident.

When this distortion is unacceptable, another method may be used which requires more calculation. The brightness value for the transformed pixel may be calculated by interpolating between the four pixels surrounding the calculated address. This is called bilinear interpolation, and is calculated simply from the fractional part of the X and Y coordinates. First the interpolation is done in one direction, and then in the other, as indicated in **Figure 72.** For a location with coordinates $j + x$, $k + y$ where x and y are the fractional part of the address, the equations for the first interpolation are

$$\begin{aligned} B_{j+x,k} &= (1-x)\cdot B_{j,k} + x\cdot B_{j+1,k} \\ B_{j+x,k+1} &= (1-x)\cdot B_{j,k+1} + x\cdot B_{j+1,k+1} \end{aligned} \tag{11}$$

and then the second interpolation, in the y direction, gives the final value

$$B_{j+x,k+y} = (1-y)\cdot B_{j+x,k} + y\cdot B_{j+x,k+1} \tag{12}$$

Weighted interpolations over larger regions are also used in some cases. One of the most popular is bicubic fitting. Whereas bilinear interpolation uses a 2×2 array of neighboring pixel values to calculate the interpolated value, the cubic method uses a 4×4 array. Using the same notation as the bilinear interpolation in **Equations 11** and **12**, the summations now go from k–1 to k+2 and from j–1 to k+2. The intermediate values from the horizontal interpolation are:

$$B_{j+x,k} = (1/6)\,(B_{j-1,k}\cdot R_1 + B_{j,k}\cdot R_2 + B_{j+1,k}\cdot R_3 + B_{j+2,k}\cdot R_4) \tag{13}$$

and the interpolation in the vertical direction is

$$B_{j+x,k+y} = (1/6)\,(B_{j+x,k-1}\cdot R_1 + B_{j+x,k}\cdot R_2 + B_{j+x,k+1}\cdot R_3 + B_{j+x,k+2}\cdot R_4) \tag{14}$$

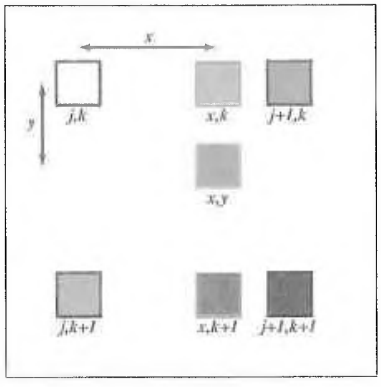


Figure 72. *Diagram of pixel interpolation. The brightness values of the neighbors are first interpolated horizontally to determine the brightness values at the locations outlined in red, and then these two values are interpolated vertically to determine the brightness at the target pixel outlined in blue, using the fractional part of the pixel addresses.*

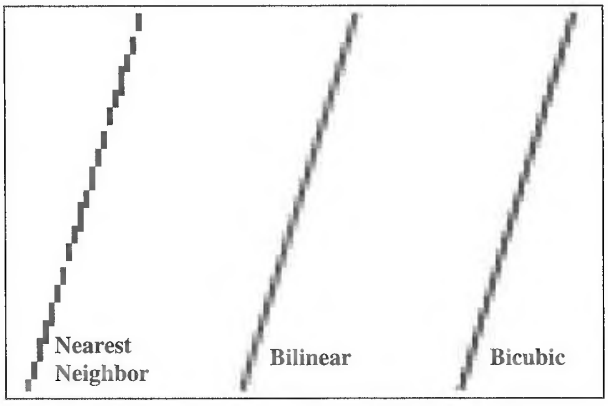


Figure 73. *Effect of rotating a 1-pixel wide black line using nearest neighbor, bilinear, and bicubic interpolation.*

where the weighting factors R_i are calculated from the real part (x or y respectively) of the address as

$$\begin{aligned} R_1 &= (3+x)^3 - 4\cdot(2+x)^3 + 6\cdot(1+x)^3 - 4\cdot x^3 \\ R_2 &= (2+x)^3 - 4\cdot(1+x)^3 + 6\cdot x^3 \\ R_3 &= (1+x)^3 - 4\cdot x^3 \\ R_4 &= x^3 \end{aligned} \tag{15}$$

The bicubic fit is more isotropic than the bilinear method. Interpolation always has the effect of smoothing the image and removing some high frequency information, but minimizes aliasing or "stair-stepping" along lines and edges. **Figure 73** shows the results of rotating a line (originally a black vertical line one pixel wide) by 17° with no interpolation (selecting the nearest neighbor pixel value), bilinear, and bicubic interpolation. The aliasing with the nearest neighbor method is evident. Bilinear interpolation reduces the line contrast more than bicubic, and both assign grey values to adjacent pixels to smooth the appearance of the line.

The advantage of interpolation is that dimensions are altered as little as possible in the transformation, and in particular boundaries and other lines are not biased or distorted. **Figure 74** shows the same examples as **Figure 71**, with bilinear interpolation used. Careful examination of the figure shows that the lines appear straight and not "aliased" or stair-stepped, because some of the pixels along the sides of the lines have intermediate grey values resulting from the interpolation. In fact, computer graphics sometimes uses this same method to draw lines on CRT displays so that the stair-stepping inherent in drawing lines on a discrete pixel array is avoided. The technique is called anti-aliasing and produces lines whose pixels have grey values according to how close they lie to the mathematical location of the line. This fools the viewer into perceiving a smooth line.

Fitting of higher-order polynomials, or adaptive spline fits to the pixel intensity values, can also be used. This can be particularly useful when enlarging images in order to reduce the perceived fuzziness that results when sharp edges are spread out by conventional interpolation. **Figure 75** shows an example (a fragment of the "flowers" image) in which a 4× enlargement has been performed using no interpolation, bilinear interpolation, adaptive spline fitting and fractal interpolation. The latter inserts false "detail" into the image, while spline fitting maintains the sharpness of edges best.

For image warping or rubber-sheeting, interpolation has the advantage that dimensions are preserved, although brightness values are not. With the nearest-pixel method achieved by rounding

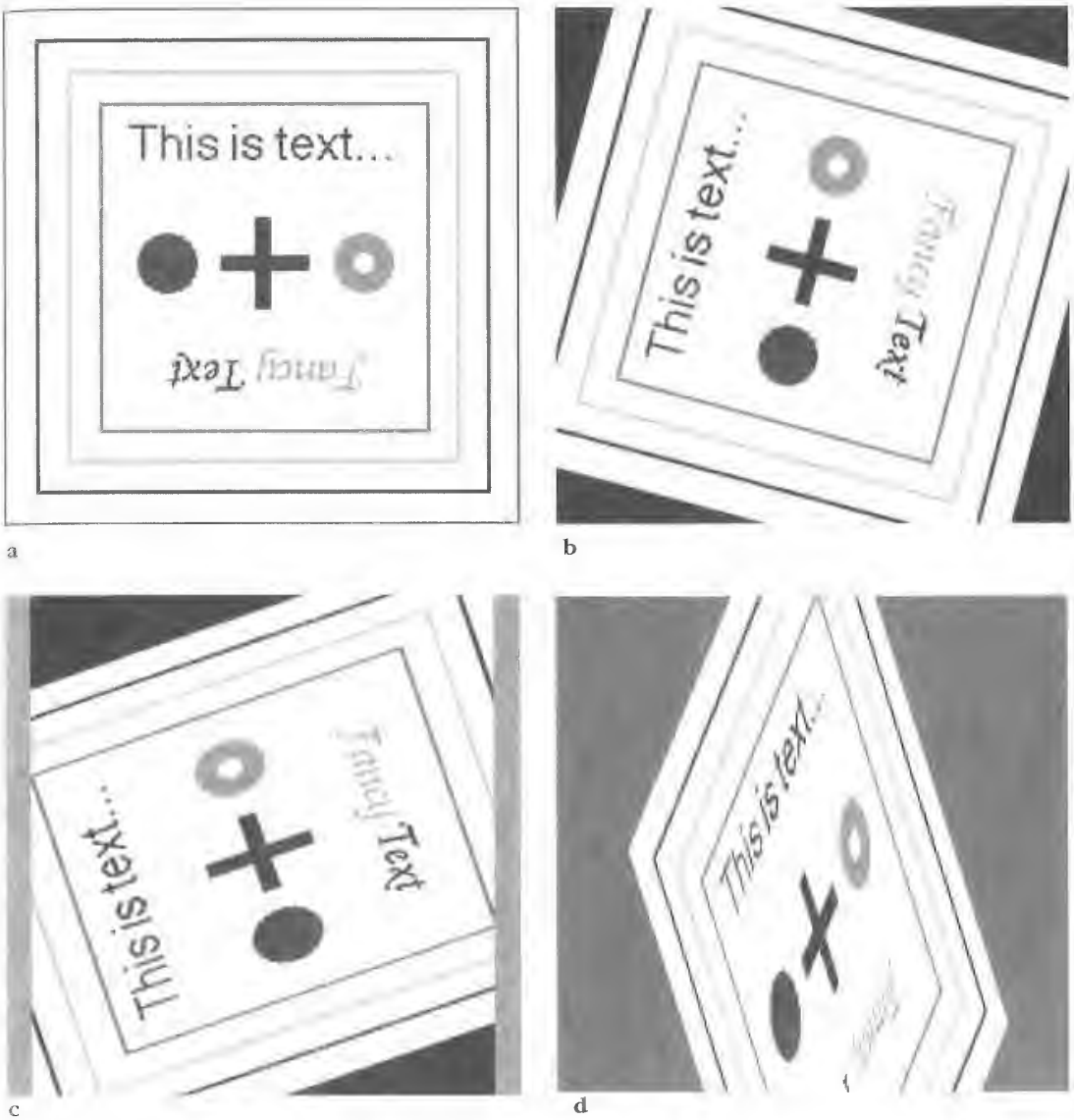


Figure 74. Same generalized rotation and stretching as in ***Figure 71****, but with bilinear interpolation. Note the smoothing of the lines and boundaries.*

the pixel addresses, the dimensions are distorted but the brightness values are preserved. Choosing which method is appropriate to a particular imaging task depends primarily on which kind of information is more important, and secondarily on the additional computational effort required for the interpolation.

Figure 76 illustrates the effect of adding higher order terms to the warping equations. With quadratic terms, the trapezoidal distortion of a short focal length lens or SEM can be corrected. It is also possible to model the distortion of a spherical surface closely over modest distances. With higher order terms arbitrary distortion is possible, but this is rarely useful in an image processing situation since the reference points to determine such a distortion are not likely to be available.

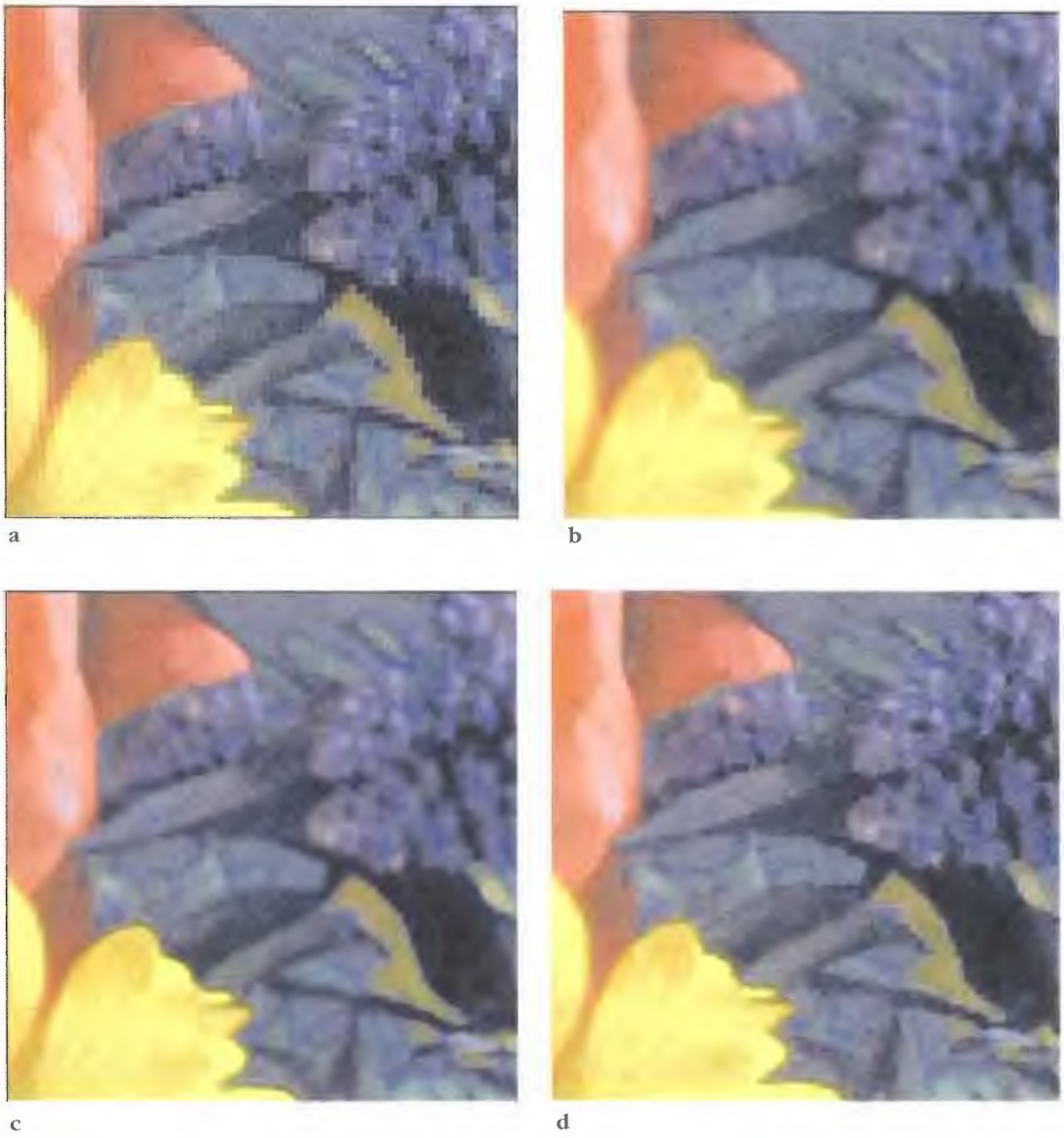


Figure 75. Enlargement of an image:
(a) *no interpolation;*
(b) *bilinear interpolation;*
(c) *spline fitting;*
(d) *fractal interpolation.*

Morphing

Programs that can perform controlled warping according to mathematically defined relationships, or calculate those matrices of values from a set of identified fiducial or reference marks that apply to the entire image are generally rather specialized. But an entire class of consumer-level programs has become available for performing image morphing based on a net of user-defined control points. The points are generally placed at corresponding locations that are distinctive in the two

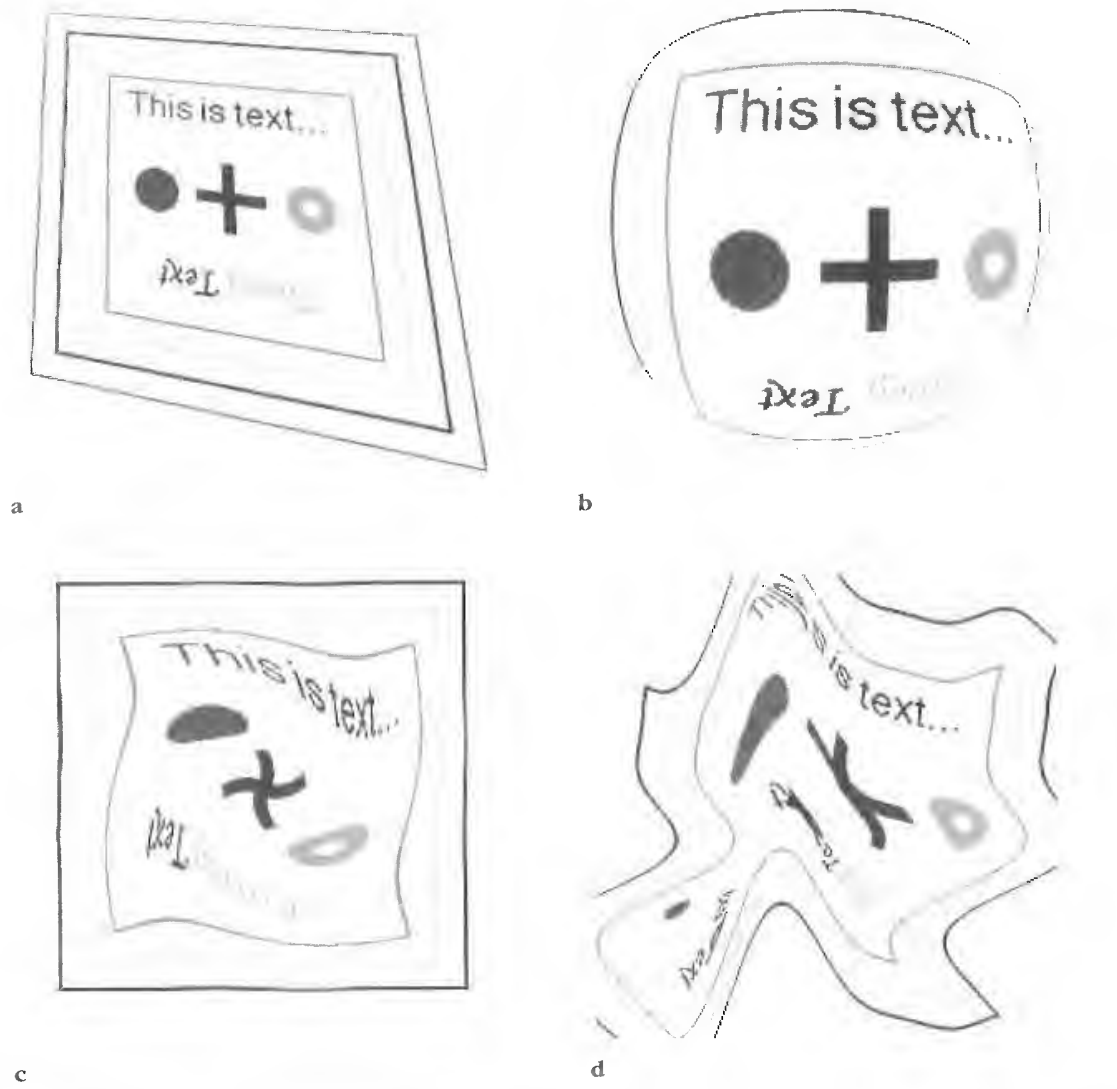


Figure 76. Some additional examples of image warping using the same original image as Figure 71:
(a) *quadratic warping showing trapezoidal foreshortening;*
(b) *cubic warping in which lines are curved (approximation here is to a spherical surface);*
(c) *twisting the center of the field while holding the edges fixed (also cubic warping);*
(d) *arbitrary warping in which higher order and trigonometric terms are required.*

images. For aligning two faces, for example, points at the tips of the eyes, corners of the mouth, along the hairline and chinline, and so forth are used as shown in the illustration in **Figure 77**.

The program uses the points to form a tesselation of the first image into triangles (technically, the choice of which points to use as corners for the triangles is defined by a procedure called a Voronoi tesselation). Each triangle is uniformly stretched to fit the location of the corner points in the second image. The sides of the triangles are uniformly stretched, so the points along the edges of adjacent triangles are not displaced, although lines may be bent where they cross the boundaries of



Figure 77. *Transformation of George into Abe (pictures from U.S. currency). The corresponding points marked on each original image control the gradual distortion from one to the other. The midpoint frame is plausible as a person and shares feature characteristics with both endpoints.*



Figure 78. *Alignment of two images of a clock. The points shown on the two images control the distortion of one image to fit the second. Within the network of triangles, no abrupt distortions are apparent. Around the edges of the image and where pixels on one do not correspond to locations on the other, however, straight lines show sharp bends.*

the triangles. **Figure 78** shows an example of this effect when this procedure is used to rotate one image to match another.

With a triangular mesh and linear stretching of each triangle, lines crossing the boundaries of the triangles would be continuous, but sharply bent. Using spline or cubic equations to control the stretch gives a better appearance by making the curves smooth, but often at the expense of preserving dimensions so that measurements can be made on such images.

The art of using these morphing programs lies primarily in using enough control points, and their careful placement. The results look quite realistic, as shown in the examples. This is especially true when a sequence of images is created with progressive motion of the control points from the

original to final locations. These morphing "movies" show one image transforming gradually into the second. These effects are used routinely in creating television advertisements. It is not clear whether there are any technical applications requiring image measurement that can be satisfactorily accomplished with these programs, considering the somewhat arbitrary placement of points and distortion of dimensions and directions.

The ability to use morphing to align images of different objects and produce visually convincing images can be a powerful tool for communicating results to others. This procedure can be useful to show the similarity between two objects, but it is extraordinarily susceptible to misuse, producing apparent matching between different images that are really not the same.

4

Image Enhancement (Processing in the Spatial Domain)

The preceding chapter discussed methods for correcting or alleviating the principal defects in as-acquired images. A fuzzy area exists between simply correcting these defects and going beyond to enhance the images. Enhancement is the subject of this chapter. Methods are available that increase the visibility of one portion, aspect or component of an image, though generally at the expense of others whose visibility is diminished. In this regard, image processing is a bit like word processing or food processing. It is possible to rearrange things to make a product that is more pleasing or interpretable, but the total amount of data does not change. In the case of images, this generally means that the number of bytes (or pixels) is not reduced.

In contrast to image processing, most image analysis procedures attempt to extract only the "important" information from the image. An example would be to identify and count features in an image, reducing the amount of data from perhaps a million bytes to a few dozen, or even a single "Yes" or "No" answer in some quality control, medical, or forensic applications.

Image processing for purposes of enhancement can be performed in either the spatial domain (the array of pixels comprising our conventional representation of the image) or other domains, such as the Fourier domain discussed in Chapter 5. In the spatial domain, pixel values may be modified according to rules that depend on the original pixel value (local or point processes). In addition, pixel values may be combined with or compared with others in their immediate neighborhood in a variety of ways. Examples of each of these approaches were used in Chapter 3, to replace brightness values in order to expand image contrast, or to smooth noise by kernel averaging or median filtering. Related techniques used in this chapter perform further enhancements.

It is worth noting here that two-dimensional (2D) images typically consist of a very large number of pixels, from about 1/4 million to several million. Even a point process that simply modifies the value of each pixel according to its previous contents requires that the computer address each pixel location. For a neighborhood process, each pixel must be addressed many times, and the processing speed slows down accordingly. Fast processors and high speed memory access (and a lot of memory) are essential requirements for this type of work. Some machines use dedicated hardware such as shift registers and array processors, or boards with dedicated memory and custom addressing circuits, or multiple processors with special programming, to permit near-real-time processing of images, when that is economically justified. As CPU speeds have increased, the need for special hardware has diminished. With more memory, entire images can be accessed rapidly without the need to bring in pieces, one at a time, from disk storage.

As desktop computer power has increased, the implementation of increasingly complex algorithms for image enhancement has become practical. Many of these algorithms are not new, dating from decades ago and often developed in conjunction with the need to process images from satellites and space probes, but until recently have only been performed using large computers or special purpose systems. Most can now be satisfactorily applied to images routinely, using personal computers.

Contrast manipulation

Chapter 3 showed examples of expanding the contrast of a dim image by reassigning pixel brightness levels. In most systems, this can be done almost instantaneously by writing a table of values into the display hardware. This lookup table (LUT) substitutes a display brightness value for each stored value and thus does not require actually modifying any of the values stored in memory for the image. Linearly expanding the contrast range by assigning the darkest pixel value to black, the brightest value to white, and each of the others to linearly interpolated shades of grey makes good use of the display and enhances the visibility of features in the image.

It was also shown, in Chapter 1, that the same LUT approach can be used with colors by assigning a triplet of red, green, and blue values to each stored grey scale value. This pseudo-color also increases the visible difference between similar pixels; sometimes it is an aid to the user who wishes to see small or gradual changes in image brightness.

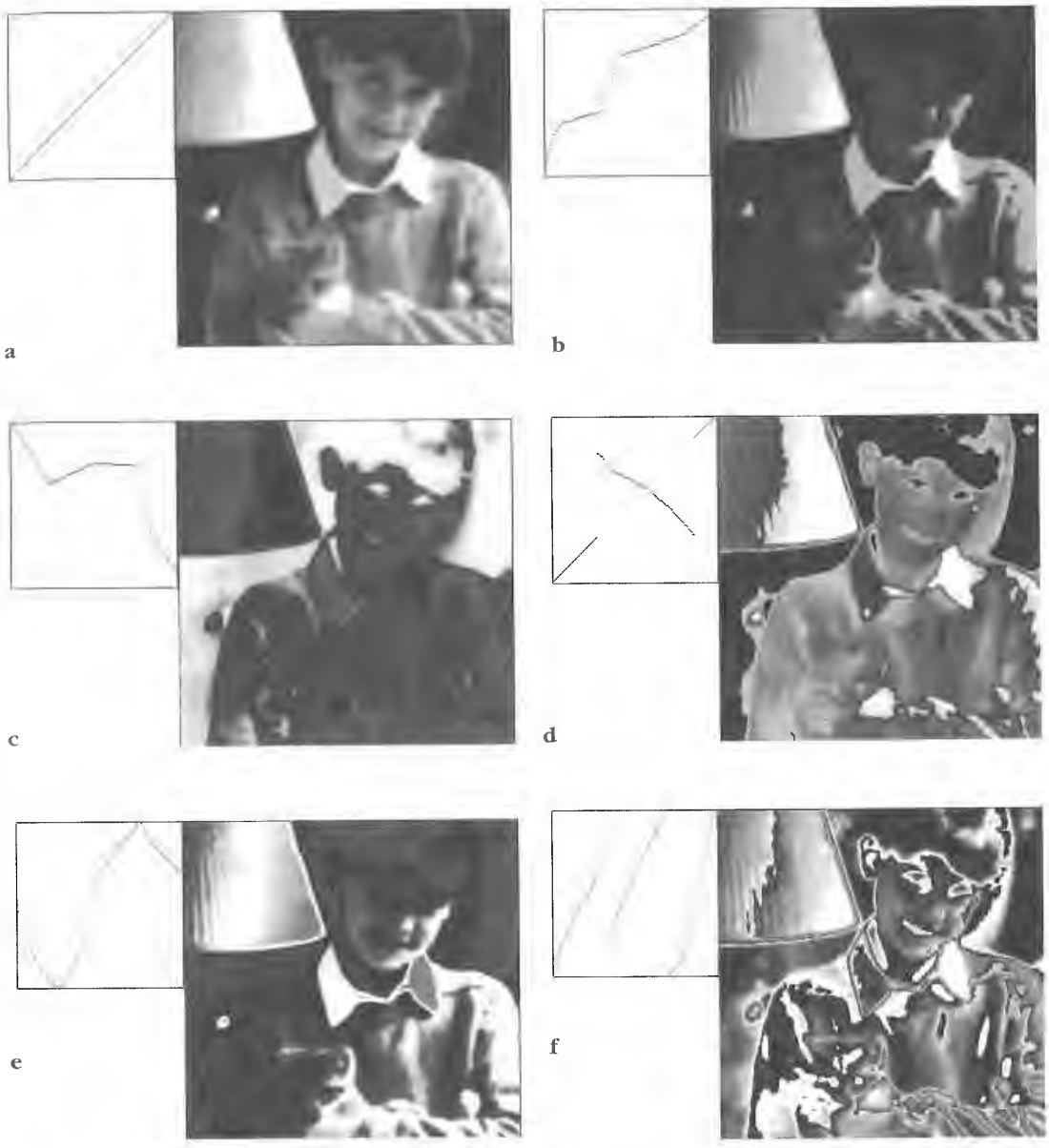
A typical computer display can show 2^8 or 256 different shades of grey, and can produce colors with the same 2^8 brightness values for each of the red, green, and blue components to produce a total of 2^{24} or 16 million different colors. This is often described as "true color," since the gamut of colors that can be displayed is adequate to reproduce most natural scenes. It does not imply, of course, that the colors displayed are photometrically accurate or identical to the original color in the displayed scene. Indeed, that kind of accuracy is very difficult and requires special hardware and calibration. If the original image has more than 256 brightness values (is more than 8 bits deep) in each color channel, some type of lookup table is required even to display it on the screen.

More important, the 16 million different colors that such a system is capable of displaying, and even the 256 shades of grey, are far more than the human eye can distinguish. Under good viewing conditions, we can typically see only a few tens of different grey levels and hundreds of distinguishable colors. That means the display hardware of the image processing system is not being used very well to communicate the image information to the user. If many of the pixels in the image are quite bright, for example, they cannot be distinguished. If there are also some dark pixels present, we cannot simply expand the contrast. Instead, a more complicated relationship between stored and displayed values is needed.

In general, we can describe the manipulation of pixel brightness in terms of a transfer function relating the stored brightness value for each pixel to a displayed value. If this relationship is one-to-one, then for each stored value there will be a unique (although not necessarily visually discernible) displayed value. In some cases, it is advantageous to use transfer functions that are not one-to-one: several stored values are displayed with the same brightness value, so that other stored values can be spread further apart to increase their visual difference.

Figure 1 shows an image in which the 256 distinct pixel brightness values cannot all be discerned even on the display cathode ray tube (CRT); the printed version of the image is necessarily much worse. As discussed in Chapter 2, the number of distinct printed grey levels in a halftone image is determined by the variation in dot size of the printer. For a simple 300-dots-per-inch (dpi) laser writer, this requires a trade-off between grey scale resolution (number of shades) and lateral resolution (number of halftone cells per inch). A typical compromise is 50 cells per inch and more

Figure 1. *An original image with a full range of brightness values, and several examples of arbitrary hand-drawn display transfer functions that expand or alter the contrast in various parts of the range. The plot with each image shows the stored pixel brightness values on the horizontal axis, and the displayed brightness on the vertical axis. Image* ***(a)*** *has a transfer function that is the identity function, so that actual stored brightnesses are displayed. Images* ***(b)*** *through* ***(f)*** *illustrate various possibilities, including reversal and increased or decreased slope over parts of the brightness range.*



than 30 grey levels, or 75 cells per inch and 17 grey levels. The imagesetter used for this book is capable of much higher resolution and more grey levels, but not as many as a CRT or even a photograph.

For comparison purposes, a good-quality photographic print can reproduce 20 to 40 grey levels (the negative can do much better). An instant print such as the Polaroid film commonly used with laboratory microscopes can show 10 to 15 although both have much higher spatial resolution, making the images appear sharper to the eye.

Even so, looking at the original image in **Figure 1**, the viewer cannot see the detail in the bright and dark regions of the image, even on the video screen (and certainly not on the print). Modifying the LUT can increase the visibility in one region or the other, or in both dark and bright regions, provided something else is given up in exchange. **Figure 1** shows several modifications to the original image produced simply by manipulating the transfer function and the LUT by drawing a new relationship between stored and displayed brightness. In this case, a freehand drawing was made that became the transfer function; it was adjusted purely for visual effect.

Note some of the imaging possibilities. A nonlinear relationship can expand one portion of the grey scale range while compressing another. In photographic processing and also in analog display electronics, this is called varying the gamma (the slope of the exposure-density curve). Using a computer, however, we can draw in transfer functions that are far more complicated, nonlinear, and arbitrary than can be achieved in the darkroom.

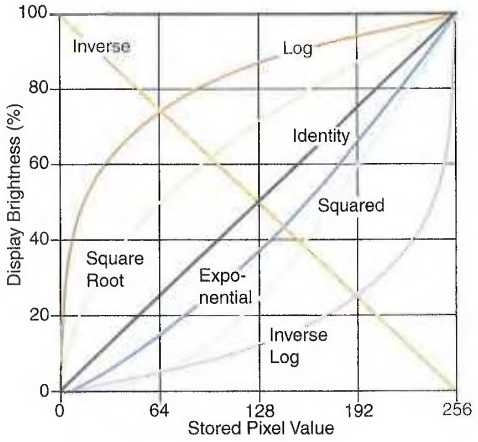
Reversing all of the contrast range produces the equivalent of a photographic negative, which sometimes improves the visibility of details. **Figure 2** illustrates this with an example of an X-ray image, which are commonly examined using negatives. Reversing only a portion of the brightness range produces a visually strange effect, called solarization by photographers, that can also be used to show detail in both shadowed and saturated areas.

Increasing the slope of the transfer function so that it "wraps around" produces an image in which several quite different stored brightness values may have the same display brightness. If the overall organization of the image is familiar to the viewer, though, this contouring may not be too



Figure 2. *X-ray image of a human hand, viewed as a positive and a negative.*

Figure 3. Examples of display transfer functions.



disruptive, and it can increase the visibility for small differences; however, as with the use of pseudo-color, this kind of treatment is easily overdone and may confuse rather than enhance most images.

Certainly, experimentally modifying the transfer function until the image "looks good" and best shows those features of most interest to the viewer provides an ultimately flexible (if dangerous) tool. The technique increases the visibility of some image details but hides others, according to the judgment of the operator. Of course, the same can be said of manipulating image contrast in darkroom printing. In most cases, it is desirable to have more reproducible and meaningful transfer functions available which can be applied equally to a series of images, so that proper comparison is possible.

The most common kinds of transfer functions are shown in **Figure 3**. These are curves of displayed vs. stored brightness following simple mathematical relationships such as logarithmic or power law curves. A logarithmic or square root curve will compress the displayed brightnesses at the bright end of the scale, while expanding those at the dark end. This kind of relationship can also convert an image from a camera with a linear response to the more common logarithmic response. An inverse log or squared curve will do the opposite. The error-function curve compresses middle grey values and spreads the scale at both the bright and dark extremes. An inverse curve simply produces a negative image.

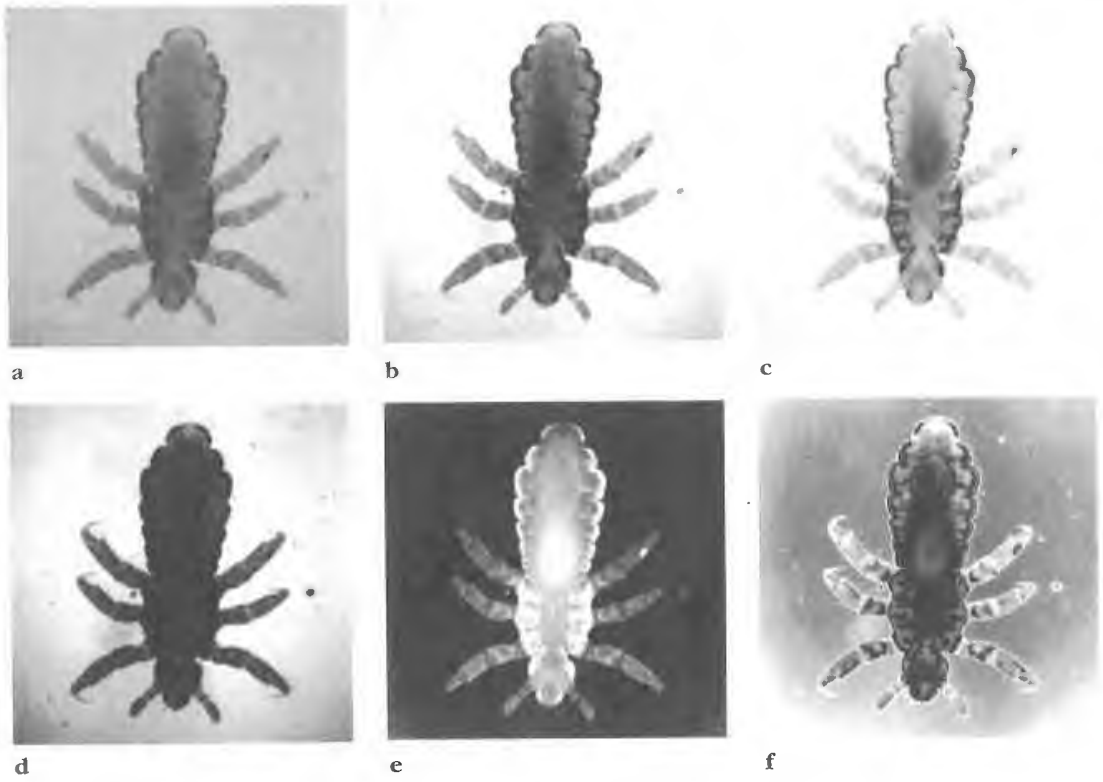
Any of these functions may be used in addition to contrast expansion, which stretches the original scale to the full range of the display. Curves or tables of values for these transfer functions are typically precalculated and stored, so that they can be loaded quickly to modify the display LUT. Many systems allow quite a few different tables to be kept on hand for use when an image requires it, just as a series of color LUTs may be available on disk for pseudo-color displays. **Figure 4** illustrates the use of several transfer functions to enhance the visibility of structures in an image.

Histogram equalization

In addition to pre-calculated and stored tables, it is sometimes advantageous to construct a transfer function for a specific image. Unlike the arbitrary hand-drawn functions shown earlier, however, we desire a specific algorithm that gives reproducible and (hopefully) optimal results. The most popular of these methods is called histogram equalization (Stark and Fitzgerald, 1996). To understand it, we must begin with the image brightness histogram.

Figure 5 shows an example of an image with its histogram. The plot shows the number of pixels in the image having each of the 256 possible values of stored brightness. Peaks in the histogram

Figure 4. Manipulation of the grey-scale transfer function:
(a) an original, moderately low contrast transmission light microscope image (prepared slide of a head louse);
(b) expanded linear transfer function adjusted to the minimum and maximum brightness values in the image;
(c) positive gamma (log) function;
(d) negative gamma (log) function;
(e) negative linear transfer function;
(f) nonlinear transfer function (high slope linear contrast over central portion of brightness range, with negative slope or solarization for dark and bright portions).



correspond to the more common brightness values, which often identify particular structures that are present. Valleys between the peaks indicate brightness values that are less common in the image. Empty regions at either end of the histogram would show that no pixels have those values, indicating that the image brightness range does not cover the full 0–255 range available.

For a color image, it is possible to show three histograms corresponding to the three color axes or channels. As shown in **Figure 6**, this can be done equally well for RGB, HSI, or YUV color coordinates. Any of these sets of histograms are incomplete, however, in that they do not show the combinations of values that are associated in the same pixels. A three-dimensional histogram in which points in the histogram have coordinates that correspond to the color values and show the number of pixels with each possible combination of values, is illustrated in **Figure 7**. Dark values indicate a large number of pixels with a particular combination of components. The projections of the three-dimensional histogram onto each of the 2D faces of the cube are shown. This tool will

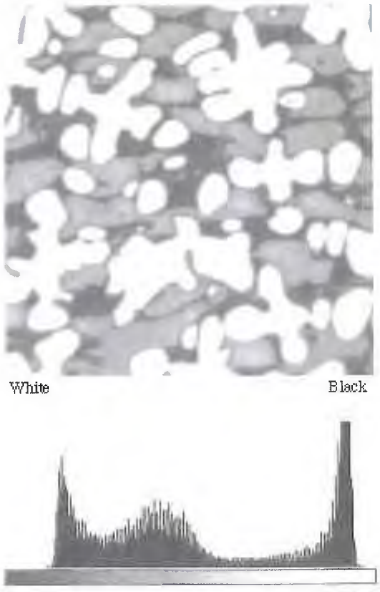


Figure 5. *Metallographic light microscope image of a three-phase metal alloy with its brightness histogram. The plot shows the number of pixels with each of the 256 possible brightness values.*

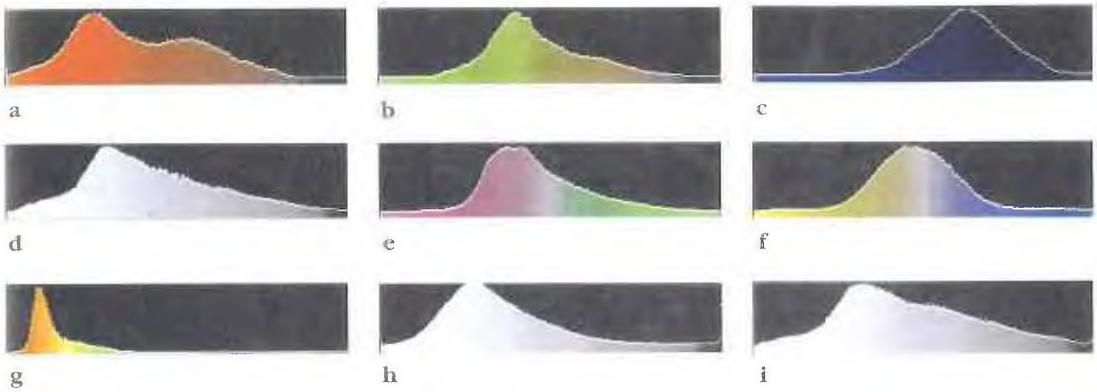


Figure 6. Histograms of individual planes in a color image (the image is a microscopic, thin section of pancreas, which is shown as Figure 52 in Chapter 1):
(a) *red;*
(b) *green;*
(c) *blue;*
(d) *luminance (Y);*
(e) *U;*
(f) *V;*
(g) *hue;*
(h) *saturation;*
(i) *intensity.*

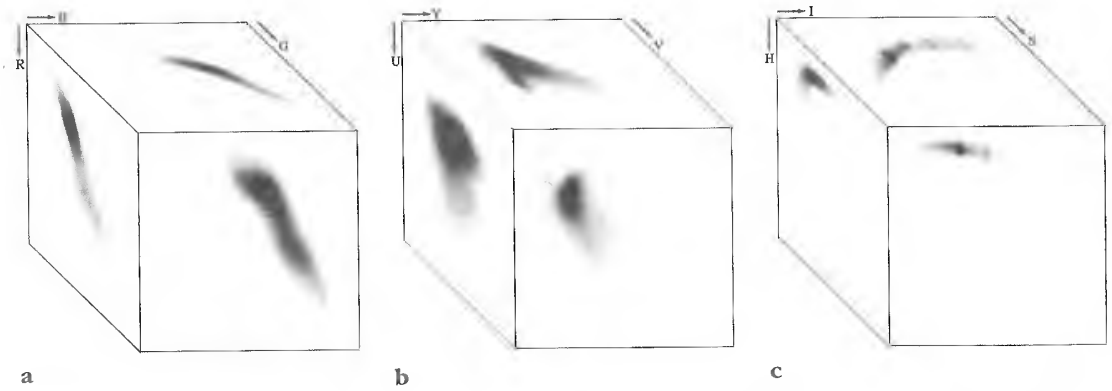
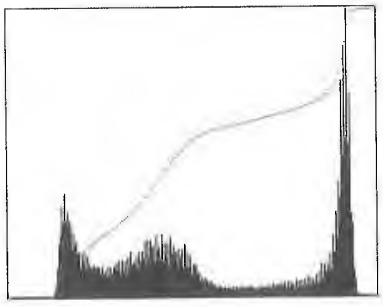


Figure 7. Three-way histograms of the same data shown in Figure 5: *(a) RGB; (b) YUV; (c) HSI.*

be used again in Chapter 6 in the context of selecting color combinations for thresholding, and in Chapter 7 for measuring the co-localization of intensity in different color channels (e.g., different fluorescent dyes).

Generally, images have unique brightness histograms. Even images of different areas of the same sample or scene, in which the various structures present have consistent brightness levels wherever they occur, will have different histograms, depending on the area fraction of each structure. Changing the overall illumination or camera settings will shift the peaks in the histogram. In addition, most real images exhibit some variation in brightness within features (e.g., from the edge to the center) or in different regions.

From the standpoint of efficiently using the available grey levels on the display, some grey scale values are under-utilized. It might be better to spread out the displayed grey levels in the peak areas selectively, compressing them in the valleys so that the same number of pixels in the display show each of the possible brightness levels. This is called histogram equalization. The transfer function is simply the original brightness histogram of the image, replotted as a cumulative plot as shown in **Figure 8**.



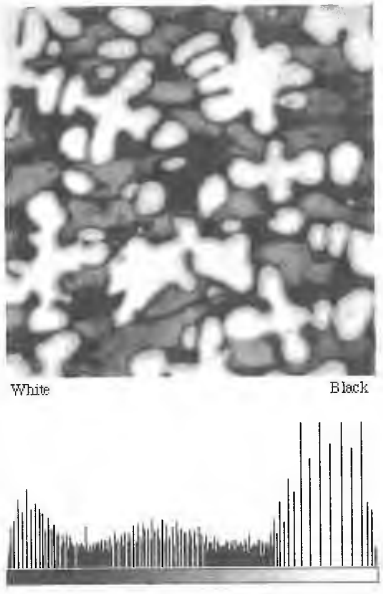


Figure 8. *The original and cumulative histogram from* ***Figure 5 (a)****. The cumulative plot (red line) gives the transfer function for histogram equalization. The processed image and its histogram show the result of applying this function.*

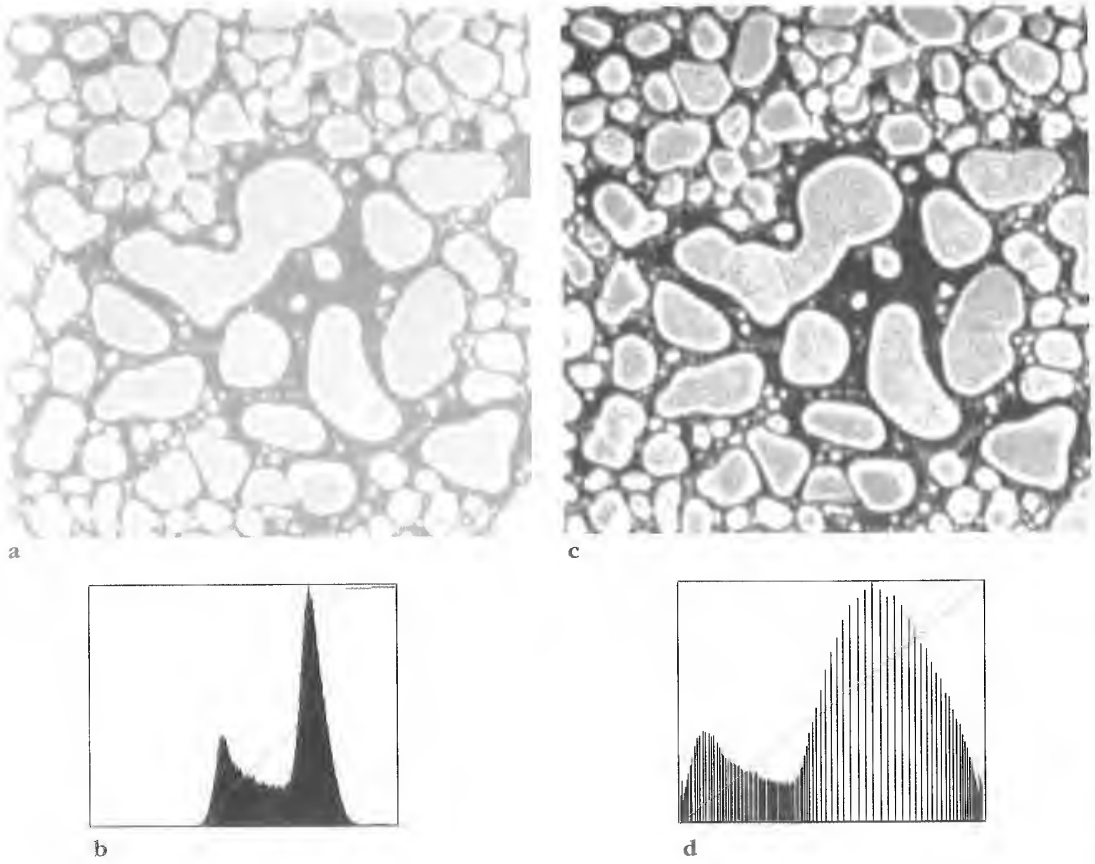


Figure 9. *An SEM image of deposited gold particles* ***(a)*** *with its histogram shown in conventional and cumulative form* ***(b)****. After histogram equalization* ***(c)*** *the cumulative histogram is a straight line* ***(d)****.*

A cumulative histogram helps to understand what histogram equalization does. **Figure 9** shows an SEM image with its original histogram shown in conventional presentation and also as a plot of the total number of pixels less than or equal to each brightness value. After equalization the latter plot is a straight line.

Histogram equalization reassigns the brightness values of pixels based on the image histogram. Individual pixels retain their brightness order (that is, they remain brighter or darker than other pixels) but the values are shifted, so that an equal number of pixels have each possible brightness value. In many cases, this spreads out the values in regions where different regions meet, showing detail in areas with a high brightness gradient.

An image having a few regions in which the pixels have very similar brightness values presents a histogram with peaks. The sizes of these peaks give the relative area of the different phase regions and are useful for image analysis. Performing a histogram equalization on the image spreads the peaks out, while compressing other parts of the histogram by assigning the same or very close brightness values to those pixels that are few in number and have intermediate brightnesses. This equalization makes it possible to see minor variations within regions that appeared nearly uniform in the original image.

The process is quite simple. For each brightness level j in the original image (and its histogram), the new assigned value k is calculated as

$$k = \sum_{i=0}^{j} N_i / T \tag{1}$$

where the sum counts the number of pixels in the image (by integrating the histogram) with brightness equal to or less than j, and T is the total number of pixels (or the total area of the histogram).

Figure 8 showed an example. The original metallographic specimen has three phase regions with dark, intermediate, and light grey values. These peaks are separated in the histogram by regions having fewer pixels with intermediate brightness values, often because they happen to fall on the boundary between lighter and darker phase regions. Histogram equalization shows the shading within the phase regions, indiscernible in the original image. It also spreads out the brightness values in the histogram, as shown.

Some pixels that originally had different values are now assigned the same value, which represents a loss of information, while other values that were once very close together have been spread out, leaving gaps in the histogram. This image of a polished metal shows well-defined peaks corresponding to different phases present. The histogram-equalized image is less satisfying in terms of showing the distinction between the phases, but much better at showing the gradation in the white phase, which was not visible in the original.

This equalization process need not be performed on an entire image. Enhancing a portion of the original image, rather than the entire area, is also useful in many situations. This is particularly true when large regions of the image correspond to different types of structures or scenes and are generally brighter or darker than the rest of the image. When portions of an image can be selected, either manually or by some algorithm based on the variation in the brightness histogram, a selective equalization can be used to bring out local detail.

Figure 10 shows regions of the same test image used previously, with three regions separately selected and equalized. Two of these regions are arbitrary rectangular and elliptical shapes, drawn by hand. In many cases, this kind of manual selection is the most straightforward way to specify a region. The third area is the lampshade, which can be isolated from the rest of the image because locally the boundary is a sharp transition from white to black. Methods for locating boundary edges are discussed later in the chapter.

Each of the three regions shown was used to construct a brightness histogram, which was then equalized. The resulting reassigned brightness values were then substituted only for the pixels within the region. In the two dark regions (the girl's face and the region near her shoulder), equalization produces an overall brightening as well as spreading out the brightness values so that the small variations are more evident. In the bright area (the lampshade), the overall brightness is reduced; again, more variation is evident.

In all cases, the brightness values in the equalized regions show some contouring, the visible steps in brightness produced by the small number of different values present in these regions in the original image. Contouring is usually considered to be an image defect, since it distorts the actual smooth variation in values present in the image. When it is introduced specifically for effect, it may be called posterization (not to be confused with the use of the same term for the result of repeatedly applying a median filter, discussed in Chapter 3).

It should be noted, of course, that once a region has been selected, either manually or automatically, the contrast can be modified in any way desired: by loading a pre-stored LUT, calculating a



Figure 10. *Selective equalization can be performed in any designated region of an image. In this example, three regions have been selected and histogram equalization performed within each one separately. In the dark regions (the girl's face and the area behind her shoulder), this lightens the image and shows some of the detail that is otherwise not visible. In the light region (the lampshade), this results in an overall average darkening and expansion of the contrast to show shading. In both cases, the process is ultimately limited by the number of discrete grey levels which were actually present in those regions in the original image.*

histogram equalization function, or simply stretching the grey scale linearly to maximize contrast; however, when these operations are applied to only a portion of the entire image, it is not possible to manipulate only the display LUT. Since the stored grey values in other regions of the image are to be shown with their original corresponding display values, it is necessary to actually modify the contents of the stored image to alter the display. This is a very fast operation, since it is performed using a transfer function that is loaded or precalculated from the histogram and each pixel is accessed one time.

Histogram equalization of regions within an image can dramatically improve the local visibility of details, but it usually alters the relationship between brightness and structure. In most cases, it is desirable for the brightness level of pixels associated with a particular type of feature in the image to be the same, wherever in the field of view it may occur. This allows rapid classification of the features for counting or measurement. Local modification of the grey scale relationship voids this assumption, making the display brightness of features dependent on other features that happen to be nearby or in the selected region.

Histogram equalization assigns the same new grey value to each pixel having a given original grey value, everywhere in the region. The result is that regions often have very noticeable and abrupt boundaries, which may or may not follow feature boundaries in the original image. Another approach performs the equalization in a region around each pixel in the image, with the result applied separately to each individual pixel. This is normally done by specifying a neighborhood size (typically round or square). All of the pixels in that region are counted into a histogram, but the equalization procedure is applied only to the central pixel. This process is then repeated for each pixel in the image, always using the original brightness values to construct the histogram.

Figure 11 shows an example in which the neighborhood size used is an approximately circular area 11 pixels wide containing 89 pixels (**Figure 12**). The process is not applied within a distance of five pixels of the edges of the image, where the region cannot be centered on a pixel. Special rules can be made for these points, if required. Some programs implement local equalization using a square neighborhood, which is slightly easier computationally but is less isotropic. The actual calculation is quite simple, because for each pixel the equalized brightness value is just the number of darker pixels in the neighborhood. The neighborhood contains 89



a b

Figure 11. *Local equalization applied to every pixel in an image. This is an image of a fingerprint on a magazine cover, which is difficult to visually distinguish on the dark and light printed regions. For each pixel in the original image* ***(a)****, a 9 × 9 neighborhood has been used to construct a histogram. Equalization of that histogram produces a new brightness value for each pixel, but this is only saved for the central pixel. The process is repeated to apply to each pixel in the original image except those too near the edge, producing the result* ***(b)****. Enlargements to shown individual pixels are shown in (c) and (d). The effect of this local equalization is to enhance edges, and reveal detail in light and dark regions; however, in the center of uniform regions, it can produce artefacts and noise.*

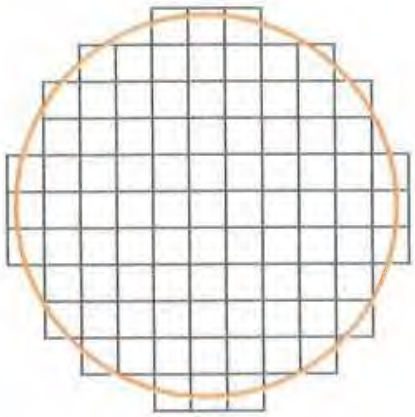


Figure 12. *Approximation to a circle 11 pixels wide contains a total of 89 pixels. Using a round neighborhood minimizes directional distortion and artefacts in the image due to local equalization.*

pixels, so the procedure produces an image in which the maximum number of distinct brightness values is 89, instead of the original 256, but this is still more than can be visually distinguished in the display. The process is applied to each pixel in the image, while constructing a new image containing the result, so that only the original values of the neighboring pixels are used to perform the equalization.

The process of local equalization enhances contrast near edges, revealing details in both light and dark regions, although it can be seen in the example that, in nominally uniform regions, this can produce artificial contrast variations that magnify the visibility of noise artefacts in the image. In general, it is important to remove or reduce noise as discussed in Chapter 3 before performing enhancement operations.

Local equalization works by making pixels that are slightly darker than their surroundings much darker, and ones that are slightly lighter, much lighter. Changing the size of the neighborhood used in local equalization offers only a small degree of control over the process. Other modifications that are more effective (and generally combined under the term "adaptive" equalization) are to weight the pixels according to how close they are to the center of the neighborhood, or according to how similar they are in brightness to the central pixel, or to include in the neighborhood only pixels "similar" to the central one. Since equalization is performed by constructing a histogram of the pixels and finding the midpoint of that histogram, the weighting is accomplished by assigning values greater or less than one to the pixels for use in summing up the histogram.

Figure 13 shows an example of adaptive equalization in which center weighting has been used to increase the visibility of fine detail, and value weighting has been used to reduce the effects of noise in the image. In this scanning electron microscope (SEM) image of dried paint, the large contrast range resulting from local variations in surface slope and the difficulty of collecting signal from within deep holes makes it difficult to see local detail. Local equalization (**Figure 13b**) enhances the visibility of the detail, but also the noise. Adaptive equalization (**Figure 13c**) produces a much better result. In both cases, the equalized result was added back to the original image to preserve some of the overall contrast range.

When it is used with color images, local or adaptive equalization is properly applied only to the intensity data. The image is converted from its original RGB format in which it is stored internally, the intensity values are modified and then combined with the original hue and saturation values so that new RGB values can be calculated to permit display of the result. This preserves the color information, which would be seriously altered if the red, green, and blue channels were equalized directly, as shown in **Figure 14** (Buzuloiu et al., 2001).

Laplacian

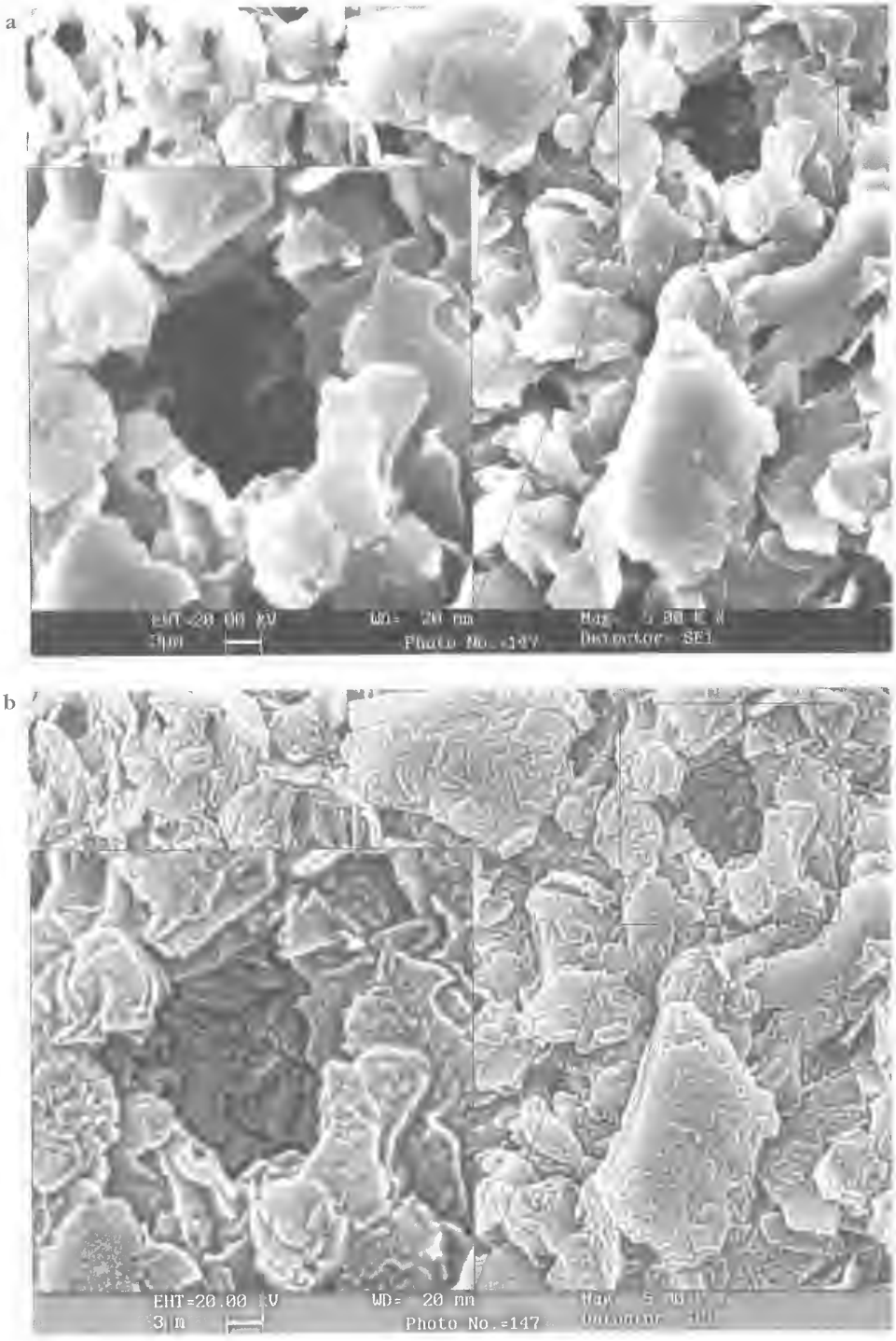
Local, or neighborhood, equalization of image contrast produces an increase in local contrast at boundaries, as shown in **Figure 15**. This has the effect of making edges easier for the viewer to see, consequently making the image appear sharper (although in the example the increased noise visibility within the uniform grains defeats this improvement). This section will discuss some other approaches to edge enhancement that are less sensitive to overall brightness levels and feature sizes than the equalization discussed above.

The first set of operations uses neighborhoods with multiplicative kernels, usually with integer weights, identical in principle to those used in Chapter 3 for noise smoothing. In the section on smoothing, kernels were written as an array of integers. For example,

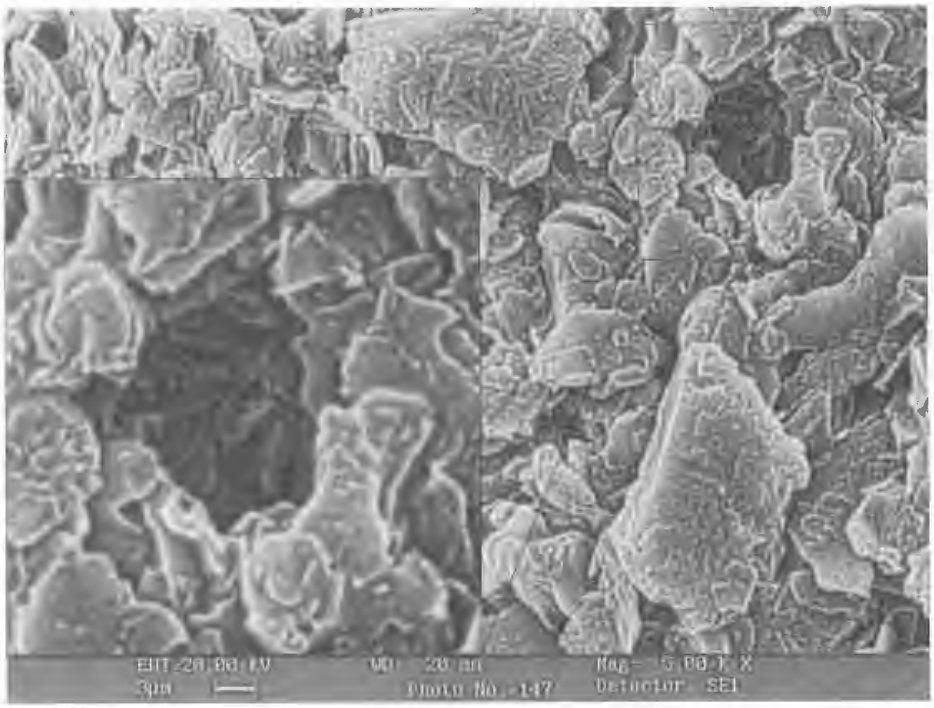
1	2	1
2	4	2
1	2	1

This 3×3 kernel is understood to mean that the central pixel brightness value is multiplied by 4, the values of the 4 touching neighbors to the sides and above and below are multiplied by 2, and the 4 diagonally touching neighbors by 1. The total value is added up and then divided by 16 (the sum of the nine weights) to produce a new brightness value for the pixel. Other arrays of weights were also shown, some involving much larger arrays than this simple 3×3. For smoothing, all of the kernels were symmetrical about the center, and the examples shown in Chapter 3 had only positive weight values.

Figure 13. SEM image of dried paint: *(**a**) original; (**b**) conventional local equalization; (**c**) adaptive local equalization with center weighting and noise reduction. In both **b** and **c**, the equalized result has been added back to the original image.*



c



A very simple kernel that is still symmetrical, but does not have exclusively positive values, is the 3 × 3 Laplacian operator

–1	–1	–1
–1	+8	–1
–1	–1	–1

This subtracts the brightness values of each of the neighboring pixels from the central pixel. Consequently, in a region of the image that is uniform in brightness or has a uniform gradient of brightness, the result of applying this kernel is to reduce the grey level to zero. When a discontinuity is present within the neighborhood in the form of a point, line, or edge, the result of the Laplacian is a non-zero value. It may be either positive or negative, depending on where the central point lies with respect to edge, etc.

In order to display the result when both positive and negative pixel values arise, it is common to add a medium grey value (128 for the case of a single-byte-per-pixel image with grey values in the range from 0 to 255) so that the zero points are middle grey and the brighter and darker values produced by the Laplacian can be seen. Some systems instead plot the absolute value of the result, but this tends to produce double lines along edges that are confusing both to the viewer and to subsequent processing and measurement operations.

As the name of the Laplacian operator implies, it is an approximation to the linear second derivative of brightness B in directions x and y

$$\nabla^2 B \equiv \frac{\partial^2 B}{\partial x^2} + \frac{\partial^2 B}{\partial y^2} \tag{2}$$

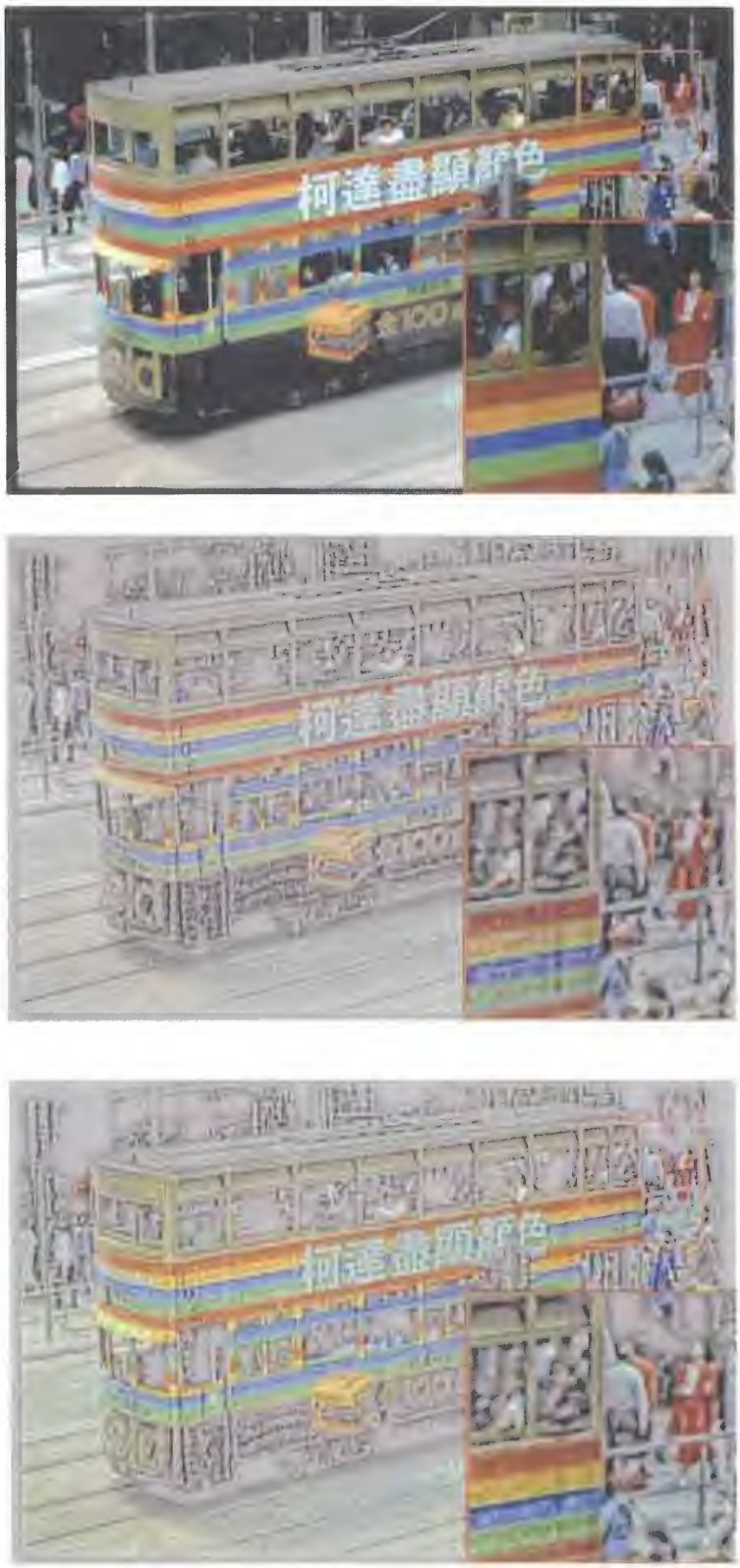


Figure 14. Color photograph *(courtesy Eastman Kodak Co.):* ***(a)*** *original;* ***(b)*** *adaptive equalization of the individual red, green, and blue channels;* ***(c)*** *adaptive equalization of intensity, leaving hue and saturation unchanged. Note the increased visibility of figures and detail in shadow areas, and the false color variations present in* ***b****.*

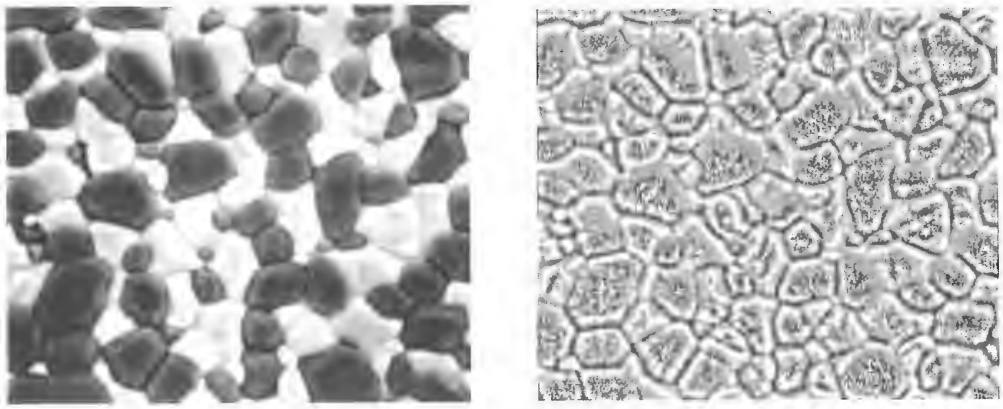


Figure 15. Local (not adaptive) equalization of a two-phase material (alumina-zirconia). Note the increased contrast at boundaries, and the noise artefacts within the grains.

which is invariant to rotation, and hence insensitive to the direction in which the discontinuity runs. This highlights the points, lines, and edges in the image and suppresses uniform and smoothly varying regions, with the result shown in **Figure 16**. By itself, this Laplacian image is not very easy to interpret. Adding the Laplacian enhancement of the edges from the original image restores the overall grey scale variation, which the human viewer can comfortably interpret. It also sharpens the image by locally increasing the contrast at discontinuities as shown in the figure. This can be done simply by changing the weights in the kernel, so that it becomes

$$\begin{matrix} -1 & -1 & -1 \\ -1 & +9 & -1 \\ -1 & -1 & -1 \end{matrix}$$

This kernel is often described as a sharpening operator, because of the improved image contrast that it produces at edges. Justification for the procedure can be found in two different explanations. First, consider blur in an image to be modeled by a diffusion process in which brightness spreads out across the image, which would obey the partial differential equation



***Figure 16. Enhancement of contrast at edges, lines, and points using a Laplacian:** (a) original SEM image of ceramic fracture; (b) application of Laplacian operator; (c) addition of the Laplacian to the original image.*

$$\frac{\partial f}{\partial t} = k\nabla^2 f \tag{3}$$

where the blur function is $f(x,y,t)$ and t is time. If this is expanded into a Taylor series around time τ, we can express the unblurred image as

$$B(x,y) = f(x,y,\tau) - \tau\frac{\partial f}{\partial t} + \frac{\tau^2}{2}\frac{\partial^2 f}{\partial t^2} - \ldots \tag{4}$$

If the higher-order terms are ignored, this is just

$$B = f - k\tau\nabla^2 f \tag{5}$$

In other words, the unblurred image B can be restored by subtracting the Laplacian (times a constant) from the blurred image. While the modeling of image blur as a diffusion process is at best approximate and the scaling constant is unknown or arbitrary, this at least gives some plausibility to the approach.

At least equally important is the simple fact that the processed image "looks good." **Figure 17** illustrates this with an astronomical photograph of Saturn. The visibility of the fine detail in the rings and atmosphere is enhanced by processing. The human visual system itself concentrates on

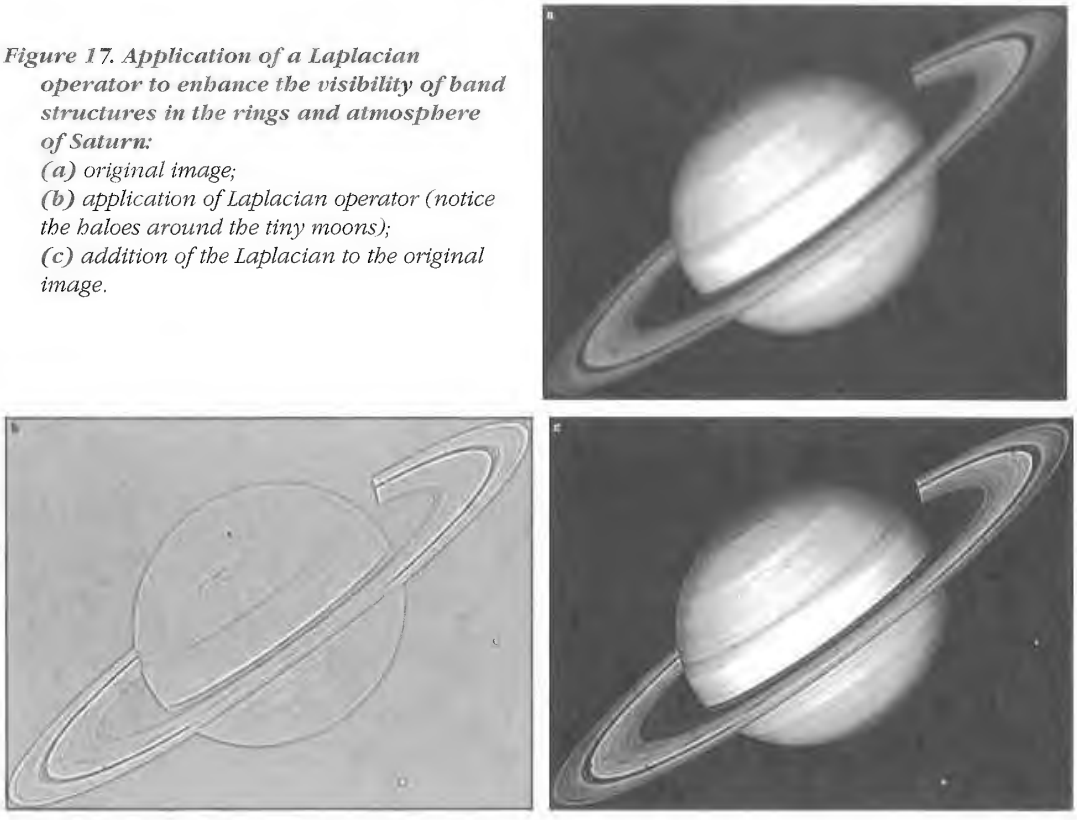


Figure 17. Application of a Laplacian operator to enhance the visibility of band structures in the rings and atmosphere of Saturn:
(a) *original image;*
(b) *application of Laplacian operator (notice the haloes around the tiny moons);*
(c) *addition of the Laplacian to the original image.*

edges and ignores uniform regions (Marr and Hildreth, 1980; Marr, 1982; Hildreth, 1983). This capability is hard-wired into our retinas. Connected directly to the rods and cones of the retina are two layers of processing neurons that perform an operation very similar to the Laplacian. The horizontal cells in the second layer average together the signals from several neighboring sensors in the first layer, and the bipolar cells in the third layer combine that signal with the original sensor output. This is called local inhibition and helps us to extract boundaries and edges. It also accounts for some of the visual illusions shown in Chapter 1.

At a more practical level, consider the ability of the eye to respond to cartoons and line drawings. These are highly abstracted bits of information about the original scene, yet they are entirely recognizable and interpretable. The cartoon provides the eye with exactly the minimum information it would otherwise have to extract from the scene itself to transmit up to higher levels of processing in the brain. (Similar inhibition in the time domain, using the next two layers of retinal neurons, helps us detect motion.)

One characteristic of human vision that confirms this behavior is the presence of Mach bands, a common illusion resulting from local brightness inhibition (Mach, 1906; Cornsweet, 1970). **Figure 18** shows a series of vertical bands of uniform intensity. The human viewer does not perceive them as uniform, but sees an undershoot and overshoot on each side of the steps as shown in the plot. This increases the contrast at the step, and hence its visibility.

To see how this works using the Laplacian, we will use a one-dimensional kernel of the form

$$-1 \quad +2 \quad -1$$

and apply it to a series of brightness values along a line profile, across a step of moderate steepness

2 2 2 2 2 4 6 6 6 6 6

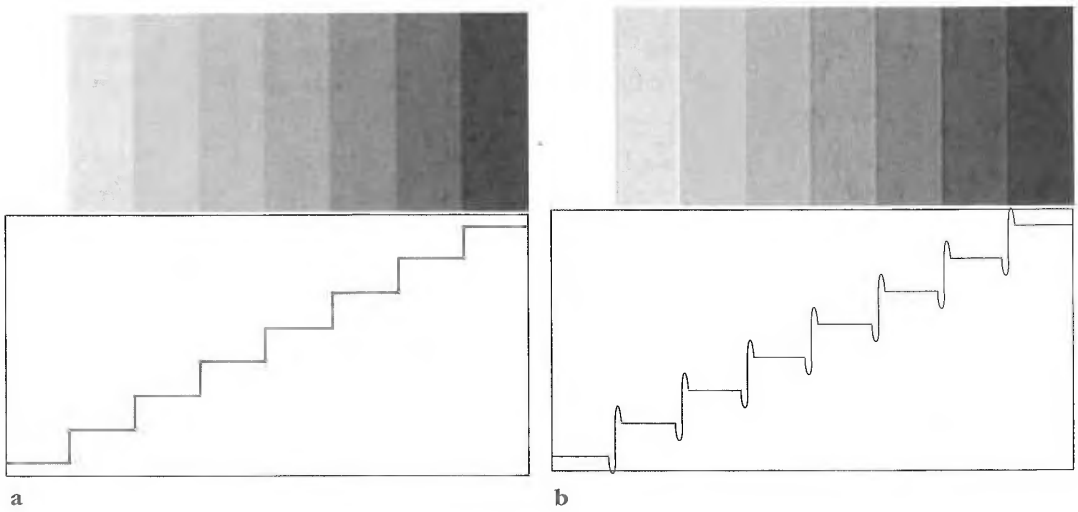


Figure 18. Illustration of Mach bands: (a) *uniform grey bands with the brightness profile of the bands;* ***(b)*** *application of a Laplacian to the bands in image a with the brightness profile, showing increased contrast at the boundaries (this is the way the human eye perceives the uniform bands in image* ***a****.*

Shifting the kernel to each position, multiplying the kernel values times the brightness values, then adding, gives the result

0 0 0 0 –2 0 +2 0 0 0 0

which can be added to the original to give

2 2 2 2 0 4 8 6 6 6 6

The undershoot and overshoot in brightness on either side of the step correspond to the Mach band effect.

Local inhibition also influences our perception of brightness and color. **Figure 19** shows the effect of the surrounding grey scale value on the apparent brightness and size of the central square (the two central grey squares are identical). This also happens in color. **Figure 20** illustrates several effects of local inhibition. In **Figure 20a** the reddish lines appear to change in color, brightness and thickness according to the varying colors of the intervening cyan/blue lines. In **Figure 20b**, the three identical central squares appear to vary in size, color and brightness according to the color of the surrounding square. In **Figure 20c**, the size of the two central squares appears different, and the color of the central square appears brighter and more saturated than when the same color occupies the surrounding outer patch. Similar visual illusions based on inhibition are commonly shown in many books and articles on vision and perception.

Inhibition is a process that takes place within the eye itself. Connections between the neurons within the retina suppress the output from one region according to its neighbors. Inhibition is useful for edge detection, but also affects our visual comparisons of size and orientation. **Figure 21** shows a typical result from an experiment in which the output from a single retinal ganglion in the optic nerve (shown as a time sequence across the top) is plotted as a small white target is slowly moved across the visual field. The cluster of white dots shows the location of the corresponding receptor on the retina. Notice the zone around the receptor that is very poor in white dots. Light shining on neighboring receptors inhibits the output from the central receptor. Presenting a dark spot in the visual field produces additional output from the receptor when it lies within the zone of inhibition, and suppresses output when it lies on the receptor. Selecting different ganglia will map other receptor cells, including those that surround the one shown. Each cell inhibits output from its neighbors. Other ganglia show temporal inhibition, in which output from a cell ignores the steady presence of either light or dark, and responds only to changes in illumination. You can demonstrate this effect by staring at a bright red pattern and then looking away; the afterimage will appear in your eye as green.

Another way to describe the operation of the Laplacian is as a high-pass filter. In Chapter 5, image processing in the Fourier domain is discussed in terms of the high- and low-frequency components of the image brightness. A low-pass filter, such as the smoothing kernels discussed in Chapter 3, removes the high-frequency variability associated with random noise, which can cause nearby pix-

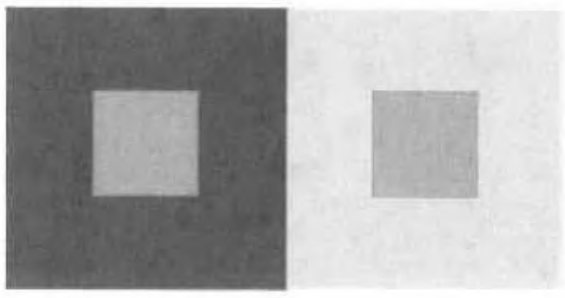


Figure 19. Visual inhibition causes the brightness of the central grey square to appear different depending on the brightness of the surroundings.

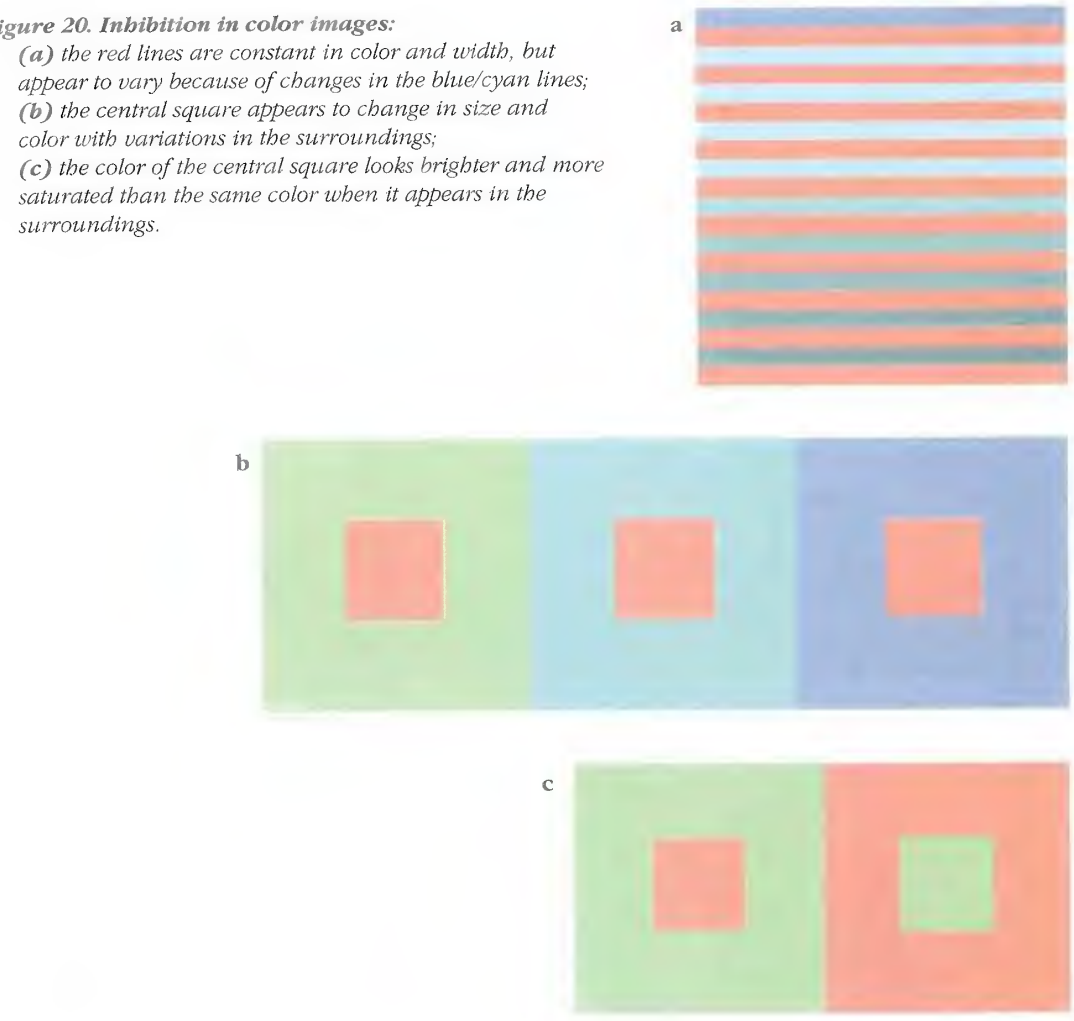


Figure 20. Inhibition in color images:
(a) the red lines are constant in color and width, but appear to vary because of changes in the blue/cyan lines; (b) the central square appears to change in size and color with variations in the surroundings; (c) the color of the central square looks brighter and more saturated than the same color when it appears in the surroundings.

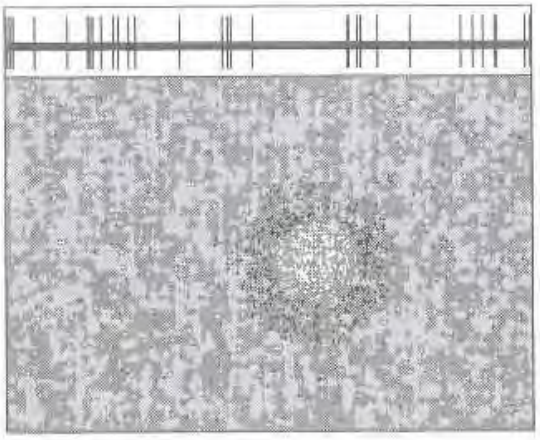


Figure 21. *Experimental results plotting the output from a single retinal ganglion in the optic nerve as targets consisting of small white or black circles are moved slowly across the visual field. The central cluster of white dots shows a positive response to the white circle, and the reduction in pulses when the white target lies in the field of a neighboring receptor. The dark target gives a complementary result.*

els to vary in brightness. Conversely, a high-pass filter allows these high frequencies to remain (pass through the filter), while removing the low frequencies corresponding to the gradual overall variation in brightness.

As with smoothing kernels, there are many different sets of integers and different-size kernels that can be used to apply a Laplacian to an image. The simplest just uses the four immediately touching pixels that share a side with the central pixel

$$\begin{matrix} & -1 & \\ -1 & +4 & -1 \\ & -1 & \end{matrix}$$

Larger kernels may combine a certain amount of smoothing by having positive weights for pixels in a small region near the central pixel, surrounded by negative weights for pixels farther away. Several examples are shown below. Because of the shape of these kernels when they are plotted as isometric views (**Figure 22**), they are sometimes described as a Mexican hat or sombrero filter; this name is usually reserved for kernels with more than one positive-weighted pixel at the center.

As in the discussion of Gaussian filters in Chapter 3, the restriction of these kernels to integer values is arbitrary based on the historic cost of computing with real (floating point) values. By using real numbers it is possible to construct smoother kernels that better fit the ideal functional shapes, which are closely approximate the difference of two Gaussians of different standard deviation, as discussed in the next section.

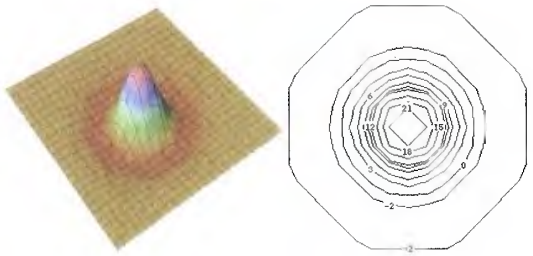


Figure 22. *Isometric and contour plot views of the 17 × 17 Mexican hat kernel.*

5 × 5 Laplacian kernel

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

7 × 7 Mexican hat kernel

0	0	-1	-1	-1	0	0
0	-1	-3	-3	-3	-1	0
-1	-3	0	7	0	-3	-1
-1	-3	7	24	7	-3	-1
-1	-3	0	7	0	-3	-1
0	-1	-3	-3	-3	-1	0
0	0	-1	-1	-1	0	0

13 × 13 Mexican hat kernel

0	0	0	0	0	-1	-1	-1	0	0	0	0	0
0	0	0	-1	-1	-2	-2	-2	-1	-1	0	0	0
0	0	-2	-2	-3	-3	-4	-3	-3	-2	-2	0	0
0	-1	-2	-3	-3	-3	-2	-3	-3	-3	-2	-1	0
0	-1	-3	-3	-1	4	6	4	-1	-3	-3	-1	0
-1	-2	-3	-3	4	14	19	14	4	-3	-3	-2	-1
-1	-2	-4	-2	6	19	24	19	6	-2	-4	-2	-1
-1	-2	-3	-3	4	14	19	14	4	-3	-3	-2	-1
0	-1	-3	-3	-1	4	6	4	-1	-3	-3	-1	0
0	-1	-2	-3	-3	-3	-2	-3	-3	-3	-2	-1	0
0	0	-2	-2	-3	-3	-4	-3	-3	-2	-2	0	0
0	0	0	-1	-1	-2	-2	-2	-1	-1	0	0	0
0	0	0	0	0	-1	-1	-1	0	0	0	0	0

17 × 17 Mexican hat kernel

0	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0
0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0
0	0	-1	-1	-1	-2	-3	-3	-3	-3	-3	-2	-1	-1	-1	0	0
0	0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-2	-1	-1	0	0
0	-1	-1	-2	-3	-3	-3	-2	-3	-2	-3	-3	-3	-2	-1	-1	0
0	-1	-2	-3	-3	-3	0	2	4	2	0	-3	-3	-3	-2	-1	0
-1	-1	-3	-3	-3	0	4	10	12	10	4	0	-3	-3	-3	-1	-1
-1	-1	-3	-3	-2	2	10	18	21	18	10	2	-2	-3	-3	-1	-1
-1	-1	-3	-3	-3	4	12	21	24	21	12	4	-3	-3	-3	-1	-1
-1	-1	-3	-3	-2	2	10	18	21	18	10	2	-2	-3	-3	-1	-1
-1	-1	-3	-3	-3	0	4	10	12	10	4	0	-3	-3	-3	-1	-1
0	-1	-2	-3	-3	-3	0	2	4	2	0	-3	-3	-3	-2	-1	0
0	-1	-1	-2	-3	-3	-3	-2	-3	-2	-3	-3	-3	-2	-1	-1	0
0	0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-2	-1	-1	0	0
0	0	-1	-1	-1	-2	-3	-3	-3	-3	-3	-2	-1	-1	-1	0	0
0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0
0	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0

Derivatives

The Laplacian is a high-pass filter, but not a particularly good tool for demarcating edges (Berzins, 1984; Heath et al., 1997). In most cases, boundaries or edges of features or regions appear at least locally as a step in brightness, sometimes spread over several pixels. The Laplacian gives a larger response to a line than to a step, and to a point than to a line. In an image that contains noise, which is typically present as points varying in brightness due to counting statistics, detector characteristics, etc., the Laplacian will show such points more strongly than the edges or boundaries that are of interest.

Another approach to locating edges uses first derivatives in two or more directions. It will be helpful to first examine simple, one-dimensional first derivatives. Some images are essentially one-dimensional, such as chromatography preparations in which proteins are spread along lanes in an electrical field (**Figure 23**) or tree ring patterns from drill cores (**Figure 24**). Applying a first derivative to such an image, in the direction of important variation, demarcates the boundaries and enhances the visibility of small steps and other details, as shown in the figures. Of course, for an

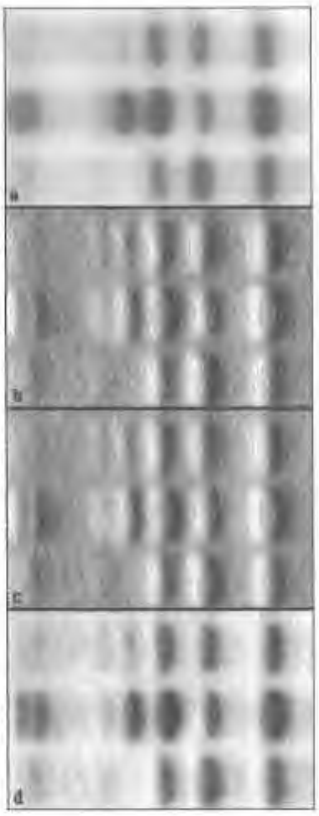


Figure 23. Image of a protein separation gel:
(a) *original;*
(b) *difference between each pixel and its neighbor to the left;*
(c) *vertical averaging of image* ***b*** *to reduce noise;*
(d) *horizontal derivative using a 3 × 3 kernel as discussed in the text.*

Figure 24. Image of tree rings in a drill core:
(a) *original;*
(b) *difference between each pixel and its neighbor to the left;*
(c) *vertical averaging of image* ***b*** *to reduce noise;*
(d) *horizontal derivative using a 3 × 3 kernel as discussed in the text.*



image with digitized finite pixels, a continuous derivative cannot be performed. Instead, the difference value between adjacent pixels can be calculated as a finite derivative. This difference is somewhat noisy, but averaging in the direction perpendicular to the derivative can smooth the result, as shown in the figures.

A derivative image with smoother appearance can be produced with fewer steps by applying an asymmetric kernel. Consider a set of kernels of the form shown below. Eight rotational orientations of this kernel about the center are possible.

1	0	−1	2	1	0	1	2	1
2	0	−2	1	0	−1	0	0	0 ...
1	0	−1	0	−1	−2	−1	−2	−1

Applying the first pattern of values shown to the tree ring and protein separation images is shown in **Figures 23** and **24**.

One improvement comes from averaging together the adjacent pixels in each vertical column before taking the difference; this reduces noise in the image. A second, more subtle effect is that these kernels are 3 pixels wide and thus replace the central pixel with the difference value. The simple subtraction described previously causes a half-pixel shift in the image, which is absent with this kernel. Finally, the method shown here is fast, requiring only a single pass through the image.

Obviously, other kernel values can be devised that will also produce derivatives. As the kernel size increases, more different directions are possible. Because this is fundamentally a one-dimensional derivative, it is possible to directly use the coefficients of Savitsky and Golay (1964), which were originally published for use with such one-dimensional data as spectrograms or other strip-chart recorder output. These coefficients, like the smoothing weights shown in Chapter 3, are equivalent to least-squares fitting a high-order polynomial to the data. In this case, however, the first derivative of the polynomial is evaluated at the central point. Both second degree (quadratic) and fourth degree (quartic) polynomials are shown in **Tables 1** and **2**.

Figure 25 shows a fragment of the brightness profile along the center of the tree ring image in **Figure 24**. The first derivative plots shown were obtained by using the pixel difference approximation, and by using these weights to fit quadratic and quartic polynomials over widths of 5, 9, and 13 pixels around each point. The fits produce much smoother results and suppress noise; however, by increasing the fitting width, the small-scale real variations in the data are suppressed. It is generally necessary, as in any fitting or smoothing operation, to keep the kernel size smaller than the features of interest.

Using one-dimensional derivatives to extract one-dimensional data from two-dimensional images is a relatively specialized operation, although extending the same principles to locating boundaries with arbitrary orientations in two-dimensional images is one of the most common of all image enhancement operations. The problem, of course, is finding a method that is insensitive to the (local) orientation of the edge.

One of the earliest approaches to this task was the Roberts' Cross operator (Roberts, 1965). It uses the same difference technique shown above for the one-dimensional case, but with two pixel differences at right angles to each other, as diagrammed in **Figure 26**. These two differences represent a finite approximation to the derivative of brightness. Two-directional derivatives can be combined to obtain a magnitude value that is insensitive to the orientation of the edge by squaring, adding, and taking the square root of the total.

Table 1. Coefficients for First Derivative Quadratic Fit

No.	5	7	9	11	13	15	17	19	21	23	25
−12											−.0092
−11										−.0109	−.0085
−10									−.0130	−.0099	−.0077
−9								−.0158	−.0117	−.0089	−.0069
−8							−.0196	−.0140	−.0104	−.0079	−.0062
−7						−.0250	−.0172	−.0123	−.0091	−.0069	−.0054
−6					−.0330	−.0214	−.0147	−.0105	−.0078	−.0059	−.0046
−5				−.0455	−.0275	−.0179	−.0123	−.0088	−.0065	−.0049	−.0038
−4			−.0667	−.0364	−.0220	−.0143	−.0098	−.0070	−.0052	−.0040	−.0031
−3		−.1071	−.0500	−.0273	−.0165	−.0107	−.0074	−.0053	−.0039	−.0030	−.0023
−2	−.2000	−.0714	−.0333	−.0182	−.0110	−.0071	−.0049	−.0035	−.0026	−.0020	−.0015
−1	−.1000	−.0357	−.0250	−.0091	−.0055	−.0036	−.0025	−.0018	−.0013	−.0010	−.0008
0	0	0	0	0	0	0	0	0	0	0	0
+1	+.1000	+.0357	+.0250	+.0091	+.0055	+.0036	+.0025	+.0018	+.0013	+.0010	+.0008
+2	+.2000	+.0714	+.0333	+.0182	+.0110	+.0071	+.0049	+.0035	+.0026	+.0020	+.0015
+3		+.1071	+.0500	+.0273	+.0165	+.0107	+.0074	+.0053	+.0039	+.0030	+.0023
+4			+.0667	+.0364	+.0220	+.0143	+.0098	+.0070	+.0052	+.0040	+.0031
+5				+.0455	+.0275	+.0179	+.0123	+.0088	+.0065	+.0049	+.0038
+6					+.0330	+.0214	+.0147	+.0105	+.0078	+.0059	+.0046
+7						+.0250	+.0172	+.0123	+.0091	+.0069	+.0054
+8							+.0196	+.0140	+.0104	+.0079	+.0062
+9								+.0158	+.0117	+.0089	+.0069
+10									+.0130	+.0099	+.0077
+11										+.0109	+.0085
+12											+.0092

This method has the same problems as the difference method used in one dimension. Noise in the image is magnified by the single-pixel differences, and the result is shifted by half a pixel in both the x and y directions. In addition, the result is not invariant with respect to edge orientation. As a practical matter, the computers in common use when this model was first proposed were not very fast, nor were they equipped with separate floating-point math coprocessors. This made the square root of the sum of the squares impractical to calculate. Two alternatives were used: adding the absolute values of the two directional differences, or comparing the two absolute values of the differences and keeping the larger one. Both of these methods make the result quite sensitive to direction. In addition, even if the square root method is used, the magnitude of the result will vary because the pixel spacing is not the same in all directions, and edges in the vertical and horizontal directions spread the change in brightness over more pixels than edges in the diagonal directions.

In several of the comparison sequences that follow, the Roberts' Cross image will be shown (**Figures 29b** and **44c**). In all cases, the square root of the sum of squares of the differences was used. Even so, the images are characterized by varying sensitivity with respect to edge orientation, as well as a high noise sensitivity.

The Sobel and Kirsch operators

Just as for the example of the horizontal derivative described earlier, the use of a larger kernel offers reduced sensitivity to noise by averaging several pixels and eliminating image shift. In fact, the

Table 2. Coefficients for First Derivative Quartic Fit

No.	5	7	9	11	13	15	17	19	21	23	25
−12											+.0174
−11										+.0200	+.0048
−10									+.0231	+.0041	−.0048
−9								+.0271	+.0028	−.0077	−.0118
−8							+.0322	−.0003	−.0119	−.0159	−.0165
−7						+.0387	−.0042	−.0182	−.0215	−.0209	−.0190
−6					+.0472	−.0123	−.0276	−.0292	−.0267	−.0231	−.0197
−5				+.0583	−.0275	−.0423	−.0400	−.0340	−.0280	−.0230	−.0189
−4			+.0724	−.0571	−.0657	−.0549	−.0431	−.0335	−.0262	−.0208	−.0166
−3		+.0873	−.1195	−.1033	−.0748	−.0534	−.0388	−.0320	−.0219	−.0170	−.0134
−2	+.0833	−.2659	−.1625	−.0977	−.0620	−.0414	−.0289	−.0210	−.0157	−.0120	−.0094
−1	−.6667	−.2302	−.1061	−.0575	−.0346	−.0225	−.0154	−.0110	−.0081	−.0062	−.0048
0	0	0	0	0	0	0	0	0	0	0	0
+1	+.6667	+.2302	+.1061	+.0575	+.0346	+.0225	+.0154	+.0110	+.0081	+.0062	+.0048
+2	−.0833	+.2659	+.1625	+.0977	+.0620	+.0414	+.0289	+.0210	+.0157	+.0120	+.0094
+3		−.0873	+.1195	+.1033	+.0748	+.0534	+.0388	+.0320	+.0219	+.0170	+.0134
+4			−.0724	+.0571	+.0657	+.0549	+.0431	+.0335	+.0262	+.0208	+.0166
+5				−.0583	+.0275	+.0423	+.0400	+.0340	+.0280	+.0230	+.0189
+6					−.0472	+.0123	+.0276	+.0292	+.0267	+.0231	+.0197
+7						−.0387	+.0042	+.0182	+.0215	+.0209	+.0190
+8							−.0322	+.0003	+.0119	+.0159	+.0165
+9								−.0271	−.0028	+.0077	+.0118
+10									−.0231	−.0041	+.0048
+11										−.0200	−.0048
+12											−.0174

derivative kernels shown above, or other similar patterns using different sets of integers, are widely used. Some common examples of these coefficients are shown in **Table 3** below.

Table 3. Examples of derivative filter weights

+1	0	−1	+1	0	−1	+1	−1	−1	+5	−3	−3
+1	0	−1	+2	0	−2	+2	+1	−1	+5	0	−3
+1	0	−1	+1	0	−1	+1	−1	−1	+5	−3	−3
+1	+1	0	+2	+1	0	+2	+1	−1	+5	+5	−3
+1	0	−1	+1	0	−1	+1	+1	−1	+5	0	−3
0	−1	−1	0	−1	−2	−1	−1	−1	−3	−3	−3
+1	+1	+1	+1	+2	+1	+1	+2	+1	+5	+5	+5
0	0	0	0	0	0	−1	+1	−1	−3	0	−3
−1	−1	−1	−1	−2	−1	−1	−1	−1	−3	−3	−3

and so forth for eight rotations.

It actually makes little difference which of these patterns of values is used, as long as the magnitude of the result does not exceed the storage capacity of the computer being used. This is often

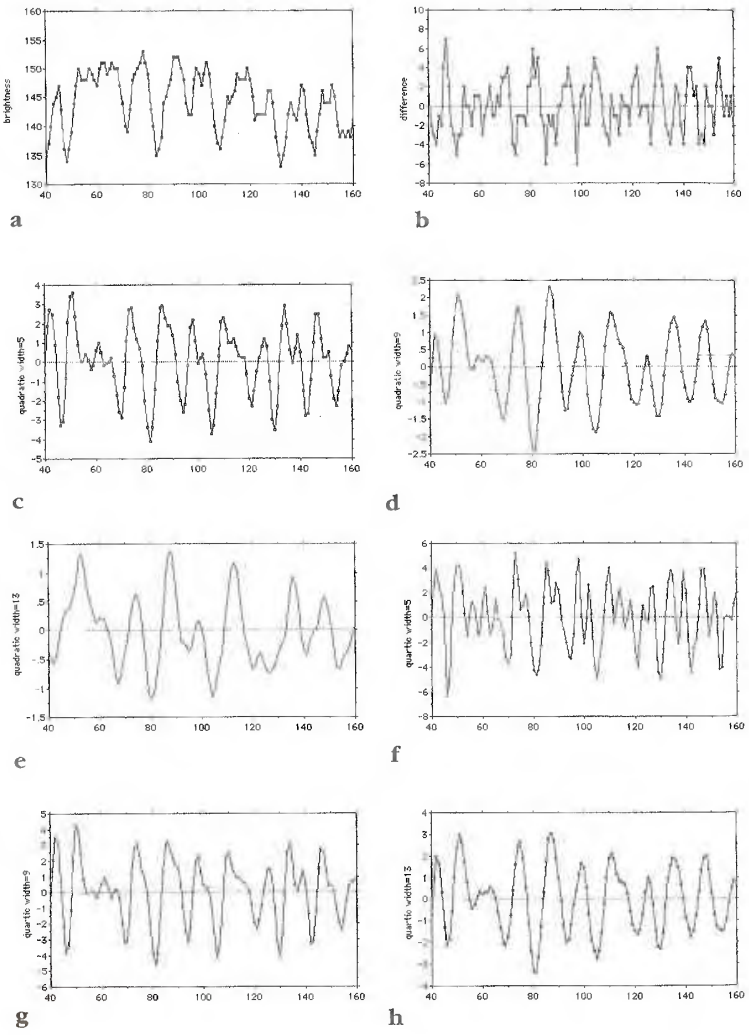


Figure 25. Horizontal brightness profile for a line in the center of the tree ring image (Figure 25), and its derivatives: *(a) original brightness plotted for each pixel along a fragment of the line; (b) finite difference plot showing the difference between adjacent pixels; (c) derivative calculated using the weights for a quadratic polynomial fit to 5 points; (d) a quadratic fit to 9 points; (e) a quadratic fit to 13 points; (f) derivative calculated using the weights for a quartic polynomial fit to 5 points; (g) a quartic fit to 9 points; (h) a quartic fit to 13 points.*

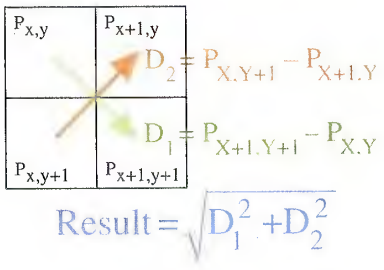


Figure 26. *Diagram of Roberts' Cross operator. Two differences in directions at right angles to each other are combined to determine the gradient.*

a single byte per pixel, which, because the result of the previous operation may be either negative or positive, can handle only values between –127 and +128. If large steps in brightness are present, this may result in the clipping of the calculated values, so that major boundaries are broadened or distorted in order to see smaller ones. The alternative is to employ automatic scaling, using the maximum and minimum values in the derivative image to set the white and dark values. To avoid loss of precision, this requires two passes through the image: one to perform the calculations and find the extreme values and the second to actually compute the values and scale the results for storage, or one to perform the calculations and temporarily store intermediate results with full precision, and a second to rescale the values to fit the range for pixel values.

As for the Roberts' Cross method, if the derivatives in two orthogonal directions are computed, they can be combined as the square root of the sums of their squares to obtain a result independent of orientation.

$$\text{Magnitude} = \sqrt{\left(\frac{\partial B}{\partial x}\right)^2 + \left(\frac{\partial B}{\partial y}\right)^2} \tag{6}$$

This is the Sobel (1970) method. It is one of the most commonly used techniques, even though it requires a modest amount of computation to perform correctly. (As for the Roberts' Cross, some computer programs attempt to compensate for hardware limitations by adding or comparing the two values, instead of squaring, adding, and taking the square root.)

With appropriate hardware, such as a shift register or array processor, the Sobel operation can be performed in essentially real time. This usually means 1/30th of a second per image, so that conventional video images can be processed and viewed. It often means viewing one frame while the following one is digitized, but in the case of the Sobel, it is even possible to view the image live, delayed only by two video scan lines. Two lines of data can be buffered and used to calculate the derivative values using the 3 × 3 kernels shown previously. Specialized hardware to perform this real-time edge enhancement is used in some military applications, making it possible to locate edges in images for tracking, alignment of hardware in midair refueling, and other purposes.

At the other extreme, some general-purpose image analysis systems that do not have hardware for fast math operations perform the Sobel operation using a series of operations. First, two derivative images are formed, one using a horizontal and one a vertical orientation of the kernel. Then each of these is modified using a LUT to replace the value of each pixel with its square. The two resulting images are added together, and another LUT is used to convert each pixel value to its square root. No multiplication or square roots are needed. However, if this method is applied in a typical system with 8 bits per pixel, the loss of precision is severe, reducing the final image to no more than 4 bits (16 grey levels) of useful information.

A more practical way to avoid the mathematical operations needed to calculate the square root of the sum of squares needed by the Sobel is the Kirsch operator (1971). This method applies each of the eight orientations of the derivative kernel and keeps the maximum value. It requires only integer multiplication and comparisons. For many images, the results for the magnitude of edges are very similar to the Sobel. **Figure 27** shows an example. Vertical and horizontal derivatives of the image, and the maximum derivative values in each of eight directions, are shown.

Figures 28 and **29** illustrate the formation of the Sobel edge-finding image and compare it to the Laplacian, Roberts' Cross, and Kirsch operators. The example image contains many continuous edges running in all directions around the holes in the carbon film, as well as some straight edges at various orientations along the asbestos fibers. The Laplacian image is quite noisy, and the Roberts' Cross does not show all of the edges equally well. The individual vertical and horizontal

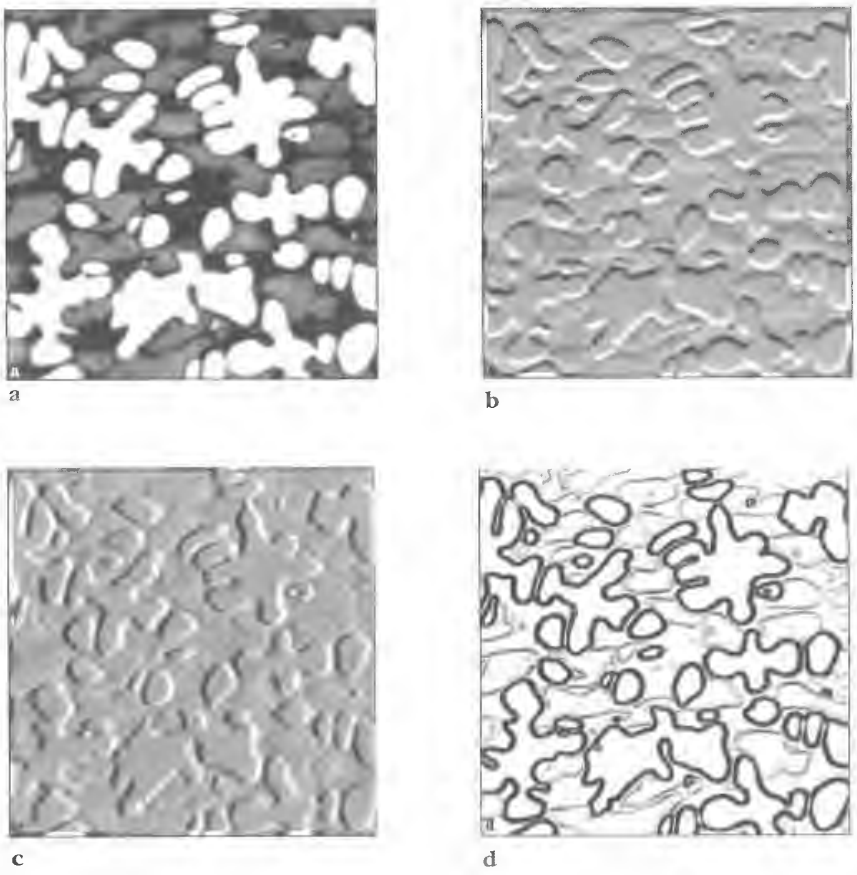


Figure 27. *A metallographic image* ***(a)*** *with two directional derivatives* ***(b*** *and* ***c)****, and the Kirsch image* ***(d)*** *produced by keeping the maximum value from each direction.*

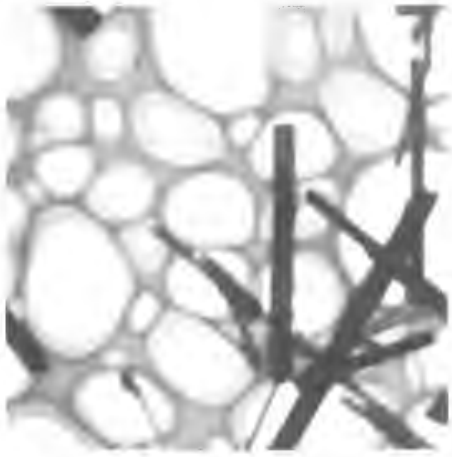


Figure 28. *Original image (asbestos fibers on a holey carbon film, imaged in a TEM). This image is the basis for the processing shown in* ***Figure 29****.*

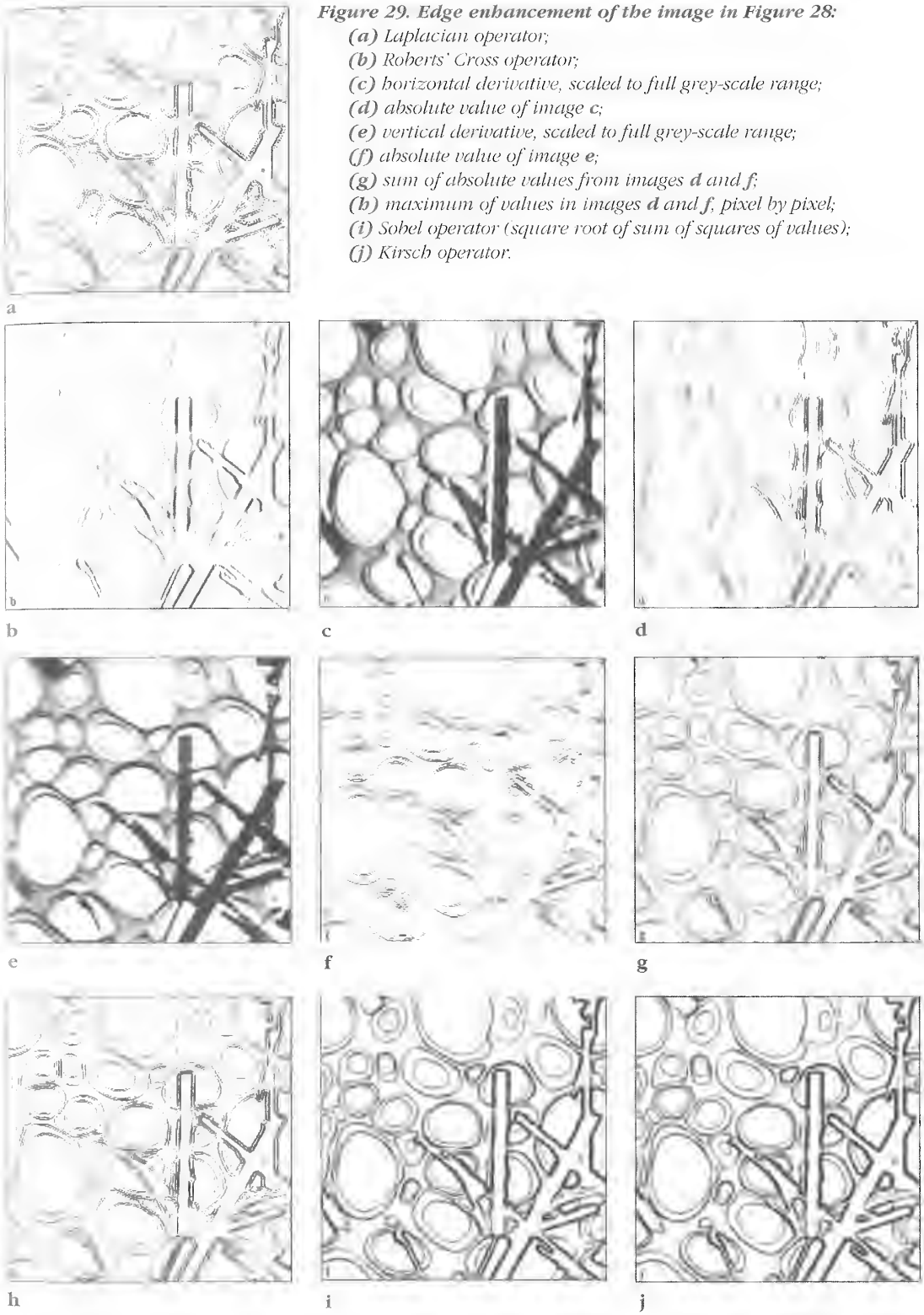


Figure 29. Edge enhancement of the image in Figure 28:
(a) *Laplacian operator;*
(b) *Roberts' Cross operator;*
(c) *horizontal derivative, scaled to full grey-scale range;*
(d) *absolute value of image* ***c****;*
(e) *vertical derivative, scaled to full grey-scale range;*
(f) *absolute value of image* ***e****;*
(g) *sum of absolute values from images* ***d*** *and* ***f****;*
(h) *maximum of values in images* ***d*** *and* ***f****, pixel by pixel;*
(i) *Sobel operator (square root of sum of squares of values);*
(j) *Kirsch operator.*

derivatives are shown with the zero value shifted to an intermediate grey, so that both the negative and positive values can be seen. The absolute values are also shown.

Combining the two directional derivatives by a sum, or maximum operator, produces quite noisy and incomplete boundary enhancement. The square root of the sum of squares produces a good image, with little noise and continuous edge markings. The result from the Kirsch operator is very similar to the Sobel for this image.

In addition to the magnitude of the Sobel operator, it is also possible to calculate a direction value (Lineberry, 1982) for each pixel as:

$$\text{Direction} = \text{Arc Tan}\left(\frac{\partial B / \partial y}{\partial B / \partial x}\right) \tag{7}$$

This assigns a value to each pixel for the gradient direction, which can be scaled to the grey scale of the image. **Figure 30** shows the vector results from applying the Sobel operator to the image of **Figure 29**. The magnitude and direction are encoded, but the vector field is too sparse to show any image details.

Figure 31 shows only the direction information for the image in **Figure 27**, using grey values to represent the angles. The progression of values around each more-or-less circular hole is evident. The use of a pseudo-color scale for this display is particularly suitable, since a rainbow of hues can show the progression without the arbitrary discontinuity required by the grey scale (in this example, at an angle of zero degrees). Unfortunately, the pixels within relatively homogeneous areas of the features also have colors assigned, because at every point there is some direction to the gradient, and these colors tend to overwhelm the visual impression of the image. A solution is to combine the magnitude of the gradient with the direction, as shown in **Figure 32**. This is the dendrite image from **Figure 28**, processed to show the magnitude and the direction of the Sobel edge gradient, the latter in color. These two images are then multiplied together so that the intensity of the color is proportional to the magnitude of the gradient. The result clearly shows the orientation of boundaries.

The magnitude and direction information can also be combined in an HSI representation of the image. In **Figure 33**, the original grey-scale image (the SEM image from **Figure 9**) has been processed to obtain the Sobel magnitude and direction. The direction information is then assigned to the hue plane and the magnitude to saturation and intensity.

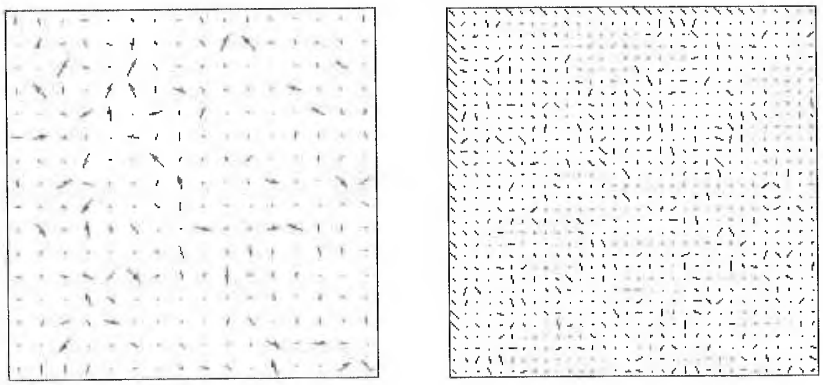


Figure 30. *Applying a Sobel operator to the image in* ***Figure 27****. Each vector has the direction and magnitude given by the operator, but even the fine vector field is too sparse to show any details of the image.*

*Figure 31. Direction of the Sobel operator for the image in **Figure 28**.*



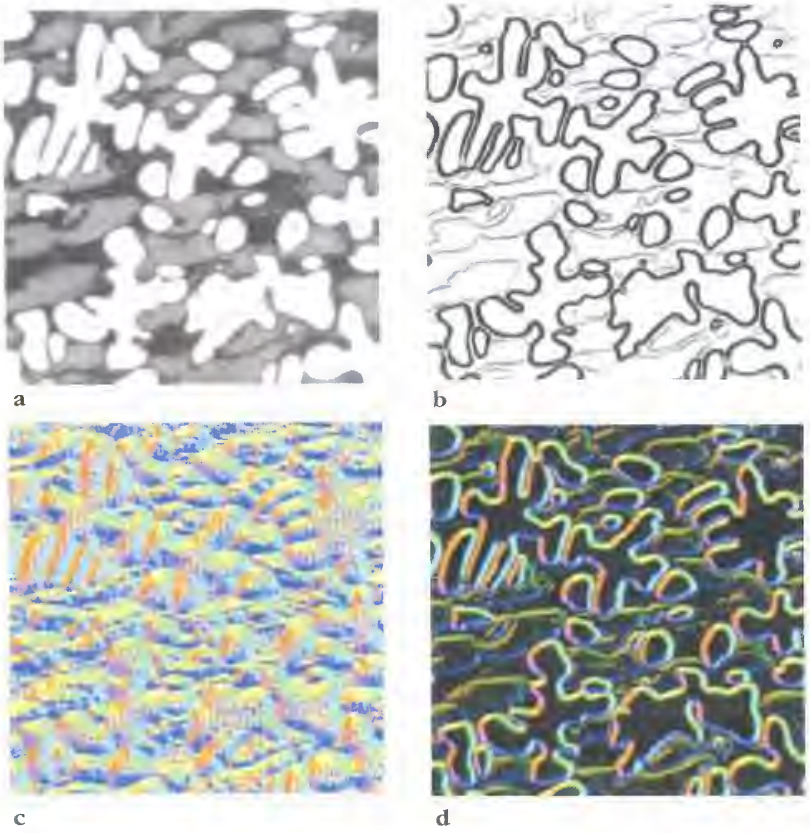
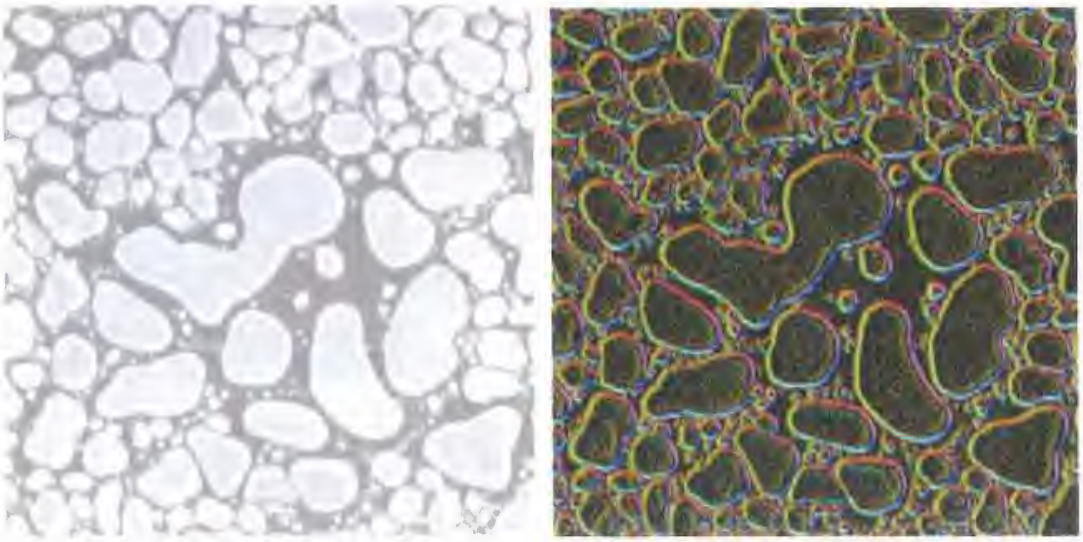


*Figure 32. **Combining the magnitude and direction information from the Sobel gradient operator:***
***(a)** original grey scale image (from **Figure 28**);*
***(b)** Sobel gradient magnitude;*
***(c)** Sobel direction (color coded);*
***(d)** product of **b** times **c**.*

Another way to improve the resulting image is to show the direction information only for those pixels which also have a strong magnitude for the brightness gradient. In **Figure 34**, this is done by using the magnitude image as a mask, selecting (by thresholding as discussed in Chapter 6) the 40% of the pixels with the largest gradient magnitude, and showing the direction only for those



a b

Figure 33. HSI representation of the gradient vector, with hue representing direction and the magnitude shown by saturation and intensity: *(a) original image;* ***(b)*** *result.*

pixels. This is particularly suitable for selecting pixels to be counted as a function of color (direction) for analysis purposes.

Figure 34 also shows a histogram plot of the preferred orientation in the image. This kind of plot is particularly common for interpreting the orientation of lines, such as dislocations in metals and the traces of faults in geographic maps. In **Figure 35**, the orientation of the collagen fibers is measured using this same technique. The plot shows that they are not isotropically oriented, with about twice as many pixels showing an orientation of about 70 degrees. Further examples of

a

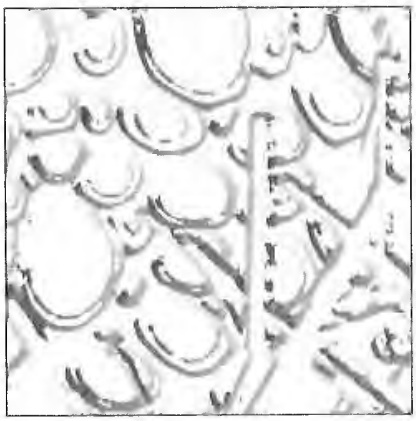
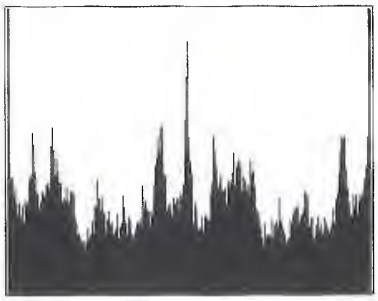


Figure 34. Uses of the edge-orientation image in Figure 31:
(a) *masking only those pixels whose edge magnitude is large (the 40% of the pixels with the largest magnitude);* ***(b)*** *generating a histogram of orientation values, in which the horizontal axis of the histogram (255 brightness values) represents the angular range 0–360°.*

b



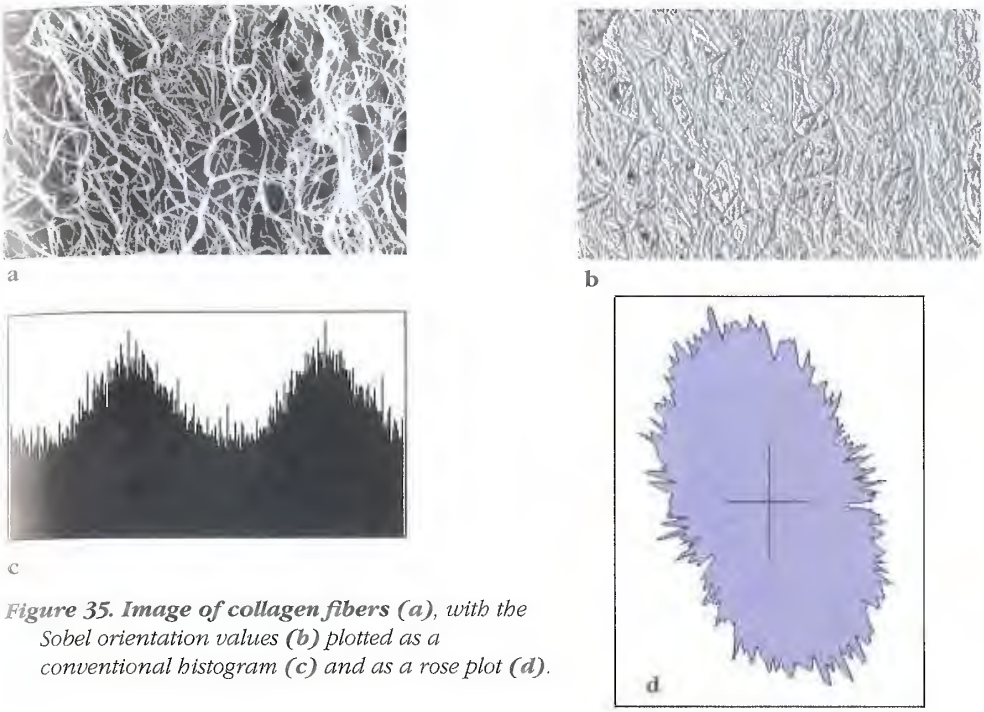


Figure 35. Image of collagen fibers (a), with the Sobel orientation values (b) plotted as a conventional histogram (c) and as a rose plot (d).

measurement using this tool will be discussed below, as will the use of the direction image to reveal image texture, or to select regions of the image based on the texture or its orientation.

When the oriented features do not fill the image, but are either isolated fibers or edges of features, they can be selected for measurement by using the Sobel magnitude image. Thresholding this image to select only the points whose gradient vector is large, and applying the resulting binary image as a selection mask to eliminate pixels that are not of interest for orientation measurement, produces a result as shown in **Figure 36**.

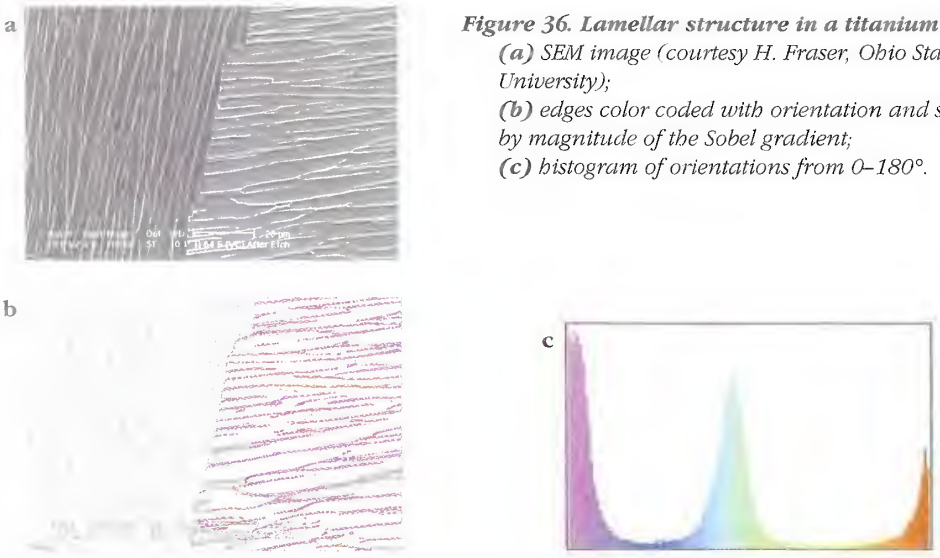


Figure 36. Lamellar structure in a titanium alloy: (a) SEM image (courtesy H. Fraser, Ohio State University); (b) edges color coded with orientation and selected by magnitude of the Sobel gradient; (c) histogram of orientations from 0–180°.

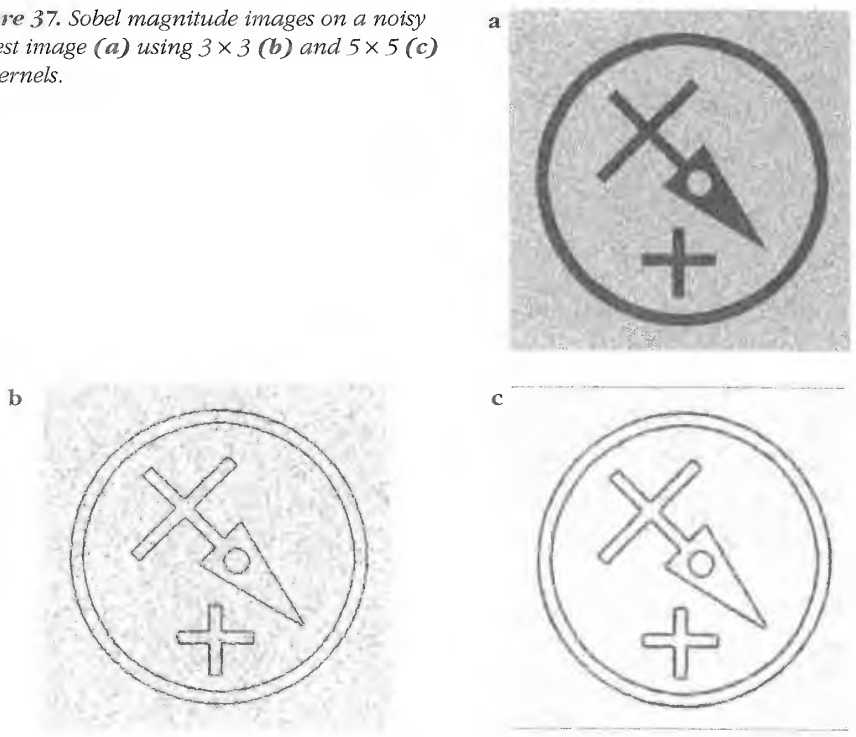


Figure 37. Sobel magnitude images on a noisy test image (a) using 3 × 3 (b) and 5 × 5 (c) kernels.

Orientation determination using the angle calculated from the Sobel derivatives is not a perfectly isotropic or unbiased measure of boundary orientations because of the effects of the square pixel grid and limited number of points sampled. With a larger kernel of values, weights can be assigned to better correspond to how far away the pixels are from the center. With a 5 × 5 array a more smoothly varying and isotropically uniform result can be obtained. **Figure 37** shows an example with a circle and some lines with superimposed image noise as a test object; the definition of the edge using the Sobel magnitude is improved by the larger neighborhood, and the effect of random noise in the image is minimized, although the line breadth becomes greater.

The use of an edge-enhancing operator to modify images is useful in many situations. We have already seen examples of sharpening using the Laplacian, to increase the contrast at edges and make images appear sharper to the viewer. Gradient or edge-finding methods also do this, but they also modify the image so that its interpretation becomes somewhat different. This contrast increase is selective; therefore, it responds to local information in the image in a way that manipulating the brightness histogram cannot.

For example, **Figure 38** shows several views of galaxy M51. In the original image, it is quite impossible to see the full range of brightness, even on a high-quality photographic negative. It is even less possible to print it. The extremely light and dark areas simply cover too great a range. Compressing the range nonlinearly, using a logarithmic transfer function, can make it possible to see both ends of the scale at the same time. However, this is accomplished by reducing small variations, especially at the bright end of the scale, so that they are not visible.

The gradient enhancement shown in **Figure 38c** uses a Sobel operator to mark edges. This shows the structure within the galaxy in a different way, by emphasizing local spatial variations regardless of the absolute value of the brightness. This produces a distinctly different result than the unsharp masking shown in **Figure 38b**.

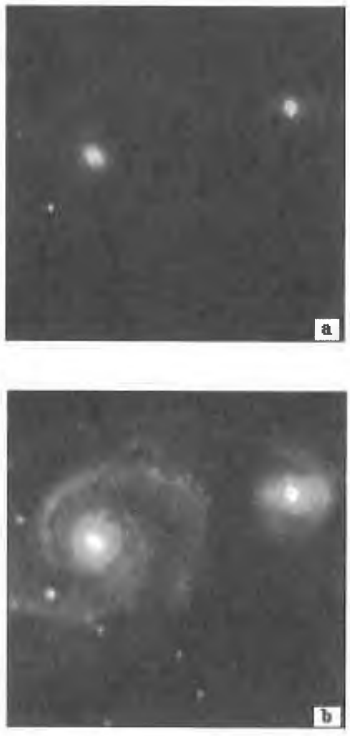
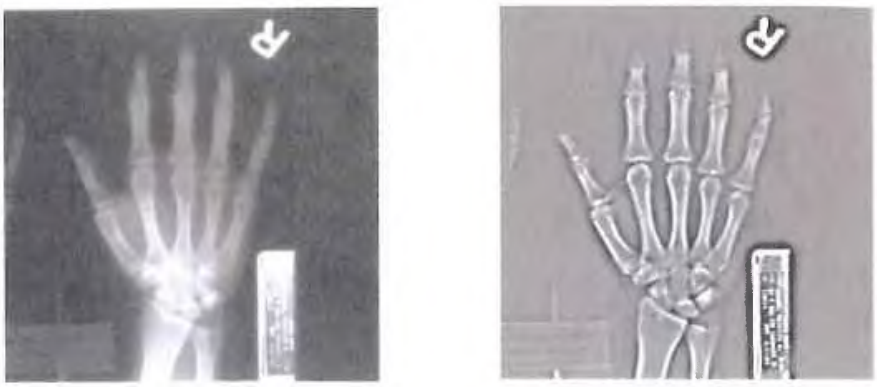


Figure 38. Enhancing an astronomical image (Messier 51):
(a) *original telescope image, with brightness range too great for printing;*
(b) *application of "unsharp masking" by subtracting a smoothed image to reduce contrast selectively and show detail;*
(c) *gradient of original image using a Sobel operator, which also shows the fine structure of the galaxy.*

Unsharp masking has traditionally been applied using photographic darkroom techniques. First, a contact print is made from the original negative onto film, leaving a small gap between the emulsion on the original and that on the film so that the image is blurred. After the film is developed, a new print is made with the two negatives clamped together. The light areas on the original negative are covered by dark areas on the printed negative, allowing little light to come through. Only regions where the slightly out-of-focus negative does not match the original are printed. This is functionally similar to the Laplacian, which subtracts a smoothed (out of focus) image from the original to suppress gradual changes and pass high frequencies or edges.

Applying the unsharp mask operator increases the visibility of fine detail, while suppressing overall variations in brightness. **Figure 39** shows an example using the same x-ray image from **Figure 2**. In the original image the bones in the fingers are thinner and hence not as dense as those in the wrist, and the written label is hardly visible. The processed image makes these more readily visible.

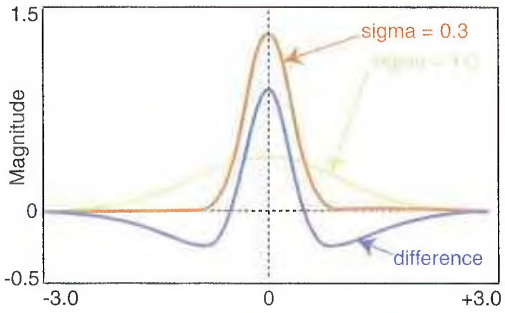
Closely related to unsharp masking, in fact the most general form of this technique, is the subtraction of one smoothed version of the image from another having a different degree of smoothing. This is called the Difference of Gaussians (DOG) method and is believed (Marr, 1982) to be similar to the way the human visual system locates boundaries and other features (the effect of inhibition increasing contrast at boundaries was shown above). Smoothing an image using a Gaussian kernel with an appropriate standard deviation suppresses high-frequency information, which corresponds to small details and spacings that are present. Large structures are not affected. The difference between the two images smoothed with two different Gaussians keeps only those structures (lines, points, etc.) that are in the intermediate size range between the two operators. A plot of two Gaussian curves with different standard deviations, and their difference, is shown in **Figure 40**; the difference is very similar to a cross section of the Laplacian when both



a b

Figure 39. Application of an unsharp mask to an x-ray image of a human hand: (a) *original; (b) processed.*

Figure 40. *The Difference-of-Gaussian (DOG) operator in one dimension. Two Gaussian curves with different standard deviations are shown, with their difference. The result is a sharpening operation much like the Laplacian.*



standard deviations are small, and to the unsharp mask when the smaller one is negligible. This edge extractor is also sometimes called a Marr–Hildreth operator. In terms of the frequencies involved, and for comparison to the Fourier-space filtering discussed in Chapter 5, this is a band pass method. Examples are shown in **Figure 41** for comparison with other methods (some of which are discussed later).

Performing derivative operations using kernels can be considered a template matching or convolution process. The pattern of weights in the kernel is a template that gives the maximum response when it matches the pattern of brightness values in the pixels of the image. The number of different kernels used for derivative calculations indicates that there is no single best definition of what constitutes a boundary. Also, it might be helpful at the same time to look for other patterns that are not representative of an edge.

These ideas are combined in the Frei and Chen algorithm (1977), which applies a set of kernels to each point in the image. Each kernel extracts one kind of behavior in the image, only a few of which are indicative of the presence of an edge. For a 3×3 neighborhood region, the kernels, which are described as orthogonal or independent basis functions, are shown in the following table:

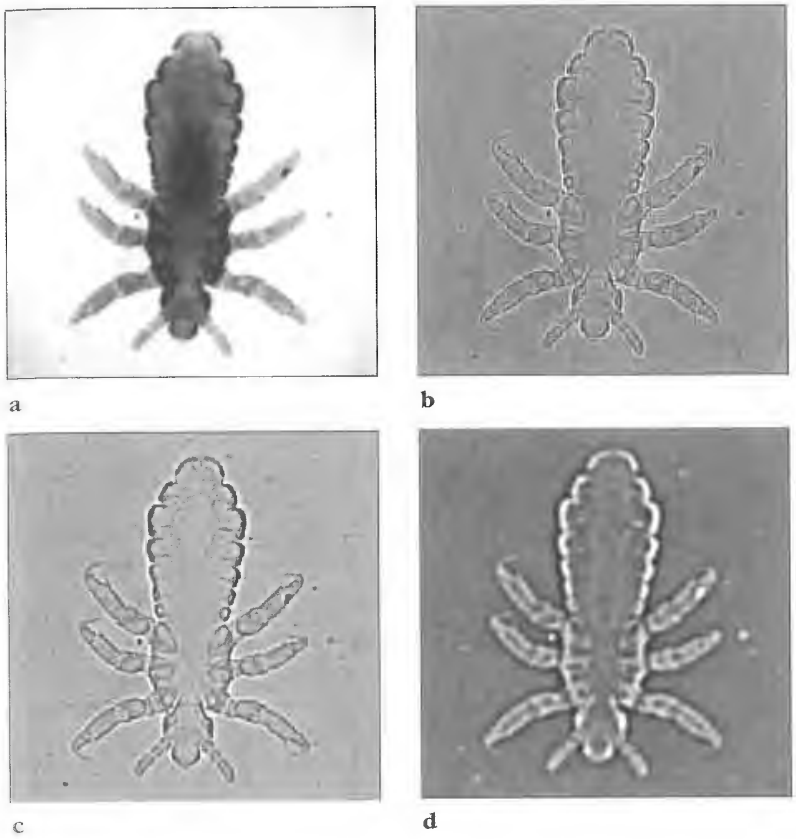


Figure 41. Difference of Gaussians applied to a test image: *(a) original image; (b) original minus 3 × 3 Gaussian smooth; (c) difference between smoothing with σ = 0.625 and σ = 1.0 pixels; (d) difference between smoothing with σ = 1.6 and σ = 1.0 pixels.*

Number	Kernel			Number	Kernel		
0	1	1	1	4	$\sqrt{2}$	−1	0
	1	1	1		−1	0	1
	1	1	1		0	1	$-\sqrt{2}$
1	−1	$-\sqrt{2}$	−1	5	0	1	0
	0	0	0		−1	0	1
	1	$\sqrt{2}$	1		0	−1	0
2	−1	0	1	6	−1	0	1
	$-\sqrt{2}$	0	$\sqrt{2}$		0	0	0
	−1	0	1		1	0	−1
3	0	−1	$\sqrt{2}$	7	1	−2	1
	1	0	−1		−2	4	−2
	$-\sqrt{2}$	1	0		1	−2	1
				8	−2	1	−2
					1	4	1
					−2	1	−2

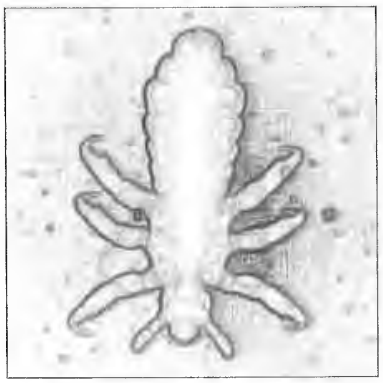


Figure 42. *Application of the Frei and Chen edge detector to the test image from* ***Figure 41a****.*

Only kernels 1 and 2 are considered to indicate the presence of an edge. The results of applying each kernel to each pixel are therefore summed to produce a ratio of the results using kernels 1 and 2 to those for the other kernels. The cosine of the square root of this value is the vector projection of the information from the neighborhood in the direction of "edgeness," and is assigned to the pixel location in the derived image.

The advantage compared to more conventional edge detectors such as the Sobel is sensitivity to a configuration of relative pixel values independent of the magnitude of the brightness, which may vary from place to place in the image. **Figure 42** shows an example of the use of the Frei and Chen operator. This may be compared to other edge-finding processing, shown in **Figures 41** through **44**, which apply some of the other operations discussed in this chapter to the same image.

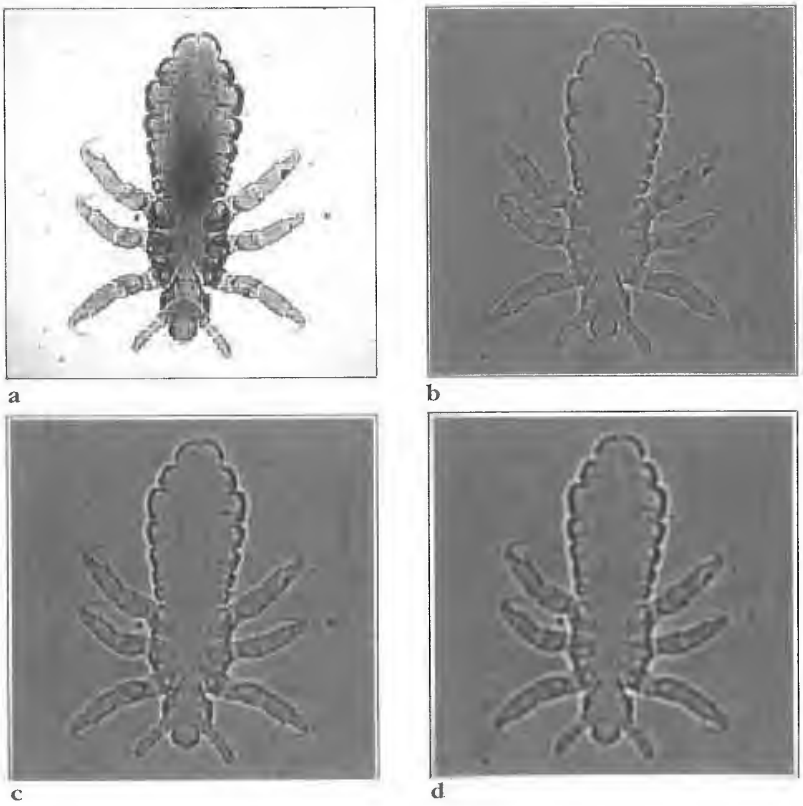


Figure 43. ***Application of Laplacian difference operations to the test image from Figure 41a****:* ***(a)*** *Sharpening (addition of the 3 × 3 Laplacian to the original grey-scale image);* ***(b)*** *5 × 5 Laplacian;* ***(c)*** *7 × 7 Laplacian;* ***(d)*** *9 × 9 Laplacian.*

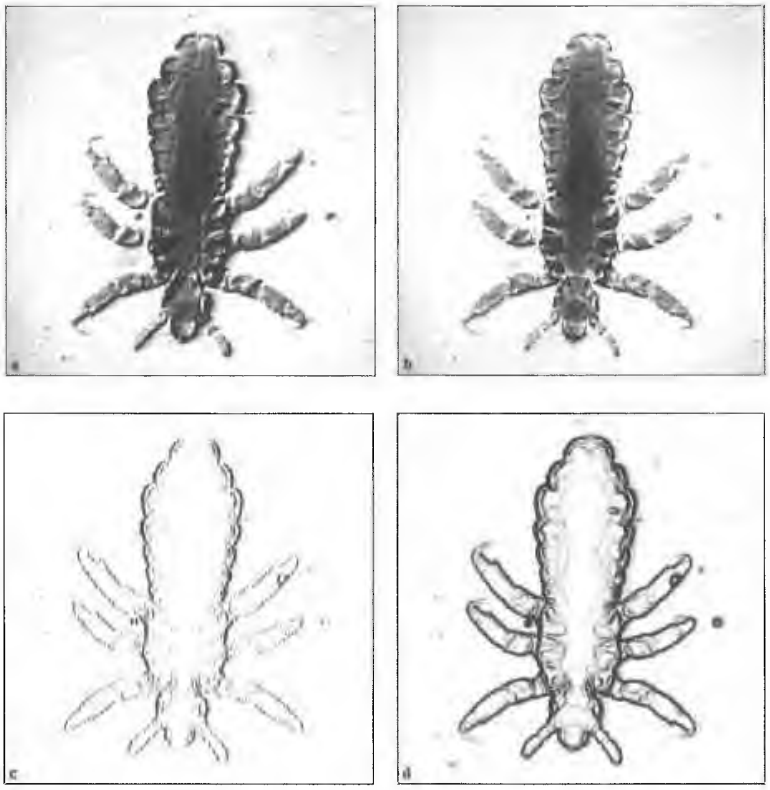


Figure 44: Derivative edge finding applied to the test image from Figure 41a: *(a) single derivative (from the upper left); (b) brightest value at each point after applying two crossed derivatives; (c) Roberts' Cross operator; (d) Sobel operator.*

The noise or blurring introduced by some of the operations is evident, as is the ability of the Frei and Chen procedure to reveal even the rather subtle edge information present in the original. **Figure 43** shows the application of a Laplacian operator, or high-pass filter, using different-sized kernels; note the similarity to the difference of Gaussians results in **Figure 41**. The use of derivatives, including the Roberts' Cross and Sobel methods, is shown in **Figure 44**.

Most of the edge-finding processing operations involve square or circular neighborhoods, with a large number of multiplications and other operations. The Canny filter produces results similar to Kirsch or Sobel operators, but is separable (Canny, 1986; Olsson, 1993). This means that it can be performed in two passes, one of which requires only values from pixels in a horizontal line adjacent to the central pixel, and the other requires only values from pixels in a vertical line. From both a computational point of view, and in terms of the complexity of addressing a large number of neighbor pixels, this offers a significant speed advantage.

The Canny filter is based on finding the zero crossing of the Laplacian of Gaussian (LoG, also called a Marr–Hildreth operator), a bandpass filter that calculates the second derivative of a smoothed image. Because of its shape, this is sometimes called a Mexican hat or sombrero operator. It is very similar in shape and performance to the difference of Gaussians (DOG filter). In the example of a 17 × 17 set of weights shown above, it is presented as a full 2D array of weights, but it can be applied as a separable operation (one-dimensional filters applied to the rows and then to the columns of pixels in an image). **Figure 45** shows the zero crossings of the L.o.G. Typically this operation is performed with several different size filters to isolate edges at different dimensional scales.



Figure 45. *Zero crossings obtained after application of the LoG filter to the image from* ***Figure 41a.***

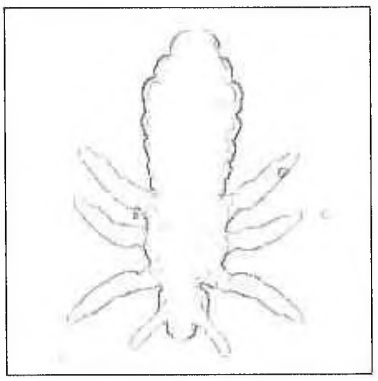


Figure 46. *Canny edge detector applied to the test image from* ***Figure 41a.***

The Canny then adds the additional step of thinning the edge results to single pixel width us a thinning method described below. As shown in **Figure 46**, this is attractive from the standpoint of locating the edge, but tends in many cases to break the line up into discontinuous pieces.

Figure 47 shows an example using a different method. The specimen is a polished aluminum metal examined in the light microscope. The individual grains exhibit different brightnesses because their crystallographic lattices are randomly oriented in space so that the etching procedure used darkens some grains more than others. It is the grain boundaries that are usually important in studying metal structures, since the configuration of grain boundaries results from prior heat treatment and processing and controls many mechanical properties.

The human visual process detects the grain boundaries using its sensitivity to boundaries and edges. Most image analysis systems use a gradient operation, such as a Sobel, to enhance the boundaries prior to measuring them. In the example in **Figure 47**, a statistical method has been employed instead. This is still a local neighborhood operation, but does not use a kernel of weights. The variance operator calculates the sum of squares of the brightness differences for the neighborhood of pixels surrounding each pixel in the original image (in the example, a circular neighborhood with a radius of 2.5 pixels). The variance value, like the other edge-enhancement operators, is very small in uniform regions of the image and becomes large whenever a step is present. In this example, the dark lines (large magnitudes of the variance) are further processed by thinning to obtain the single pixel lines that are superimposed on the original image. This thinning or ridge-finding method is discussed in the next section.

Rank operations

The neighborhood operations discussed in the preceding section use linear arithmetic operations to combine the values of various pixels. Another class of operators that also use neighborhoods

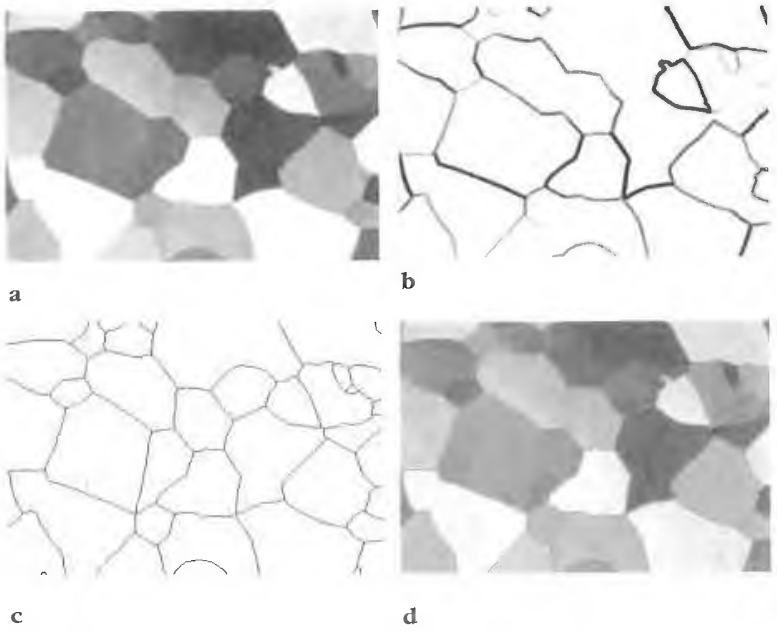


Figure 47. Delineating boundaries between grains: ***(a)*** *Aluminum metal, polished and etched to show different grains (contrast arises from different crystallographic orientation of each grain, so that some boundaries have less contrast than others);* ***(b)*** *Variance edge-finding algorithm applied to image* ***a****;* ***(c)*** *grey-scale skeletonization (ridge finding) applied to* ***b*** *(points not on a ridge are suppressed);* ***(d)*** *thresholding and skeletonization of the boundaries to a single line of pixels produces the grain boundaries, shown superimposed on the original image.*

instead performs comparisons and ranking. In Chapter 3, the median filter was introduced. This sorts the pixels in a region into brightness order, finds the median value, and replaces the central pixel with that value. Used to remove noise from images, this operation completely eliminates extreme values from the image.

Rank operations also include the maximum and minimum operators, which find the brightest or darkest pixels in each neighborhood and place that value into the central pixel. By loose analogy to the erosion and dilation operations on binary images, which are discussed in Chapter 7, these are sometimes called grey scale erosion and dilation (Heijmans, 1991). The erosion effect of this ranking operation was demonstrated in Chapter 3, in the context of removing features from an image to produce a background for leveling.

Rank operators can be used to produce an edge-enhancement filter conceptually similar in some ways to a Sobel filter. The horizontal "derivative" is calculated as the difference between the median value of the right hand column of pixels and that of the left hand column, in the neighborhood. A vertical derivative is obtained in the same way, and the edge strength is then computed as the square root of the sum of squares. This does not perform well on diagonal lines, so two more differences are calculated in the diagonal directions to determine another square root value, and the greater of the two is used. The method requires more computation than a traditional Sobel and does not generally produce superior results.

One of the important variables in the use of a rank operator is the size of the neighborhood. Generally, shapes that are squares (for convenience of computation) or approximations to a circle (to minimize directional effects) are used. As the size of the neighborhood is increased, however, the computational effort in performing the ranking increases rapidly. Also, these ranking operations

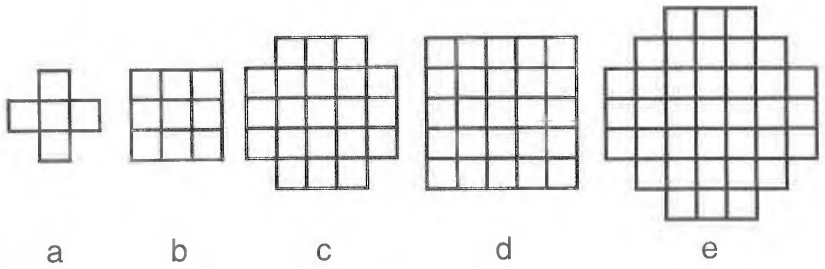


Figure 48. Neighborhood patterns used for ranking operations: *(a) 4 nearest-neighbor cross;* ***(b)*** *3 × 3 square containing nine pixels;* ***(c)*** *5 × 5 octagonal region with 21 pixels;* ***(d)*** *5 × 5 square containing 25 pixels;* ***(e)*** *7 × 7 octagonal region containing 37 pixels.*

cannot be easily programmed into specialized hardware, such as array processors. In typical cases, regions such as those shown in **Figure 48** are used.

Several uses of rank operators are appropriate for image enhancement and the selection of one portion of the information present in an image. For example, the top hat operator (Bright and Steel, 1987) has been implemented in various ways, but can be described using two different-size regions, as shown in **Figure 49**. If we assume that the goal of the operator is to find bright points, then the algorithm compares the maximum brightness value in a small central region to the maximum brightness value in a larger surrounding one. If the difference between these two values exceeds some arbitrary threshold, then the value of the central pixel is retained. Otherwise it is removed. This principle was also illustrated in Chapter 3.

Figure 50 shows the application of a top hat filter, in which the bright points in the diffraction pattern (calculated using a fast Fourier transform as discussed in Chapter 5) are retained and the overall variation in background brightness is suppressed. A top hat with inner radius of 2 pixels and outer radius of 4 pixels was used. Performing the inverse transform on only the major points provides an averaged image, combining the repetitive information from many individually noisy atom images.

The top hat filter is basically a point or feature finder. The size of the feature is defined by the smaller of the two neighborhood regions and may be as small as a single pixel in some cases. The larger region defines the local background, which the points of interest must exceed in brightness. Unlike the Laplacian, which subtracts the average value of the surrounding background from the central point, the top hat method finds the maximum brightness in the larger surrounding region (the "brim" of the hat) and subtracts that from the brightest point in the interior region. If the difference exceeds some arbitrary threshold (the "height" of the hat's crown), then the central pixel is kept.

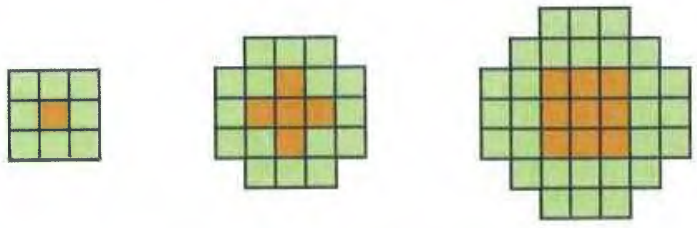


Figure 49. *Neighborhood patterns used for a top hat filter. The brightest (or darkest) value in the outer (green) region is subtracted from the brightest (or darkest) value in the inner (red) region. If the difference exceeds a threshold, then the central pixel is kept. Otherwise it is erased.*

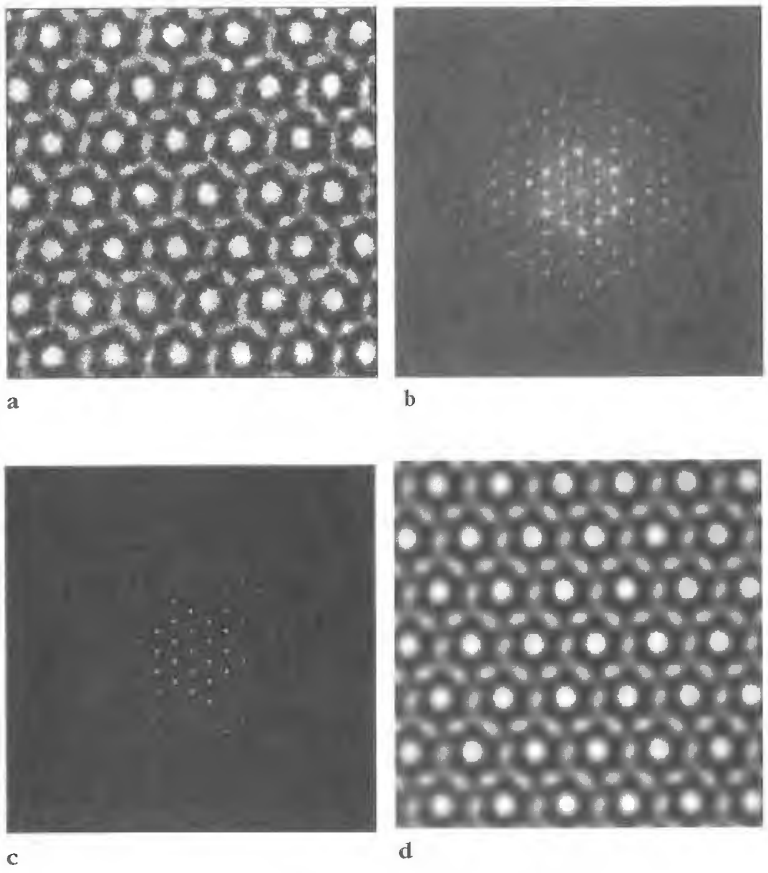


Figure 50. Application of a top hat filter to select points in a diffraction pattern: *(a) high resolution TEM image of silicon nitride, showing atomic positions; (b) FFT magnitude image calculated from image a; (c) application of a top hat filter to image b, selecting the locally bright points on a nonuniform background; (d) inverse transform using only the points selected in image c, enlarged to show detail.*

In the usual description of the operation of the algorithm, the brim rests on the data and any point that pokes through the crown of the hat is kept. If the central region is larger than a single pixel, though, this depiction is not quite accurate. When any of the points in the central region exceeds the brim value, the central pixel value (which may not be the bright one) is kept. This may produce small haloes around single bright points that are smaller than the central neighborhood region, although any bright point that is part of a feature larger than the central region will not be found by the top hat filter.

Of course, it should be noted that some images have features of interest that are darker rather than lighter than their surroundings. The logic of the top hat filter works just as well when it is inverted, and darkness rather than lightness is the test criterion. **Figure 51** shows such a case, in which the small (and quite uniformly sized) dark features are gold particles in Golgi stained muscle tissue. In this example, simply thresholding the dark particles does not work, because other parts of the image are just as dark. A linear filter method (convolution) such as unsharp masking, accomplished in the example by subtracting the average value (using a Gaussian smoothing filter with a standard deviation of 0.6 pixels) from the original, produces an image in which the particles are quite visible to a human viewer, but are still not distinguishable to a computer program.

It is possible to apply a top hat filter to one-dimensional data. The method is sometimes used, for instance, to select peaks in spectra. **Figure 52** shows an example. The brim of the hat rests on the profile and in only a few places does the plot poke through the crown. Peaks that are too low, too broad, or adjacent to others are not detected.

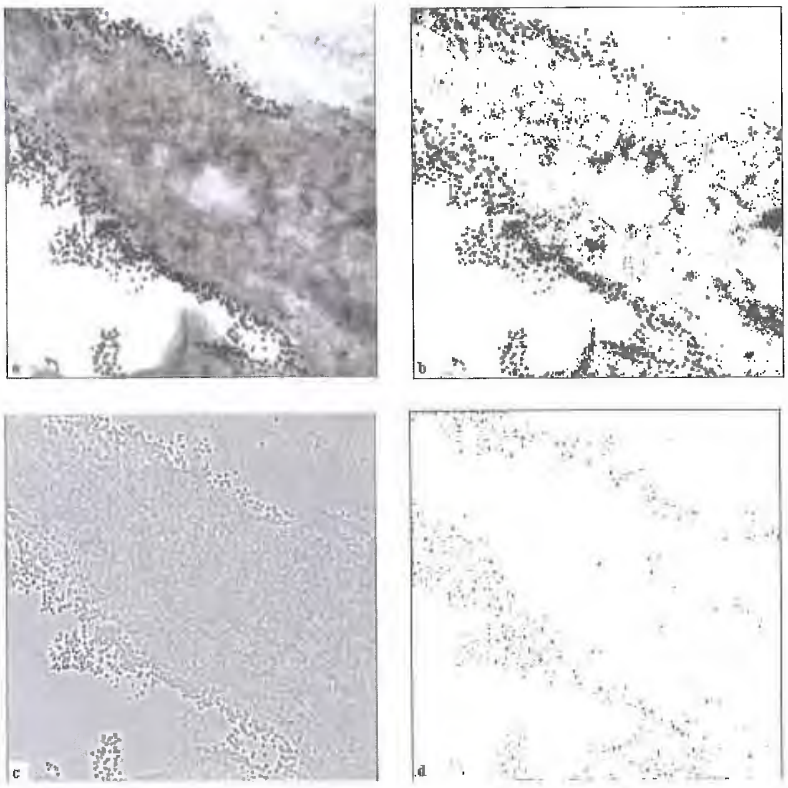


Figure 51. Application of a top-hat filter: *(a) TEM image of golgi-stained rat skeletal muscles, with gold particles (dark) on variable background; (b) thresholding of image a selects dark pixels, but cannot isolate the particles; (c) unsharp masking, produced by subtracting a smoothed version of the image from the original; (d) the top hat filter finds (most of) the particles in spite of the variation in the background.*

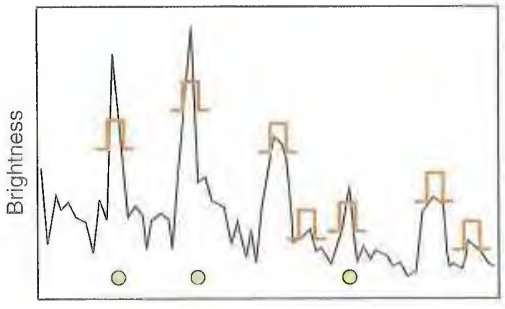


Figure 52. *Diagram of application of the top-hat filter to one-dimensional data (a brightness profile). Several placements of the filter are shown, but only for those marked with a green dot does the central value exceed the threshold.*

It is also possible, in principle, to design a top hat filter that is not circularly symmetrical, but has regions shaped to select particular features of interest, even lines. Of course, the orientation of the feature must match that of the filter, so this is not a very general operation, although it is sometimes used to locate and remove scratches in movie film as mentioned in Chapter 3. When features with a well-defined shape and orientation are sought, cross-correlation in either the spatial or frequency domain is generally more appropriate.

A close relative of the top hat filter is the rolling ball filter described in Chapter 3 as a noise removal tool. Instead of selecting points that exceed the local background, this filter eliminates them, replacing those pixel values with the neighborhood value. In **Figure 53** this filter has been used to remove the dark gold particles from the image to produce a background image with just the cell organelles. The ratio of the original image to the background isolates just the gold particles.

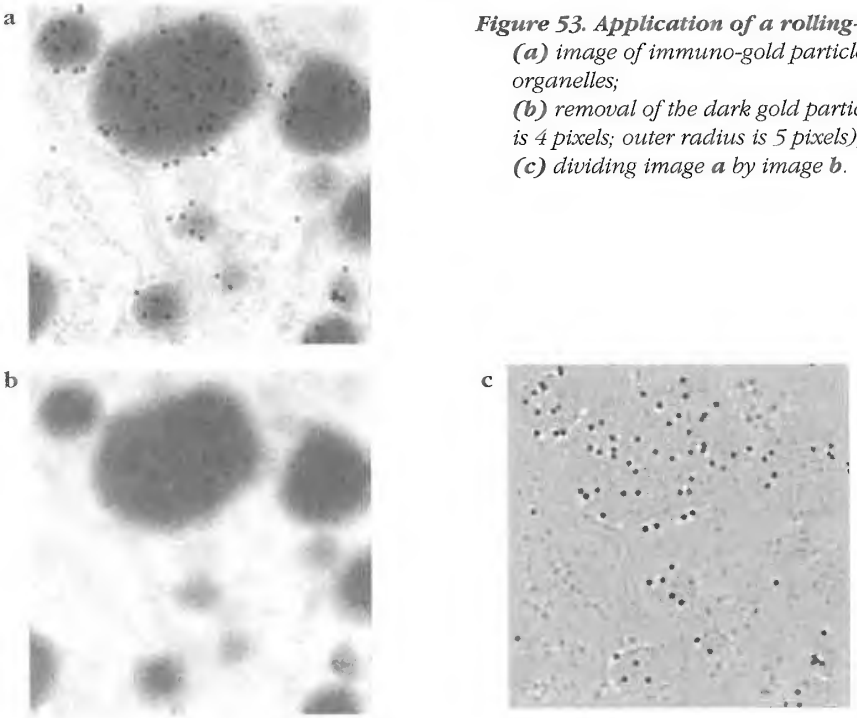


Figure 53. Application of a rolling-ball filter: *(a) image of immuno-gold particles attached to cell organelles; (b) removal of the dark gold particles (inner radius is 4 pixels; outer radius is 5 pixels); (c) dividing image **a** by image **b**.*

The top hat filter is an example of a suppression operator. It removes pixels from the image if they do not meet some criteria of being "interesting" and leaves alone those pixels that do. Another operator of this type, which locates lines rather than points, is variously known as ridge-finding or grey-scale skeletonization (by analogy to skeletonization of binary images, discussed in Chapter 7). We have already seen a number of edge-finding and gradient operators that produce bright or dark lines along boundaries. These lines are often wider than a single pixel and it may be desirable to reduce them to a minimum width (Bertrand et al., 1997).

The ridge-finding algorithm suppresses pixels (by reducing their value to zero) if they are not part of the center ridge of the line. **Figure 46** showed an example of the application as part of the Canny edge finding filter. When the method is applied to any of the edge-finding gradient operations, it tends to reduce the image to a sketch. **Figure 54** shows an example using a color image, in which the Sobel gradient filter was first applied to the individual red, green, and blue channels, followed by ridge-finding.

Many structures are best characterized by a continuous network, or tesselation, of boundaries. **Figure 55** shows an example, a ceramic containing grains of two different compositions (the dark grains are alumina and the light ones are zirconia). The grains are easily distinguishable by the viewer, but it is the grain boundaries that are important for measurement to characterize the structure, not just the boundaries between light and dark grains, but also those between light and light or dark and dark grains.

Figure 56 shows the application of various edge-finding operators to this image. In this case, the variance produces the best boundary demarcation without including too many extraneous lines due to the texture within the grains. Thresholding this image (as discussed in Chapter 6) and processing the binary image (as discussed in Chapter 7) to thin the lines to single pixel width produces an image of only the boundaries, as shown in **Figure 57**. It is then possible to use the brightness

Figure 54. Thinning or ridge-finding applied to a color image:
(a) original;
(b) Sobel gradient filter applied to the R, G, B channels;
(c) each color channel thinned.



a



b



c

of the pixels in the original image to classify each grain as either α or β. Images of only those boundaries lying between α and α, β and β, or α and β can then be obtained (as shown in the figure and discussed in Chapter 7). It is also possible to count the number of neighbors around each grain and use a color or grey scale to code them, as shown in **Figure 58**. This goes beyond the usual scope of image processing, however, and into the realm of image measurement and analysis, which is taken up in Chapter 9.

Figure 55. *SEM image of thermally etched alumina-zirconia multiphase ceramic. The two phases are easily distinguished by brightness, but the boundaries between two light or two dark regions are not. (Image courtesy of Dr. K. B. Alexander, Oak Ridge National Labs, Oak Ridge, TN)*

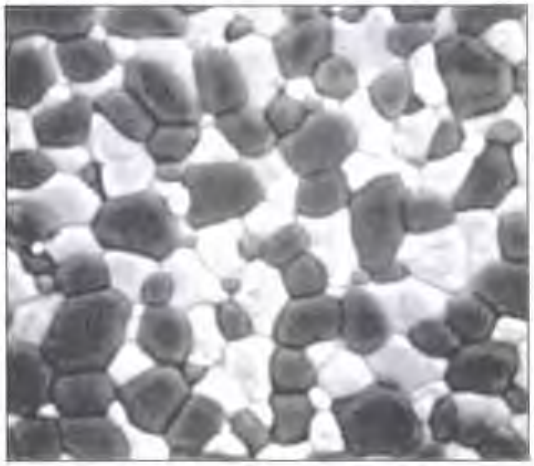
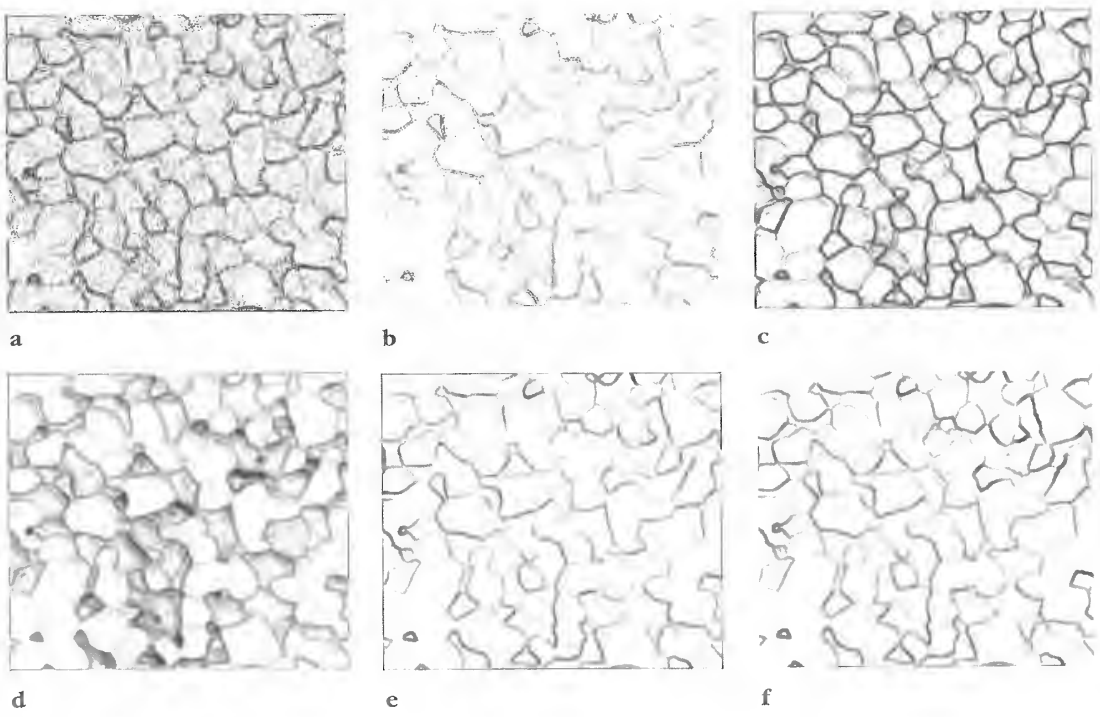


Figure 56. Processing of Figure 55 to enhance edges (all operations performed on 3 × 3 neighborhood regions):
(a) *absolute value of the Laplacian;*
(b) *absolute value of the difference between original and Gaussian smooth;*
(c) *Sobel gradient operator;*
(d) *Frei and Chen edge operator;*
(e) *variance operator;*
(f) *range operator.*



A range image, such as those from the atomic force microscope (AFM) or interference microscope, or from radar imaging, assigns each pixel a brightness value representing elevations, so it is not necessary to arbitrarily interpret the grey-scale brightness as such a structure. Rank operations are particularly well suited to such images, and can often be used to locate boundaries. **Figure 59** shows an example, an AFM image of the topography of a deposited coating. There is some variation in elevation from noise in the image, producing a local variation in grey levels that "roughens" the surface. This can be reduced with a rolling ball filter as shown.

Performing a grey-scale dilation (replacing each pixel with the brightest pixel in a 5-pixel wide neighborhood) and subtracting the original produces a set of bright lines along the boundaries. Thresholding and skeletonizing this image (using techniques discussed in Chapters 6 and 7) produce a network of lines as shown in the figure. Overlaying these lines on the original shows that the correspondence to the visual boundaries between regions is not perfect. This is due primarily to the fact that the surface slope on each side of the boundaries is different, and the subtraction process therefore produces an asymmetric border that has a midline that is offset from the boundary. Nevertheless, the tesselation of lines is useful for counting and measuring the individual structures in the coating.

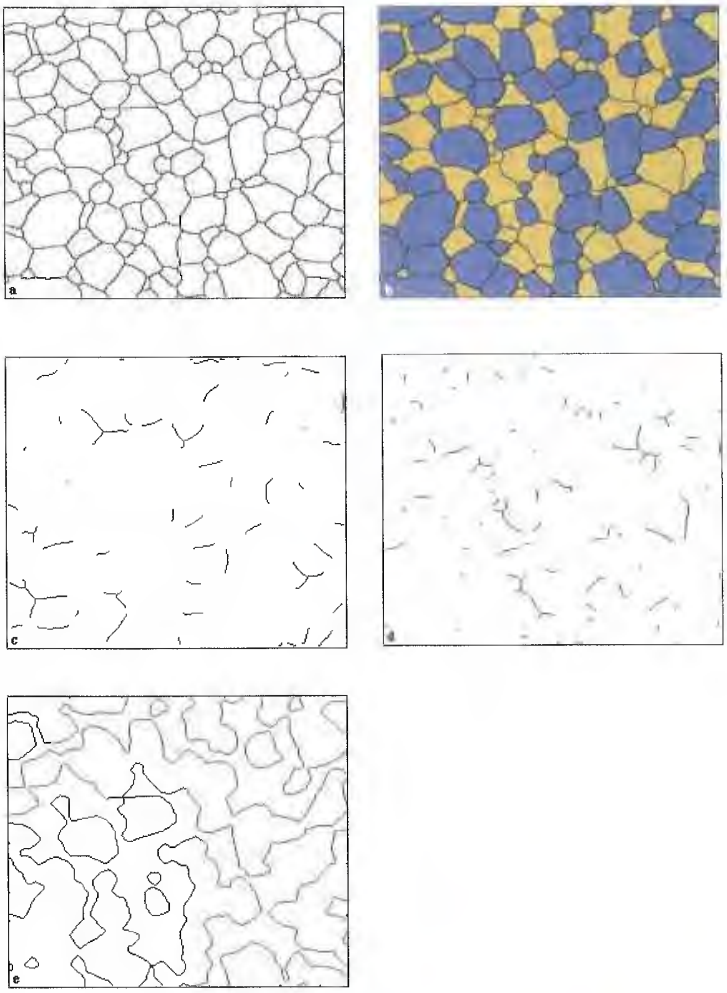


Figure 57. Grain boundary images derived from the variance image in Figure 56e by thresholding and skeletonization: *(a) boundaries between the grains; (b) grains grey-scale coded to show phase identification; (c) only those boundaries between two dark (alumina) grains, making up 16.2% of the total; (d) only those boundaries between two light (zirconia) grains, making up 15.2% of the total; (e) only those boundaries between a light and dark grain, making up 68.6% of the total boundary.*

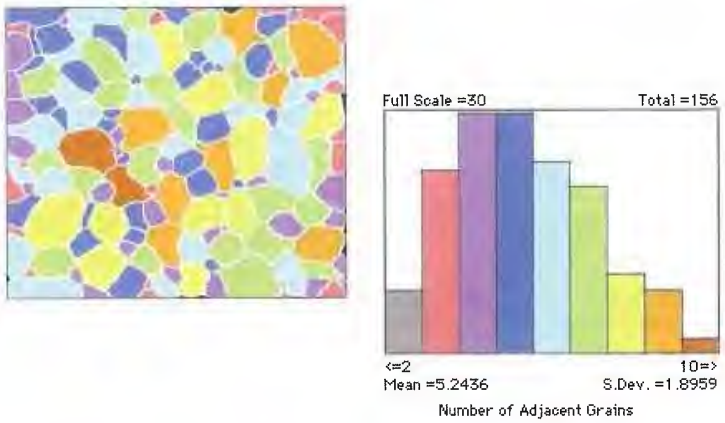
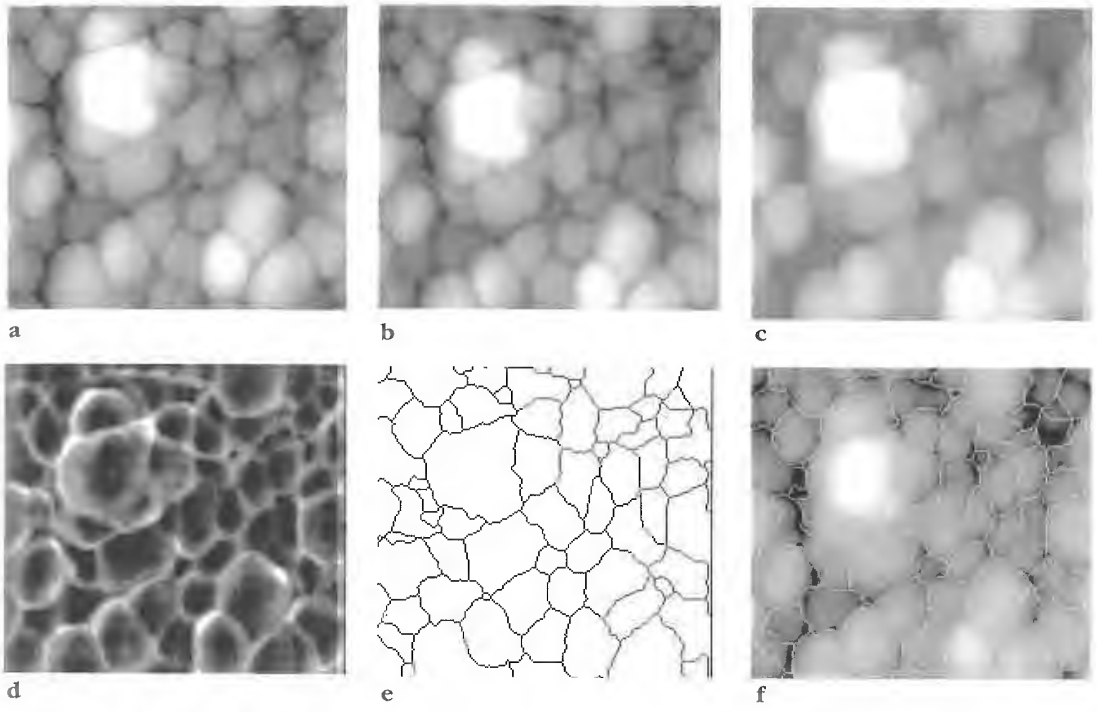


Figure 58. *These grains from* ***Figure 57*** *are color-coded to show the number of neighbors touching each and a plot of the frequency of each number of neighbors. Further analysis shows that both the size and the number of neighbor plots are different for the two different phases.*

Figure 59. An atomic force microscope (AFM) image of a deposited coating, showing a series of gently rounded bumps:
(a) the original range image (grey scale proportional to elevation);
(b) noise smoothed by the application of a rolling ball filter;
(c) use of a rank filter to replace each pixel with its brightest neighbor;
(d) difference between b and c, showing boundary delineation;
(e) thresholding and skeletonizing image d to get lines;
(f) boundary lines overlaid on the original image.



For range images (particularly radar images used in surveillance), searching for a target pattern can be accomplished using a special class of adaptive operators. A top hat filter of the right size and shape can be used for the task, but better performance can be achieved by adjusting the parameters of size and especially height according to the local pixel values (Verly and Delanoy, 1993). For example, the threshold difference in brightness between the inner and outer regions could be made a percentage of the brightness rather than a fixed value. In principle, the more knowledge available about the characteristics of the target and of the imaging equipment, the better an adaptive filter can be made to find the features and separate them from background. In practice, these approaches seem to be little used, and are perhaps too specific for general applications.

Texture

Many images contain regions characterized not so much by a unique value of brightness, but by a variation in brightness that is often called texture. This is a somewhat loosely defined term that refers to the local variation in brightness from one pixel to the next or within a small region. If the brightness is interpreted as elevation in a representation of the image as a surface, then the texture is a measure of the surface roughness, another term without a single accepted or universal quantitative meaning.

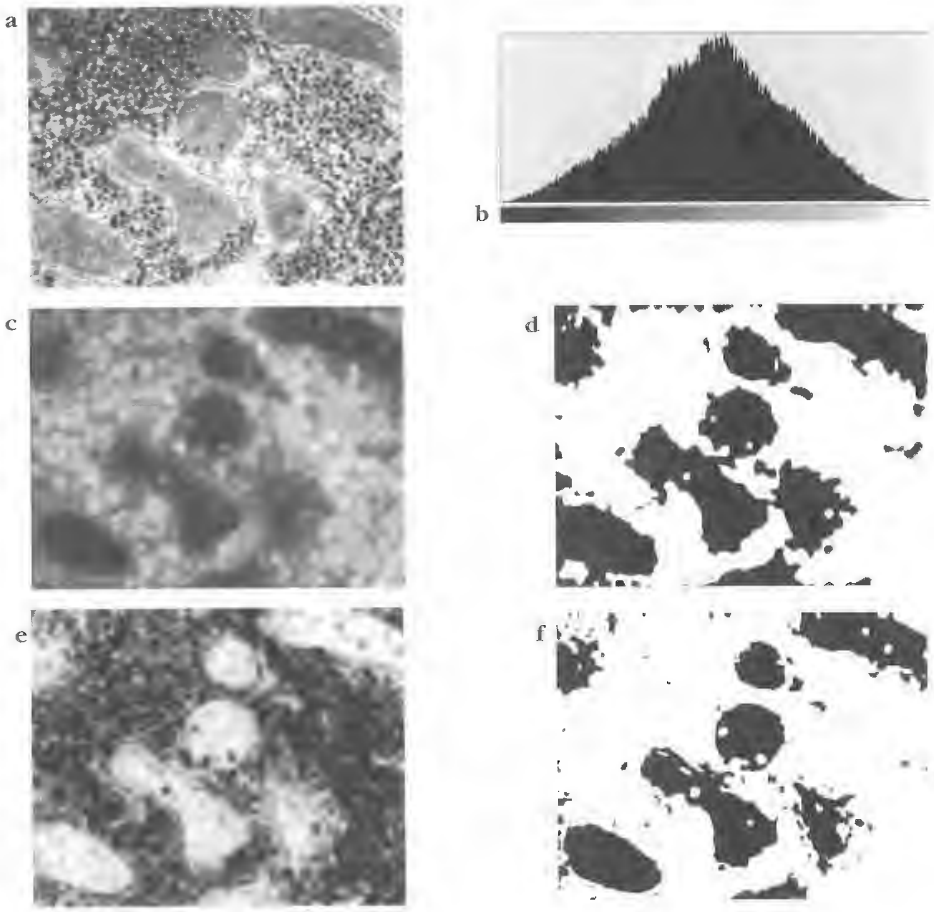


Figure 60. Enhancement of texture:
(a) *transmission electron microscope image of liver thin section;*
(b) *histogram of* ***a****;*
(c) *range image (difference between maximum and minimum brightness values in a neighborhood with 3.5-pixel radius);*
(d) *binary image produced by automatic thresholding of* ***c****;*
(e) *variance of pixel values in a neighborhood with 3.5-pixel radius;*
(f) *binary image produced by automatic thresholding of* ***e****.*

Rank operations are also used to detect this texture in images. One of the simplest of the texture operators is simply the range or difference between maximum and minimum brightness values in the neighborhood. For a flat or uniform region, the range is small. Larger values of the range correspond to surfaces with a larger roughness. The size of the neighborhood region must be large enough to include dark and light pixels, which generally means being larger than any small uniform details that may be present. **Figure 60** shows an example in which the original image has a histogram with a single broad peak and no ability to distinguish the visually smooth and textured regions based on brightness. The range image produces different brightness values which allow thresholding. A second texture extraction method, also shown in the figure, is the calculation of the statistical variance of the pixel values in a moving neighborhood, which also is able to distinguish the regions in this image. Note that both the range and variance operators are also sometimes used with a smaller neighborhood size to locate edges, as shown in **Figure 61**.

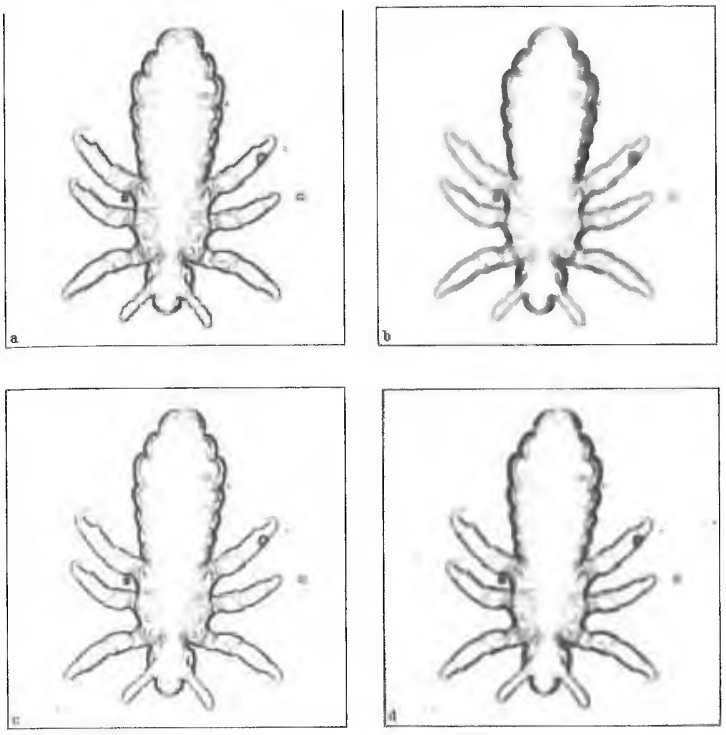
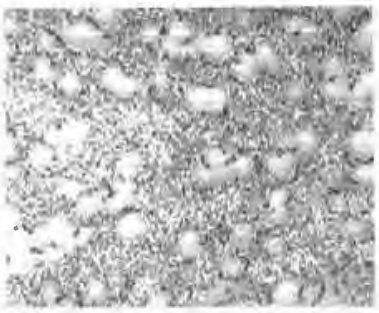


Figure 61. Application of range and variance operators to the test image from Figure 41a: *(a) range (3 × 3 neighborhood); (b) range (5-pixel-wide circular neighborhood); (c) variance (3 × 3 neighborhood); (d) variance (5-pixel-wide circular neighborhood).*

When visual examination of an image suggests that the basis for discriminating various structural regions is a texture rather than color or brightness, it is often possible to use a simple texture operator such as the range to extract it. This is often the case in biological specimens, and particularly in foods. **Figure 62** shows another example, a microscope image of the curds and whey protein in cheese. The smooth regions (curds) produce a low range value, while the highly textured whey produces a larger range value. The overall shading of the image has also been removed.

Satellite images are especially appropriate for characterization by texture operators. Categorization of crops, construction, and other land uses produces distinctive textures that humans can recognize. Therefore, methods have been sought that duplicate this capability in software algorithms. In a classic paper on the subject, Haralick listed 14 such texture operators that utilize the pixels within a region and their brightness differences (Haralick et al., 1973; Haralick, 1979; Weszka et al., 1976). This region is not a neighborhood around each pixel, but comprises all of the pixels within a contiguous block delineated by some boundary or other identifying criterion such as brightness, etc. A table is constructed with the number of adjacent pixel pairs within the region as a function of their brightnesses. This pixel table is then used to calculate the texture parameters.

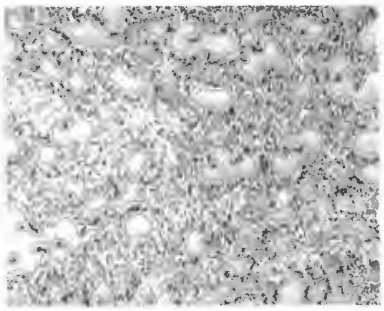
In the expressions below, the array P(i,j) contains the number of nearest-neighbor pixel pairs (in 90-degree directions only) whose brightnesses are i and j, respectively. R is a renormalizing constant equal to the total number of pixel pairs in the image or any rectangular portion used for the calculation. In principle, this can be extended to pixel pairs that are separated by a distance d and to pairs aligned in the 45-degree direction (whose separation distance is greater than ones in the 90-degree directions). The summations are carried out for all pixel pairs in the region. Haralick applied this to rectangular regions, but it is equally applicable to pixels within irregular outlines.



a



b



c

Figure 62. Application of a range operator to an image of curds and whey protein: (a) *original;* ***(b)*** *range operator applied (5-pixel-wide circular neighborhood);* ***(c)*** *region outlines from thresholding superimposed on original.*

The first parameter shown is a measure of homogeneity using a second moment. Since the terms are squared, a few large differences will contribute more than many small ones. The second one shown is a difference moment, which is a measure of the contrast in the image. The third is a measure of the linear dependency of brightness in the image, obtained by correlation.

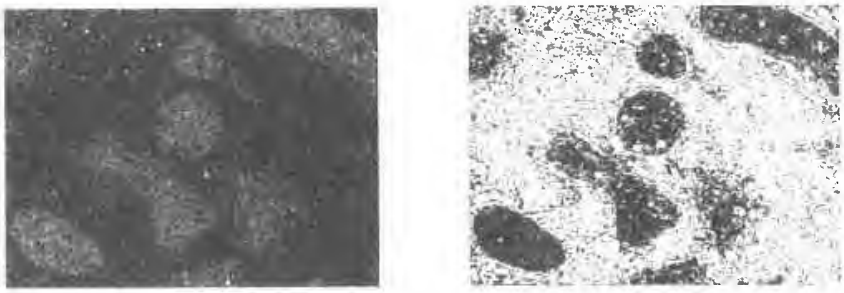
$$f_1 = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{P(i,j)}{R}\right)^2$$

$$f_2 = \sum_{n=0}^{N-1} n^2 \left\{ \sum_{|i-j|=n} \left(\frac{P(i,j)}{R}\right) \right\} \qquad (8)$$

$$f_3 = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}[i \cdot j \cdot P(i,j)/R] - \mu_x \cdot \mu_y}{\sigma_x \cdot \sigma_y}$$

In these expressions, N is the number of grey levels, and μ and σ are the mean and standard deviation, respectively, of the distributions of brightness values accumulated in the x and y directions. Additional parameters describe the variance, entropy, and information measure of the brightness value correlations. Haralick has shown that when applied to large rectangular areas in satellite photos, these parameters can distinguish water from grassland, different sandstones from each other, and woodland from marsh or urban regions.

Some of these operations are easier to calculate than others for all of the pixels in an image. The resulting values can be scaled to create a derived image that can be discriminated with brightness



a b

Figure 63. Application of Haralick texture operators to the image from Figure 60:
(a) homogeneity;
(b) difference moment.

thresholding. In any given instance, it sometimes requires experimentation with several texture operators to find the one that gives the best separation between the features of interest and their surroundings.

Some of these same operations can be applied to individual pixels to produce a new image, in which the brightness is proportional to the local texture. **Figure 63** illustrates the use of the Haralick homogeneity and angular second moment operators (f_1 and f_2 above) applied in a moving neighborhood centered on each pixel to calculate a texture value, which is then assigned to the pixel. This result can be compared to **Figure 60**.

Fractal analysis

The characterization of surface roughness by a fractal dimension has been applied to fracture surfaces, wear and erosion, corrosion, etc. (Mandelbrot et al., 1984; Underwood and Banerji, 1986; Mecholsky and Passoja, 1985; Mecholsky et al., 1986, 1989; Srinivasan et al., 1991; Fahmy et al., 1991). It has also been shown (Pentland, 1983; Peleg et al., 1984) that the brightness pattern in images of fractal surfaces is also mathematically a fractal and that this also holds for SEM images (Russ, 1990a). A particularly efficient method for computing the fractal dimension of surfaces from elevation images is the Hurst coefficient, or rescaled range analysis (Hurst et al., 1965; Feder, 1988; Russ, 1990c). This procedure plots the greatest difference in brightness (or elevation, etc.) between points along a linear traverse of the image or surface as a function of the search distance, on log-log axes. When the range is scaled by dividing by the standard deviation of the data, the slope of the resulting line is directly related to the fractal dimension of the profile.

Performing such an operation at the pixel level is interesting, because it may permit local classification that can be of use for image segmentation. Processing an image so that each pixel value is converted to a new brightness scale indicating local roughness (in the sense of a Hurst coefficient) permits segmentation by simple brightness thresholding. It uses two-dimensional information on the brightness variation, compared to the one-dimensional comparison used in measuring brightness profiles.

Figure 64 shows a neighborhood region consisting of 37 pixels in a 7-pixel-wide octagonal shape. The size is a compromise between the desire to include many pixel values (for accurate results) and the need for a local operator. Qualitatively similar results are obtained with 5-, 9-, and 11-pixel-wide regions. Unlike some neighborhood operators, such as smoothing kernels, the use of progressively larger neighborhood regions for the Hurst operator does not select information with different dimensions or scales in the image. Instead, it increases the precision of the fit and reduces

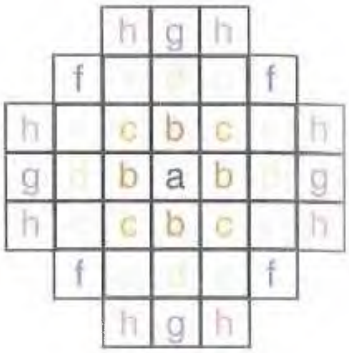
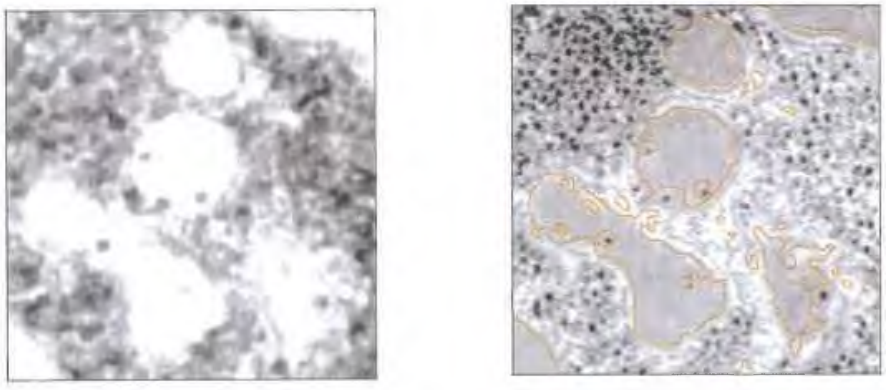


Figure 64. *Octagonal 7-pixel-wide neighborhood (37 pixels total) used for local rescaled-range (Hurst coefficient) calculation. Pixel labels identify groups with the same distance from the central pixel.*

the noise introduced by individual light or dark pixels, at least up to a region size as large as the defined structures in the image. Each of the pixels in the diagram is labeled to indicate its distance from the center of the octagon. The distances (in pixel units) range from 1 pixel (the 4 touching neighbors sharing a side with the central pixel) to 3.162 pixels ($\sqrt{10}$).

Application of the operator proceeds by examining the pixels in the neighborhood around each pixel in the original image. The brightest and darkest pixel values in each of the distance classes are found and their difference used to construct a Hurst plot. Performing a least-squares fit of the slope of the log (brightness difference) vs. log (distance) relationship is simplified because the distance values (and their logarithms) are unvarying and can be stored beforehand in a short table. It is also unnecessary to divide by the standard deviation of pixel brightnesses in the image, since this is a constant for each pixel in the image and the slope of the Hurst plot will be scaled to fit the brightness range of the display anyway. Building the sums for the least-squares fit and performing the necessary calculations is moderately complex. Thus, it is time consuming compared to simple neighborhood operations such as smoothing, etc., but still well within the capability of typical desktop computer systems.

Figure 65 shows a portion of the TEM image of a thin section of liver tissue, used in **Figures 60** and **63**. The conversion of the original image based on the texture information permits thresholding of the



a **b**

Figure 65. Application of the Hurst texture operator to the image from Figure 60 (a), and the region outlines from thresholding superimposed on the original (b).

regions is shown in the figure. **Figure 66** shows highly magnified images of two representative locations in each of these regions, with outlines showing specific locations of the octagonal neighborhood. Sorting through the pixels in each distance class, finding the brightest and darkest and their difference, and constructing a plot of log (brightness range) vs. log (distance) is shown for these two specific pixel locations in **Figure 67**. Notice that the slopes of the Hurst plots are quite different and that the lines fit the points rather well. Scaling the Hurst values to the brightness range of an image and applying the operation to each pixel in the image produces the result shown in **Figure 65**.

Figure 68 shows an image of the broken end of a steel test specimen. Part of the surface was produced by fatigue and part by the terminal tearing failure. Visual examination easily distinguishes the two regions based on texture. Measuring the area of the fatigue crack is important for determining the mechanical properties of the steel, although the boundary is quite difficult to locate by computer processing. The brightness histograms of two regions of the image overlap extensively, so that simple thresholding is not possible. Applying the local Hurst operator to this image produces the result shown, in which the two regions are clearly delineated.

Implementation notes

Many of the techniques discussed in this chapter and in Chapter 3 are neighborhood operators that access pixels in a small area around each central pixel, perform some calculation or comparison with those values, and then derive a new value for the central pixel. In all cases, this new value is used to produce a new image, and it is the original values of pixels which are used in the neighborhood around the next pixel as the operation is repeated throughout the image.

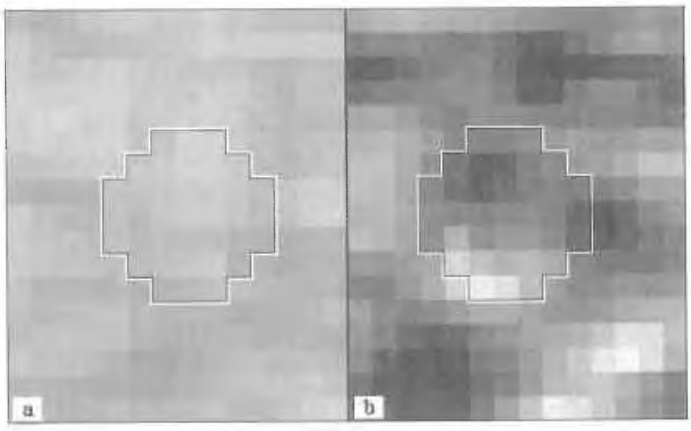


Figure 66. *Expanded detail of two locations in the image from* ***Figure 65a****, with positions of the measurement neighborhood inidcated.*

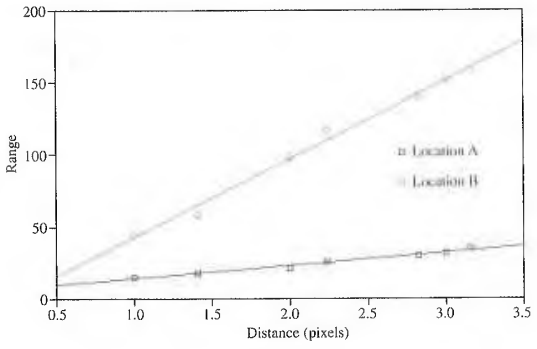
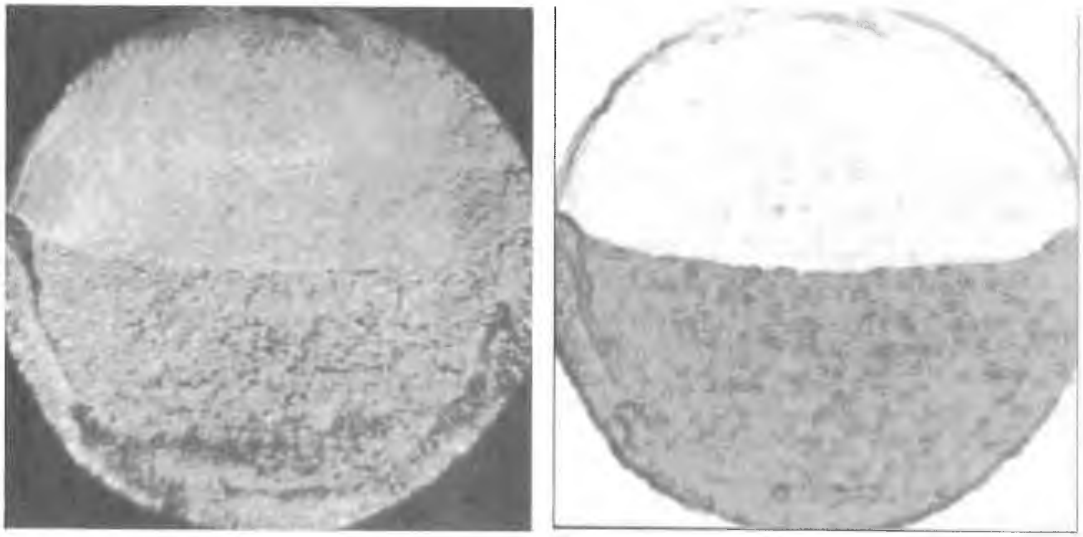


Figure 67. *Hurst plots of the logarithm of the maximum brightness range vs. log of distance for the two locations shown in* ***Figure 66.***



a b

Figure 68. Fracture surface of a steel test specimen. The smooth portion of the fracture occurred by fatigue, and the rougher portion by tearing:
(a) *original;*
(b) *after application of the Hurst texture operator.*

Most image analysis systems, particularly those operating in desktop computers, have limited memory (particularly when the large size of images is considered). Creating a new image for every image processing operation is an inefficient use of this limited resource. Consequently, the strategy generally used is to perform the operation "in place," to process one image and replace it with the result.

This requires only enough temporary memory to hold a few lines of the image. The operations are generally performed left to right along each scan line and top to bottom through the image. Duplicating the line that is being modified, and keeping copies of the preceding lines whose pixels are used, allows the new (modified) values to be written back to the original image memory. The number of lines is simply $(n + 1)/2$ where n is the neighborhood dimension (e.g., 3×3, 5×5, etc.). Usually, the time required to duplicate a line from the image is small and by shuffling through a series of pointers, it is only necessary to copy each line once when the moving process reaches it, then re-use the array for subsequent vertical positions.

Some of the image processing methods described above create two or more intermediate results. For example, the Roberts' Cross or Sobel filters apply two directional derivatives whose magnitudes are subsequently combined. It is possible to do this pixel by pixel, so that no additional storage is required; however, in some implementations, particularly those that can be efficiently programmed into an array processor (which acts on an entire line through the image at one time), it is faster to obtain the intermediate results for each operator applied to each line and then combine them for the whole line. This requires only a small amount of additional storage for the intermediate results.

Another consideration in implementing neighborhood operations is how to best treat pixels near the edges of the image. Many of the example images shown here are taken from the center of larger original images, so edge effects are avoided. In others, a band around the edge of the image is skipped, which is one of the most common ways to respond to the problem. With this approach, the programs skip pixels within a distance of $(n - 1)/2$ pixels from any edge, where n is the total width of the neighborhood.

Other possibilities include having special neighborhood rules near edges to sort through a smaller set of pixels, duplicating rows of pixels at edges (i.e., assuming each edge is a mirror), or using wrap-around addressing (i.e., assuming that the left and right edges and the top and bottom edges of the image are contiguous). None of these methods is particularly attractive, in general. Since the largest and smallest brightness values are used to find the maximum range, the duplication of rows of pixels would not provide any extra information for range operations. In all cases, the use of fewer pixels for calculations would degrade the precision of the results. There is no reason whatsoever to assume that the image edges should be matched, and indeed, quite different structures and regions will normally occur there. The most conservative approach is to accept a small shrinkage in useful image size after processing, by ignoring near-edge pixels.

Image math

The image processing operations discussed so far in this chapter operate on one image and produce a modified result, which may be stored in the same image memory. Another class of operations uses two images to produce a new image (which may replace one of the originals). These operations are usually described as image arithmetic, since operators such as addition, subtraction, division, and multiplication are included. They are performed pixel by pixel, so that the sum of two images simply contains pixels whose brightness values are the sums of the corresponding pixels in the original images. There are some additional operators used, as well, such as comparing two images to keep the brighter (or darker) pixel, obtaining the absolute difference, or basing the comparison on neighborhood values (for instance, keeping the pixel with the greater local variance, illustrated in Chapter 3 as a means of combining images with different focus settings). Other two-image operations, such as Boolean OR or AND logic, are generally applied to binary images; they will be discussed in that context in Chapter 7.

Actually, image addition has already been used in a method described previously. In Chapter 3, the averaging of images to reduce noise was discussed. The addition operation is straightforward, but a decision is required about how to deal with the result. If two 8-bit images (with brightness values from 0 to 255 at each pixel) are added together, the resulting value can range from 0 to 510. This exceeds the capacity of the image memory. One possibility is simply to divide the result by two, obtaining a resulting image that is correctly scaled to the 0 to 255 range. This is what is usually applied in image averaging, in which the N images added together produce a total, which is then divided by N to rescale the data.

Another possibility is to find the largest and smallest actual values in the sum image, and then dynamically rescale the result to this maximum and minimum, so that each pixel is assigned a new value B = range × (sum − minimum)/(maximum − minimum), where range is the capacity of the image memory, typically 255. This is superior to performing the division by two and then subsequently performing a linear expansion of contrast, as discussed in Chapter 3, because the precision of the resulting values is higher. When the integer division by two is performed, fractional values are truncated and some information may be lost.

On the other hand, when dynamic ranging or automatic scaling is performed, it becomes more difficult to perform direct comparison of images after processing, since the brightness scales may not be the same. In addition, autoscaling takes longer, since two complete passes through the image are required: one to determine the maximum and minimum and one to apply the autoscaling calculation. Many of the images printed in this book have been autoscaled in order to maximize printed contrast. In most cases this operation has been performed as part of the processing operation to maintain precision.

Adding together images superimposes information and can in some cases be useful to create composites, which help to communicate complex spatial relationships. We have already seen that

adding the Laplacian or a derivative image to the original can help provide some spatial guidelines to interpret the information from the filter. Usually, this kind of addition is handled directly in the processing by changing the central value of the kernel. For the Laplacian, this modification is called a sharpening filter, as noted previously.

Subtracting images

Subtraction is widely used and more interesting than the addition operation. In Chapter 3, subtraction was used to level images by removing background. This chapter has already mentioned uses of subtraction, such as that employed in unsharp masking, where the smoothed image is subtracted, pixel by pixel, from the original. In such an operation, the possible range of values for images whose initial range is 0 to 255 becomes –255 to +255. The data can be rescaled to fit into a single byte, replacing the original image, by dividing by two and adding 128, or the same autoscaling method described above for addition may be employed. The same advantages and penalties for fixed and flexible scaling are encountered.

In some cases, the absolute difference may be preferred to simple subtraction of a background. **Figure 69** shows a phase-contrast image of cells on a slide. Images such as this in which features have pronounced bright and dark shadows on opposite sides are very difficult to measure, because on two sides there are different criteria to define an edge, and on the top and bottom the edges are not revealed at all but are inferred by the viewer. Using a large median filter to remove the shadows produces a "background" image without the features. The absolute difference between this and the original shows both shadows as bright. Thresholding this image produces two disconnected arcs, but dilation of the binary image (discussed in Chapter 7) merges the two sides. After filling and eroding this back to the original size, the outlines of the cells are adequately delineated for useful measurement.

Subtraction is primarily a way to discover differences between images. **Figure 70** shows two images of coins and their difference. The parts of the picture that are essentially unchanged in the two images cancel out and appear as a uniform medium grey except for minor variations due to the precision of digitization, changes in illumination, etc. The coin that has been moved between the two image acquisitions is clearly shown. The dark image shows where the feature was; the bright one shows where it has gone.

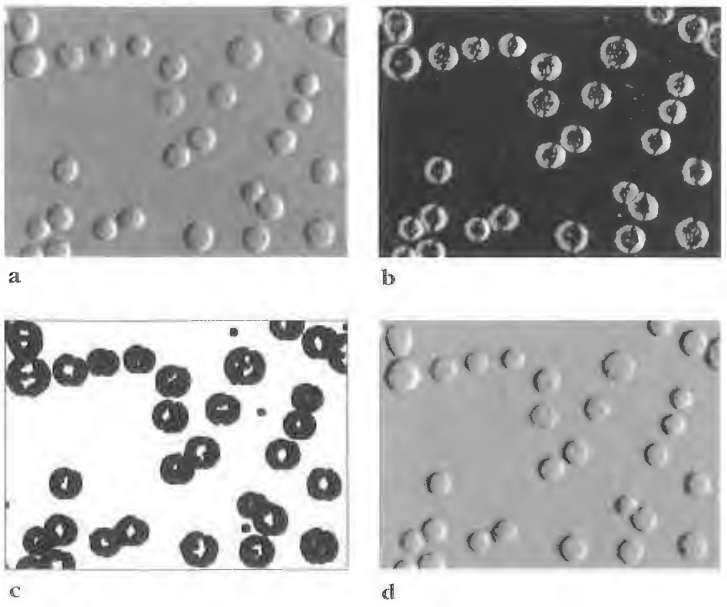


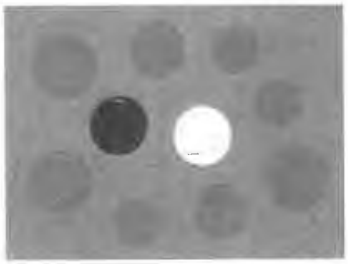
Figure 69. Isolating cells in a phase contrast image: *(a) original image; (b) absolute difference between original and a median-filtered copy; (c) image b thresholded and dilated (as discussed in Chapter 6) to join left and right shadows; (d) resulting cell boundaries superimposed on the original image.*



a



b



c

Figure 70. Showing image differences by subtraction: *(a) original image; (b) image after moving one coin; (c) difference image after pixel by pixel subtraction.*

Subtracting one image from another effectively removes from the difference image all features that do not change, while highlighting those that do. If the lighting and geometry of view is consistent, the only differences in pixel values where no changes occur are statistical variations in the brightness, due to camera or electronic noise. The bright and dark images show features that have been removed from or added to the field of view, respectively.

Even in the presence of some noise, subtraction of two images can be an effective way to identify small differences that might otherwise escape notice. **Figure 71** shows an example. The image shows two films from a Debye–Scherer x-ray camera. The vertical lines show the exposure of the film by x-rays that were diffracted from a tiny sample, each line corresponding to reflection from one plane of atoms in the structure of the material. Comparing the films from these similar samples shows that most of the lines are similar in position and intensity, because the two samples are in fact quite similar in composition. The presence of trace quantities of impurities is revealed by additional faint lines in the image. Subtraction of one set of lines from the second increases the relative amount of noise, but reveals the presence of lines from the trace compounds. These can then be measured and used for identification.

Figure 71. Image subtraction to enhance the visibility of details: *(a, b) scanned images of films from a Debye–Scherer x-ray camera, taken with similar compounds; (c) the difference between **b** and **a** showing the low intensity lines present in one film due to the presence of trace compounds in the sample.*



a



b



c

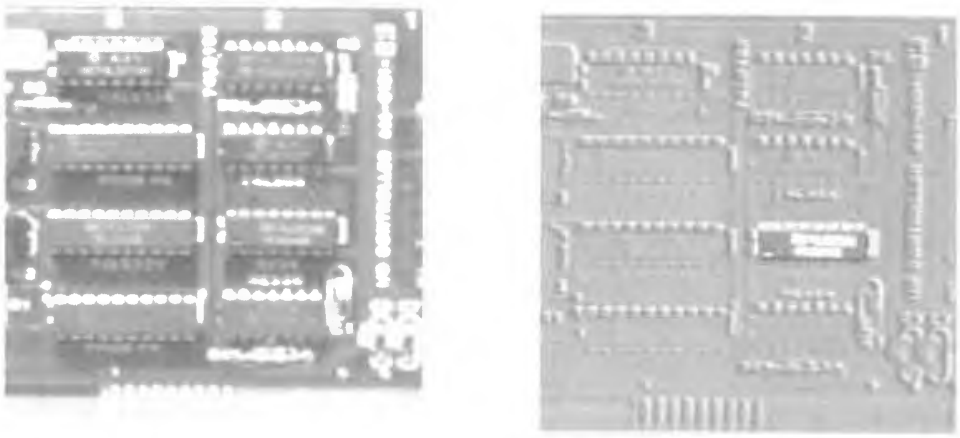


Figure 72. Difference images for quality control. A master image is subtracted from images of each subsequent part. In this example, the missing chip in a printed circuit board is evident in the difference image.

A major use of image subtraction is quality control. A master image is acquired and stored that shows the correct placement of parts on circuit boards (**Figure 72**), the alignment of labels on packaging, etc. When the image is subtracted from a series of images acquired from subsequent objects, the differences are strongly highlighted, revealing errors in production. This subtraction is often carried out at video frame rates using dedicated hardware. Since it is unrealistic to expect parts to be exactly aligned, a tolerance can be specified by the area of bright and dark (mismatched) pixels present after the subtraction.

The same technique is used in reconnaissance photos to watch for the appearance or disappearance of targets in a complex scene. Image warping, as discussed in Chapter 3, may be required to align images taken from different points of view before the subtraction can be performed. A similar method is used in astronomy. "Blinking" images taken of the same area of the sky at different times is the traditional way to search for moving planets or asteroids. This technique alternately presents each image to a human viewer, who notices the apparent motion of the point of light that is different in the two images while ignoring the stationary stars. Some use of computer searching using subtraction has been used, but, for dim objects in the presence of background noise, has not proved as sensitive as a human observer.

In the example of **Figure 73**, two images were acquired about 1 minute apart. The difference between the two clearly shows the motion of the minute hand and the change in position of the pendulum. The very small motion of the hour hand is also shown, which would be much too small to be noticed by viewing the images side by side.

Object motion can be measured using subtraction, if the features are large enough and the sequential images are acquired fast enough that they overlap in successive frames. In this case, the subtraction shows a bright area of mismatch that can be measured. The length of the unmatched region divided by the elapsed time gives the velocity; direction can be determined by the orientation of the region. This technique is used at microscopic scales to track the motion of cells on slides (**Figure 74**) in response to chemical cues.

At the other extreme, subtraction is used to track ice floes in the north Atlantic from satellite photos. For motion between two successive images that is too large for this method, it may be possible to identify the same objects in successive images based on size, shape, etc. and thus track motion. Or, one can assume that where paths cross, the points causing the least deviation of the path give the correct match (**Figure 75**); however, the direct subtraction technique is much simpler and more direct.



Figure 73. *Difference between two images of a clock taken 1 minute apart. The motion of the hands is evident.*



Figure 74. *Two frames from a videotape sequence of free swimming single-celled animals in a drop of pond water and the difference image. The length of the white region divided by the time interval gives the velocity.*

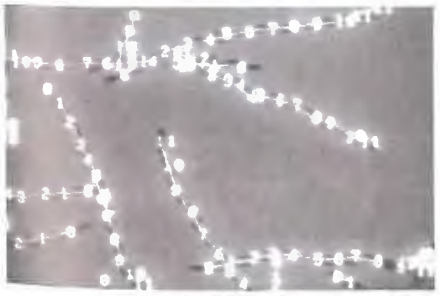


Figure 75. *Analysis of motion in a more complex situation than shown in* ***Figure 74.*** *Where the paths of the swimming microorganisms cross, they are sorted out by assuming that the path continues in a nearly straight direction (from Gualtieri and Coltelli, 1991).*

Multiplication and division

Image multiplication is perhaps the least used of the mathematics modes, but it is generally included for the sake of completeness in systems offering the other arithmetic operations. Multiplication was used above in **Figure 32** to combine the edge magnitude and direction data from the Sobel operator. Another possible use is superimposing one image on another in the particular case when the superimposed data is proportional to the absolute brightness of the original image. An example is texture; **Figure 76** shows an illustration of what is often called "bump-mapping." A brightness pattern (generated using fractal iteration) is superimposed on the smooth polygonal approximation of a shaded sphere in order to provide an impression of roughness. Similar multiplicative superimposition may be used to add fluorescence or other emission images to a reflection or transmission image.

One of the difficulties with multiplication is the extreme range of values that may be generated. With 8-bit images whose pixels can have a range between 0 and 255, the possible products can range from 0 to more than 65,000. This is a 2-byte product, only the high byte of which can be stored back into the same image memory unless automatic scaling is used. A significant loss of precision may result for values in the resulting image.

The magnitude of the numbers also creates problems with division. First, division by 0 must be avoided. This is usually done by adding 1 to all brightness values, so that the values are interpreted as 1 to 256 instead of 0 to 255. Then it is necessary to first multiply each pixel in the numerator by some factor that will produce quotients covering the 0 to 255 range, while maintaining some useful precision for the ends of the range. Automatic scaling is particularly useful for these situations, but it cannot be used in applications requiring comparison of results to each other or to a calibration curve.

An example of division in which automatic scaling is useful is the removal of background (as discussed in Chapter 3) when linear detectors or cameras are used. An example of division when absolute values are required is calculating ratios of brightness from two or more Landsat bands (an example is shown in Chapter 1) or two or more filter images when examining fluorescent probes in the light microscope. In fluorescence microscopy, the time variation of emitted light intensity is normalized by alternately collecting images through two or more filters at different wavelengths above and below the line of interest, and calibrating the ratio against the activity of the element(s)



Figure 76. Multiplication of images can be used to superimpose texture on an image:
(a) *smooth faceted globe;*
(b) *iterated midpoint displacement (fractal) pattern;*
(c) *product of* ***a*** *times* ***b****.*

of interest. In satellite imagery, ratios of intensities (particularly Band 4 = 0.5 to 0.6 μm, Band 5 = 0.6 to 0.7 μm, Band 6 = 0.7 to 0.8 μm, and Band 7 = 0.8 to 1.1 μm) are used for terrain classification and the identification of some rock types. The thermal inertia of different rock formations may also be determined by ratioing images obtained at different local times of day, as the formations heat or cool.

As an example of mineral identification, silicates exhibit a wavelength shift in the absorption band with composition. Granites, diorites, gabbros, and olivene peridotes have progressively decreasing silicon content. The absorption band shifts to progressively longer wavelengths in the 8 to 12 μm thermal infrared band as the bond-stretching vibrations between Si and O atoms in the silicate lattice change. The Thermal Infrared Multispectral Mapper satellite records six bands of image data in this range, which are combined and normalized to locate the absorption band and identify rock formations. Carbonate rocks (dolomite and limestone) have a similar absorption response in the 6 to 8 μm range, but this is difficult to measure in satellite imagery because of atmospheric absorption. At radar wavelengths, different surface roughnesses produce variations in reflected intensity in the Ka, X, and L bands and can be combined in the same ways to perform measurements and distinguish the coarseness of sands, gravels, cobbles, and boulders (Sabins, 1987).

In the same way, Bands 1 (0.55 to 0.68 μm, or visible red) and 2 (0.72 to 1.10 μm, or reflected infrared) from multispectral satellite imagery are used to recognize vegetation. Band 1 records the chlorophyll absorption and Band 2 gives the reflection from the cell structure of the leaves. The ratio (B2 – B1)/(B2 + B1) eliminates variations due to differences in solar elevation (illumination angle) and is used to measure the distribution of vegetation in images. Typically, this approach also combines data from successive scans to obtain the spectral vegetation index as a function of time. Other ratios have been used to image and to measure chlorophyll concentrations due to phytoplankton in the ocean (Sabins, 1987). **Figure 77** shows a simplified approximation of this method using the ratio of near infrared to blue to isolate vegetation from satellite imagery.

Ratios are also used in astronomical images. **Figure 78** shows infrared images of the star-forming region in NGC-2024. Infrared light penetrates the dust that blocks much of the visible light. Ratios or differences of the different wavelength images show details in the dust and enhance the visibility of the young stars.

Image math also includes the logical comparison of pixel values. For instance, two images may be combined by keeping the brighter (or darker) of the corresponding pixels at each location. This is used, for instance, to build up a confocal scanning light microscope (CSLM) image with great depth of field. Normal light microscope images have limited depth of field because of the high numerical aperture of the lenses. In the CSLM, the scanning light beam and aperture on the detector reduce this depth of field even more by eliminating light scattered from any point except the single point illuminated in the plane of focus.

A single, two-dimensional image is formed by scanning the light over the sample (or, equivalently, by moving the sample itself with the light beam stationary). For a specimen with an irregular surface, this image is very dark, except at locations where the surface lies in the plane of focus. By moving the specimen vertically, many such planar images can be acquired. The display of a complete three-dimensional set of such images is discussed in Chapter 10, although at each pixel location, the brightest value of light reflectance occurs at the in-focus point. Consequently, the images from many focal depths can be combined by keeping only the brightest value at each pixel location to form an image with an unlimited depth of field. **Figure 79** shows an example.

An additional effect can be produced by shifting each image slightly before performing the comparison and superposition. **Figure 80** shows an example. The 26 individual images, four of which

of interest. In satellite imagery, ratios of intensities (particularly Band 4 = 0.5 to 0.6 μm, Band 5 = 0.6 to 0.7 μm, Band 6 = 0.7 to 0.8 μm, and Band 7 = 0.8 to 1.1 μm) are used for terrain classification and the identification of some rock types. The thermal inertia of different rock formations may also be determined by ratioing images obtained at different local times of day, as the formations heat or cool.

As an example of mineral identification, silicates exhibit a wavelength shift in the absorption band with composition. Granites, diorites, gabbros, and olivene peridotes have progressively decreasing silicon content. The absorption band shifts to progressively longer wavelengths in the 8 to 12 μm thermal infrared band as the bond-stretching vibrations between Si and O atoms in the silicate lattice change. The Thermal Infrared Multispectral Mapper satellite records six bands of image data in this range, which are combined and normalized to locate the absorption band and identify rock formations. Carbonate rocks (dolomite and limestone) have a similar absorption response in the 6 to 8 μm range, but this is difficult to measure in satellite imagery because of atmospheric absorption. At radar wavelengths, different surface roughnesses produce variations in reflected intensity in the Ka, X, and L bands and can be combined in the same ways to perform measurements and distinguish the coarseness of sands, gravels, cobbles, and boulders (Sabins, 1987).

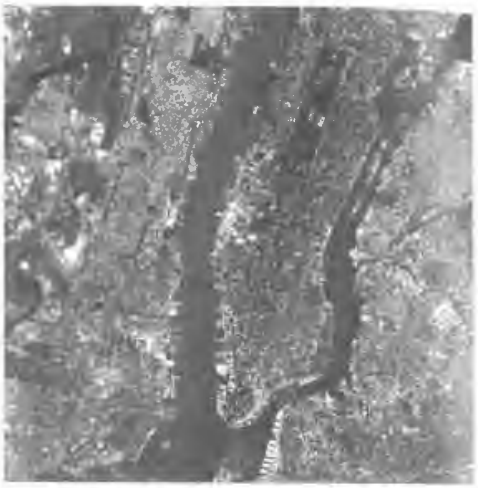
In the same way, Bands 1 (0.55 to 0.68 μm, or visible red) and 2 (0.72 to 1.10 μm, or reflected infrared) from multispectral satellite imagery are used to recognize vegetation. Band 1 records the chlorophyll absorption and Band 2 gives the reflection from the cell structure of the leaves. The ratio (B2 – B1)/(B2 + B1) eliminates variations due to differences in solar elevation (illumination angle) and is used to measure the distribution of vegetation in images. Typically, this approach also combines data from successive scans to obtain the spectral vegetation index as a function of time. Other ratios have been used to image and to measure chlorophyll concentrations due to phytoplankton in the ocean (Sabins, 1987). **Figure 77** shows a simplified approximation of this method using the ratio of near infrared to blue to isolate vegetation from satellite imagery.

Ratios are also used in astronomical images. **Figure 78** shows infrared images of the star-forming region in NGC-2024. Infrared light penetrates the dust that blocks much of the visible light. Ratios or differences of the different wavelength images show details in the dust and enhance the visibility of the young stars.

Image math also includes the logical comparison of pixel values. For instance, two images may be combined by keeping the brighter (or darker) of the corresponding pixels at each location. This is used, for instance, to build up a confocal scanning light microscope (CSLM) image with great depth of field. Normal light microscope images have limited depth of field because of the high numerical aperture of the lenses. In the CSLM, the scanning light beam and aperture on the detector reduce this depth of field even more by eliminating light scattered from any point except the single point illuminated in the plane of focus.

A single, two-dimensional image is formed by scanning the light over the sample (or, equivalently, by moving the sample itself with the light beam stationary). For a specimen with an irregular surface, this image is very dark, except at locations where the surface lies in the plane of focus. By moving the specimen vertically, many such planar images can be acquired. The display of a complete three-dimensional set of such images is discussed in Chapter 10, although at each pixel location, the brightest value of light reflectance occurs at the in-focus point. Consequently, the images from many focal depths can be combined by keeping only the brightest value at each pixel location to form an image with an unlimited depth of field. **Figure 79** shows an example.

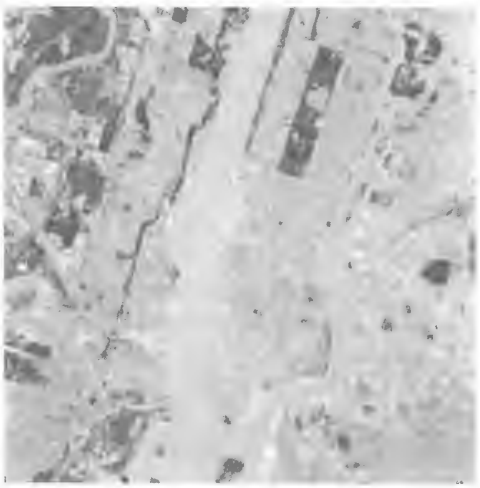
An additional effect can be produced by shifting each image slightly before performing the comparison and superposition. **Figure 80** shows an example. The 26 individual images, four of which



a



b



c

Figure 77. Landsat thematic mapper images of New York City:
(a) Band 1 (visible blue);
(b) Band 4 (near infrared);
(c) ratio of Band 4 to Band 1 (showing vegetation areas).

are shown, are combined in this way to produce a perspective view of the surface (this image was also shown in Chapter 1 as an example of one mode of collecting and displaying three-dimensional imaging information).

Another use of combining several images arises in polarized light microscopy. As shown in **Figure 81**, the use of polarizer and analyzer with specimens such as mineral thin sections produces images in which some grains are colored and others dark, as a function of analyzer rotation. Combining images from many rotations and keeping just the brightest pixel value produces an image that shows all of the grains. The same technique can be used with transmission electron microscope images of thin metal foils, to combine images in which some grains are darkened due to electron diffraction effects, or to remove dark contours that result from bending of the foil.

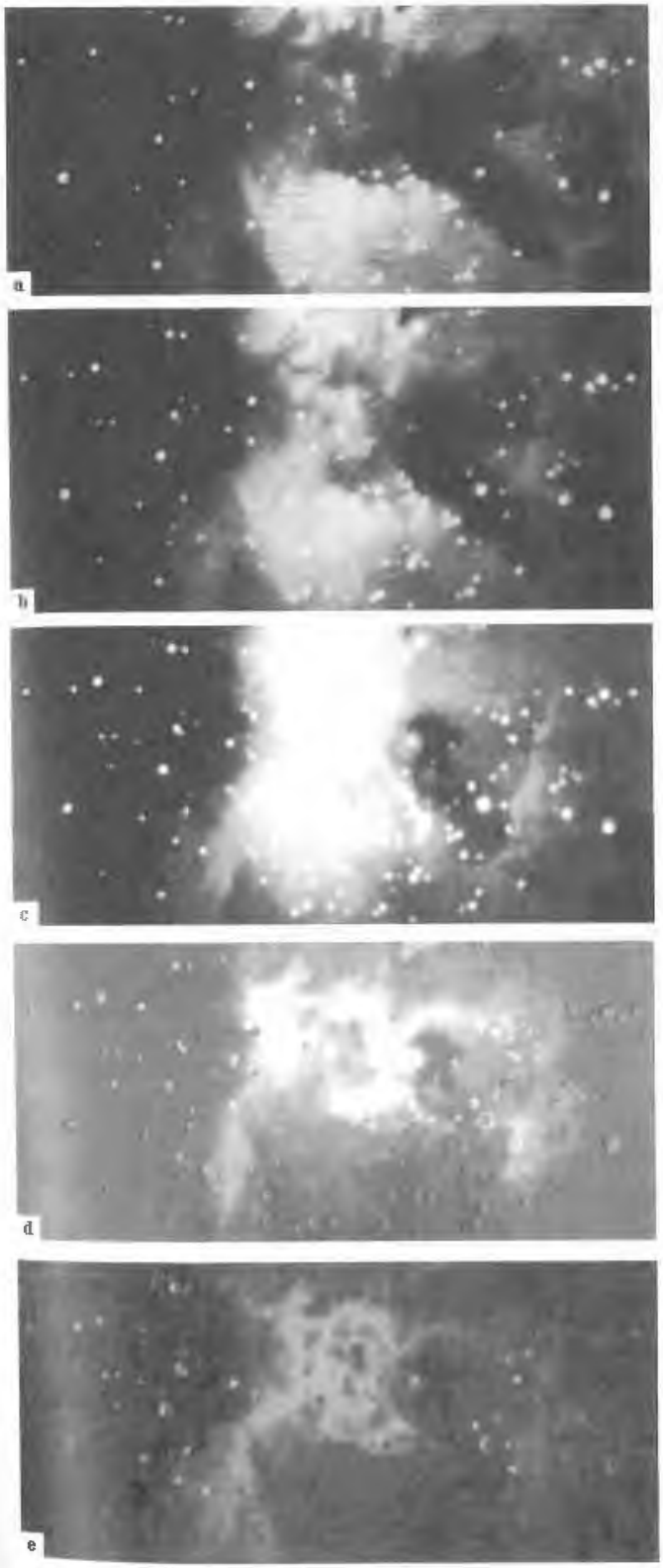


Figure 78. Combining views of NGC-2024 to show star-forming regions and dust: *(a) 1.2 μm infrared image; (b) 1.6 μm infrared image; (c) 2.2 μm infrared image; (d) 2.2 μm image minus 1.6 μm image; (e) 1.6 μm image divided by 1.2 μm image.*



a b

Figure 79. *Combining CSLM images by keeping the brightest value at each pixel location. Image* ***a*** *shows one individual focal plane image from a series of 25 on an integrated circuit. Only the portion of the surface which is in focus is bright. Because the in-focus point is brightest, combining all of the individual planes produces image* ***b****, which shows the entire surface in focus.*

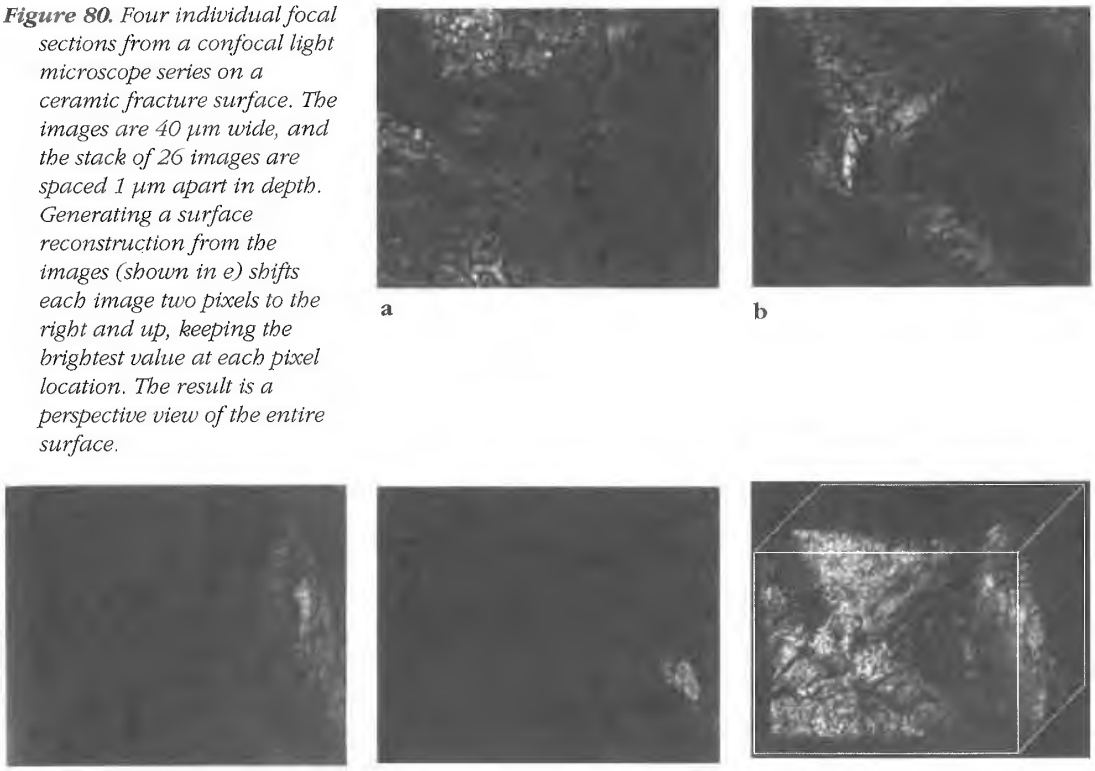
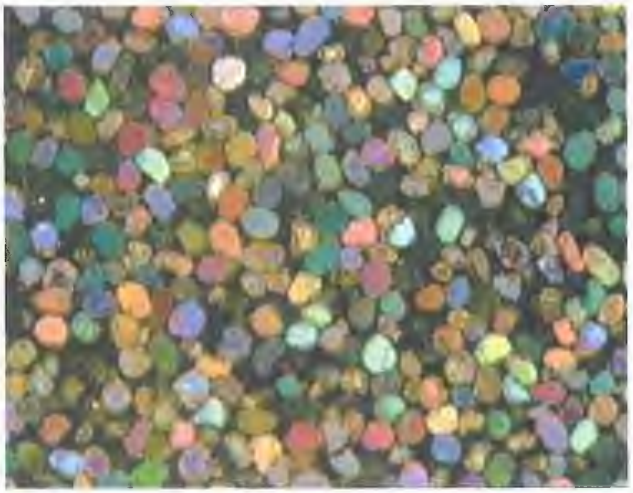


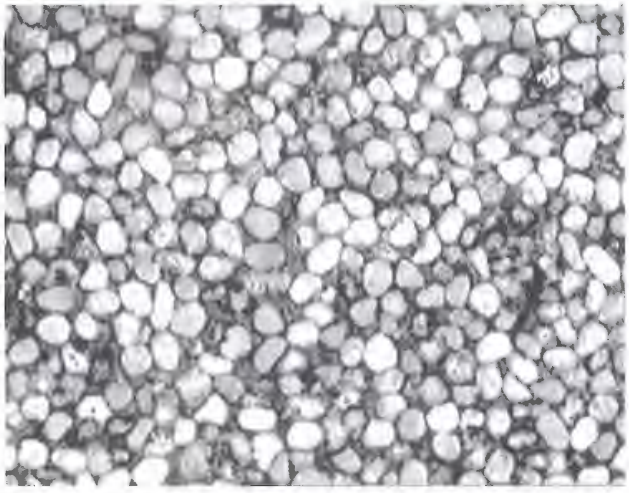
Figure 80. *Four individual focal sections from a confocal light microscope series on a ceramic fracture surface. The images are 40 µm wide, and the stack of 26 images are spaced 1 µm apart in depth. Generating a surface reconstruction from the images (shown in e) shifts each image two pixels to the right and up, keeping the brightest value at each pixel location. The result is a perspective view of the entire surface.*



a



b



c

Figure 81. Thin section of sandstone viewed by polarized light. (a, b) *different orientations of the analyzer;* ***(c)*** *maximum brightness image from six rotations.*

5

Processing Images in Frequency Space

Some necessary mathematical preliminaries

What frequency space is all about

It is unusual to pick up a book on image analysis without finding at least a portion of it devoted to a discussion of Fourier transforms (see especially Pratt, 1991; Gonzalez and Wintz, 1987; Jain, 1989). In part, this is due to the utility of working in frequency space to perform certain image measurement and processing operations. Many of these same operations can be performed in the original (spatial or pixel domain) image only with significantly greater computational effort. Another reason for the lengthy sections on these methods is that the authors frequently come from a background in electrical engineering and signal processing and so are familiar with the mathematics and the use of these methods for other types of signals, particularly the one-dimensional (time varying) electrical signals that make up much of our modern electronics.

The typical image analyst interested in applying computer methods to images for purposes of enhancement or measurement, however, may not be comfortable with the pages of mathematics (and intimidating notation) used in these discussions. Furthermore, he or she may not have the fortitude to relate these concepts to the operation of a dedicated image analysis computer. Unable to see the connection between the topics discussed and the typical image problems encountered in real life, the potential user might therefore find it easier to skip the subject. This is a loss, because the use of frequency space methods can offer benefits in many real-life applications, and it is not essential to deal deeply with the mathematics to arrive at a practical working knowledge of these techniques.

The Fourier transform and other frequency space transforms are applied to two-dimensional images for many different reasons. Some of these have little to do with the purposes of enhancing visibility and selection of features or structures of interest for measurement. For instance, some of these transform methods are used as a means of image compression, in which less data than the original image may be transmitted or stored. In this type of application, it is necessary to reconstruct the image (bring it back from the frequency to the spatial domain) for viewing. It is desirable to be able to accomplish both the forward and reverse transform rapidly and with a minimum loss of image quality. Image quality is a somewhat elusive concept that certainly includes the alteration of grey levels and color values, definition and location of feature boundaries, and introduction or removal of fine-scale texture in the image. Usually, the greater the degree of compression, the greater the loss of image fidelity, as shown in Chapter 2.

Speed is usually a less important concern to image measurement applications, since the acquisition and subsequent analysis of the images are likely to require some time anyway, but the computational advances (both in hardware and software or algorithms) made to accommodate the requirements of the data compression application help to shorten the time for some other processing operations, as well. On the other hand, the amount of image degradation that can be tolerated by most visual uses of the compressed and restored images is far greater than is usually acceptable for image analysis purposes. Consequently, the amount of image compression that can be achieved with minimal loss of fidelity is rather small, as discussed in Chapter 2.

In most cases, the transmission of an image from the point of acquisition to the computer used for analysis is not a major concern; therefore, we will ignore this entire subject here and assume that the transform retains all of the data, even if this means that there is no compression at all. Indeed, most of these methods are free from any data loss. The transform encodes the image information completely and it can be exactly reconstructed, at least to within the arithmetic precision of the computer being used (which is generally better than the precision of the original image sensor or analog-to-digital converter).

Although many different types of image transforms can be used, the best known (at least, the one with the most recognizable name) is the Fourier transform. This is due in part to the availability of a powerful and very efficient algorithm for computing it, known as the fast Fourier transform (FFT) (Cooley and Tukey, 1965; Bracewell, 1989), which we will encounter in due course. Although some programs actually perform the computation using the Fast Hartley Transform or FHT (Hartley, 1942; Bracewell, 1984, 1986; Reeves, 1990), the frequency space images are usually presented in the same form that the Fourier method would yield. For the sake of explanation, it is easiest to describe the better-known method.

The usual approach to developing the mathematical background of the Fourier transform begins with a one-dimensional waveform and then expands to two dimensions (an image). In principle, this can also be extended to three dimensions, although it becomes much more difficult to visualize or display. Three-dimensional transforms between the spatial domain (now a volume image constructed of voxels instead of pixels) and the three-dimensional frequency space are used, for example, in some tomographic reconstructions.

The mathematical development that follows has been kept as brief as possible, but if you suffer from "integral-o-phobia" then it is permitted to skip this section and go on to the examples and discussion, returning here only when (and if) a deeper understanding is desired.

The Fourier transform

Using a fairly standard nomenclature and symbology, begin with a function $f(x)$, where x is a real variable representing either time or distance in one direction across an image. It is very common to refer to this function as the spatial or time domain function and the transform F introduced below as the frequency space function. The function f is a continuous and well-behaved function. Do not be disturbed by the fact that in a digitized image, the values of x are not continuous but discrete (based on pixel spacing), and the possible brightness values are quantized as well. These values are considered to sample the real or analog image that exists outside the computer.

Fourier's theorem states that it is possible to form any one-dimensional function $f(x)$ as a summation of a series of sine and cosine terms of increasing frequency. The Fourier transform of the function $f(x)$ is written $F(u)$ and describes the amount of each frequency term that must be added together to make $f(x)$. It can be written as

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-2\pi i u x} dx \tag{1}$$

where i is (as usual) $\sqrt{-1}$. The use of the exponential notation relies on the mathematical identity (Euler's formula)

$$e^{-2\pi i u x} = \cos(2\pi u x) - i \sin(2\pi u x) \tag{2}$$

One of the very important characteristics of this transform is that given $F(u)$, it is possible to recover the spatial domain function $f(x)$ in the same way.

$$f(x) = \int_{-\infty}^{+\infty} F(u) e^{2\pi i u x} du \tag{3}$$

These two equations together comprise the forward and reverse Fourier transform. The function $f(x)$ is generally a real function, such as a time-varying voltage or a spatially-varying image brightness; however, the transform function $F(u)$ is generally complex, the sum of a real part R and an imaginary part I.

$$F(u) = R(u) + iI(u) \tag{4}$$

It is usually more convenient to express this in polar rather than Cartesian form

$$F(u) = |F(u)| \cdot e^{i\Phi(u)} \tag{5}$$

where $|F|$ is called the magnitude and ϕ is called the phase. The square of the magnitude $|F(u)|^2$ is commonly referred to as the power spectrum, or spectral density of $f(x)$.

The integrals from minus to plus infinity will in practice be reduced to a summation of terms of increasing frequency, limited by the finite spacing of the sampled points in the image. The discrete Fourier transform is written as

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \cdot e^{-i2\pi u x / N} \tag{6}$$

where N depends on the number of sampled points along the function $f(x)$, which are assumed to be uniformly spaced. Again, the reverse transform is similar (but not identical; note the absence of the $1/N$ term and the change in sign for the exponent).

$$f(x) = \sum_{u=0}^{N-1} F(u) \cdot e^{i2\pi u x / N} \tag{7}$$

The values of u from 0 to N–1 represent the discrete frequency components added together to construct the function $f(x)$. As in the continuous case, $F(u)$ is complex and may be written as real and imaginary or as magnitude and phase components.

The summation is normally performed over terms up to one-half the dimension of the image (in pixels), since it requires a minimum of two pixel brightness values to define the highest frequency present. This limitation is described as the Nyquist frequency. Because the summation has half as many terms as the width of the original image, but each term has a real and imaginary part, the

total number of values produced by the Fourier transform is the same as the number of pixels in the original image width, or the number of samples of a time-varying function. Since the original pixel values are usually small integers (0..255 for an 8-bit image), while the values produced by the Fourier transform are floating point numbers (and double precision ones in the best implementations), this actually represents an expansion in the storage requirements for the data.

In both the continuous and the discrete cases, a direct extension from one-dimensional functions to two- (or three-) dimensional ones can be made by substituting $f(x,y)$ for $f(x)$ and $F(u,v)$ for $F(u)$, and performing the summation or integration over two (or three) variables instead of one. Because the dimensions x,y,z are orthogonal, so are the u,v,w dimensions. This means that the transformation can be performed separately in each direction. For a two-dimensional image, for example, it would be possible to perform a one-dimensional transform on each horizontal line of the image, producing an intermediate result with complex values for each point. Then a second series of one-dimensional transforms can be performed on each vertical column, finally producing the desired two-dimensional transform.

The program fragment listed below shows how to compute the FFT of a function. It is written in Fortran, but can be translated into any other language (you may have to define a type to hold the complex numbers). On input to the subroutine, *F* is the array of values to be transformed (usually the imaginary part of these complex numbers will be 0) and *LN* is the power of 2 (up to 10 for the maximum 1024 in this implementation). The transform is returned in the same array *F*. The first loop reorders the input data, the second performs the successive doubling that is the heart of the FFT method, and the final loop normalizes the results.

```
      SUBROUTINE FFT(F,LN)
      COMPLEX F(1024),U,W,T,CMPLX
      PI=3.14159265
      N=2**LN
      NV2=N/2
      NM1=N-1
      J=1
      DO 3 I=1,NM1
            IF (I.GE.J) GOTO 1
            T=F(J)
            F(J)=F(I)
            F(I)=T
1           K=NV2
2           IF (K.GE.J) GOTO 3
            J=J-K
            K=K/2
            GOTO 2
3           J=J+K
      DO 5 L=1,LN
            LE=2**L
            LE1=LE/2
            U=(1.0,0.0)
            W=CMPLX(COS(PI/LE1),-SIN(PI/LE1))
            DO 5 J=1,LE1
                  DO 4 I=J,N,LE
                        IP=I+LE1
                        T=F(IP)*U
                        F(IP)=F(I)-T
```

```
4                           F(I)=F(I)+T
5                   U=U*W
      DO 6 I=1MN
6               F(I)=F(I)/FLOAT(N)
      RETURN
      END
```

Applying this one-dimensional transform to each row and then each column of a two-dimensional image is not the fastest way to perform the calculation, but it is by far the simplest and is actually used in many programs. A somewhat faster approach, known as a butterfly because it uses various sets of pairs of pixel values throughout the two-dimensional image, produces identical results. Storing the array of W values can also provide a slight increase in speed. Many software math packages include highly optimized FFT routines. Some of these allow for array sizes that are not an exact power of two.

The resulting transform of the original image into frequency space has complex values at each pixel. This is difficult to display in any readily interpretable way. In most cases, the display is based on only the magnitude of the value, ignoring the phase. If the square of the magnitude is used, this may be referred to as the image's power spectrum, because different frequencies are represented at different distances from the origin, different directions represent different orientations in the original image, and the power at each location shows how much of that frequency and orientation is present in the image. This display is particularly useful for isolating periodic structures or noise, which is discussed below; however, the power spectrum by itself cannot be used to restore the original image. The phase information is also needed, although it is rarely displayed and is usually difficult or impossible to interpret visually.

Fourier transforms of real functions

A common illustration in introductory-level math textbooks on the Fourier transform (which usually deal only with the one-dimensional case) is the quality of the fit to an arbitrary, but simple, function by the sum of a finite series of terms in the Fourier expansion. **Figure 1** shows the familiar case of a step function, illustrating the ability to add up a series of sine waves to produce the desired step. The coefficients in the Fourier series are the magnitudes of each increasing frequency needed to produce the fit. **Figure 2a** shows the result of adding together the first 4, 10, and 25 terms. Obviously, the greater the number of terms included, the better the fit (especially at the sharp edge). **Figure 2b** shows the same comparison for a ramp function. One of the important characteristics of the Fourier transform is that the first few terms include much of the information and adding more terms progressively improves the quality of the fit.

Notice in both of these cases that the function is actually assumed to be repetitive or cyclical. The fit goes on past the right and left ends of the interval as though the function were endlessly repeated in both directions. This is also the case in two dimensions; the image in the spatial domain is effectively one tile in an endlessly repeating pattern. If the right and left edges or the top and bottom edges of the image are different, this can produce very noticeable effects in the resulting transform. One solution is to embed the image in a larger one consisting of either zeroes or the average brightness value of the pixels. This kind of padding makes the image twice as large in each direction, requiring four times as much storage and calculation. It is needed particularly when correlation is performed, as discussed below. Padding out to a larger size is also the simplest way to deal with an image that does not have dimensions that are an exact power of two.

The magnitude of the Fourier coefficients from the transforms shown in **Figure 2** is plotted as amplitude vs. frequency (**Figure 3**). Notice that the step function consists only of odd terms, while the magnitudes for the ramp function transform decrease smoothly. Rather than the magnitudes, it

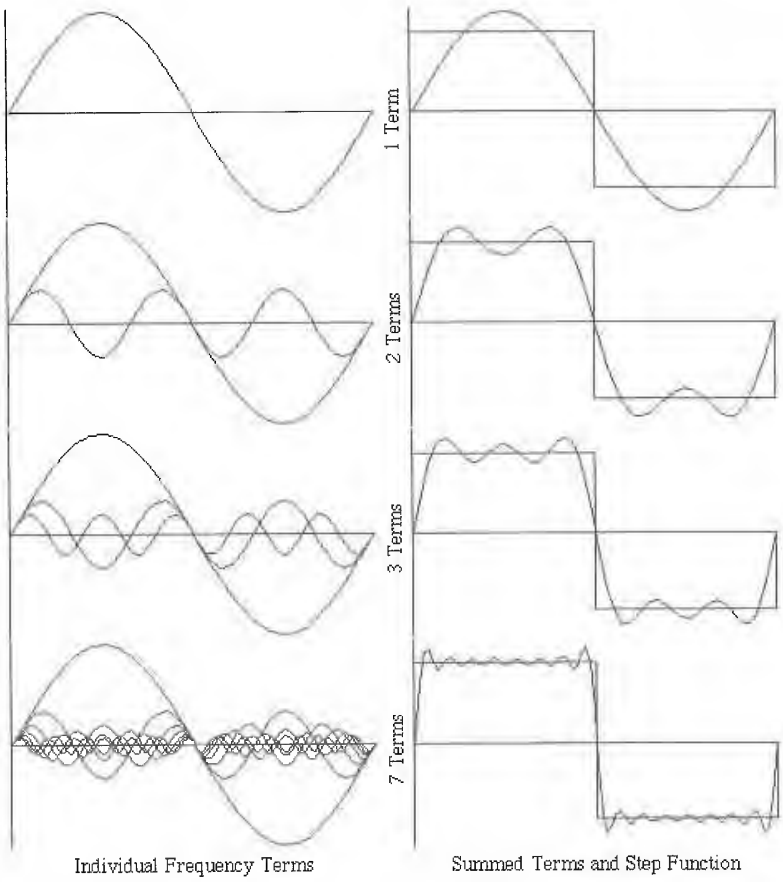
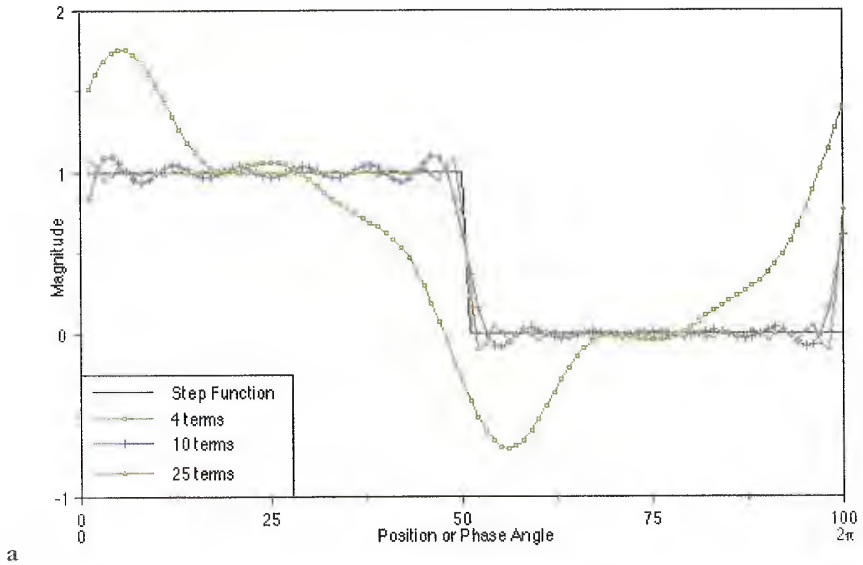


Figure 1. *Summation of Fourier frequency terms to fit a simple step function.*

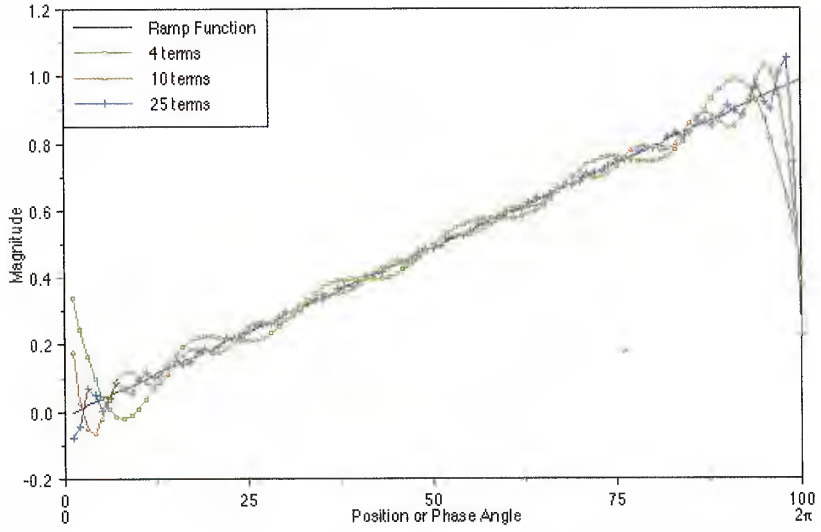
is somewhat more common to plot the power spectrum of the transform, and to plot it as a symmetric function extending to both sides of the origin (zero frequency, or the DC level). As noted above, the power is simply the square of the magnitude. Because the range of values can be very large, the power spectrum is often plotted with a logarithmic or other compressed vertical scale to show the smaller terms usually present at high frequencies, along with the lower frequency terms.

Figure 4 reiterates the duality of the Fourier transform process. The spatial and frequency domains show the information in very different ways, but the information is the same. Of course, the plot of amplitude or power in the frequency transform does not show the important phase information, but we understand that the values are actually complex. Shifting the spatial domain image does not alter the amplitude values, but does change the phase values for each sinusoidal component.

It is important to recall, in examining these transforms, that the axes represent frequency. The low-frequency terms provide the overall shape of the function, while the high-frequency terms are needed to sharpen edges and provide fine detail. The second point to be kept in mind is that these terms are independent of each other (this is equivalent to the statement that the basis functions — the sinusoidal waves — are orthogonal). Performing the transform to determine coefficients to higher and higher frequencies does not change the previous ones, and selecting any particular range of terms to reconstruct the function will do so to the greatest accuracy possible with those frequencies.



a



b

Figure 2. *Match between a step function* ***(a)*** *and a ramp function* ***(b)*** *and the first 4, 10, and 25 Fourier terms.*

Proceeding to two dimensions, **Figure 5** shows four images of perfectly sinusoidal variations in brightness. The first three vary in spacing (frequency) and orientation; the fourth is the superposition of all three. For each, the two-dimensional frequency transform is particularly simple. Each of the pure tones has a transform consisting of a single point (identifying the frequency and orientation). Because of the redundancy of the plotting coordinates, the point is shown in two symmetrical locations around the origin, which by convention lies at the center of the power spectrum plot.

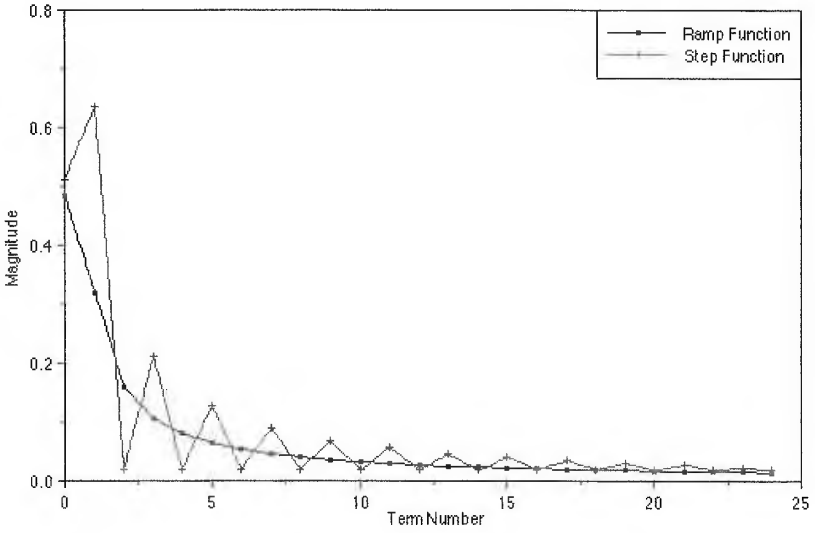


Figure 3. *Magnitude of the first 25 Fourier terms fit to the step and ramp in* ***Figure 2.***

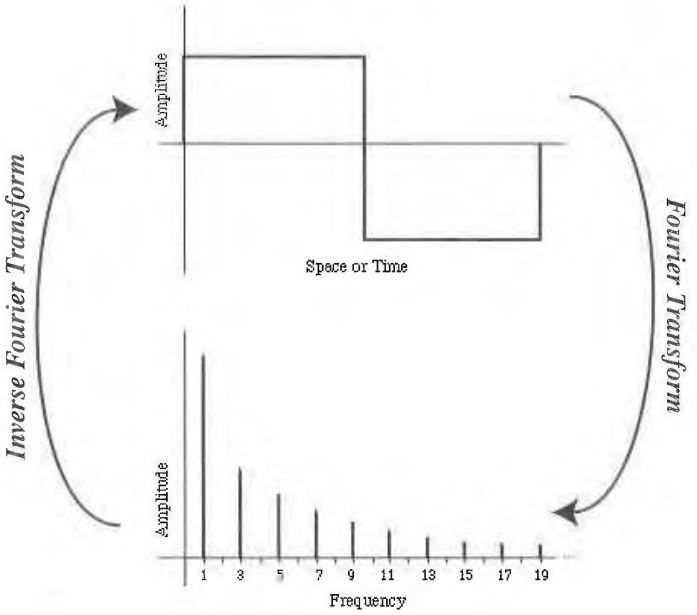


Figure 4. *Role of the forward and inverse transform and the spatial and frequency domain representations of a step function.*

Two-dimensional power spectra are easiest to describe using polar coordinates. The frequency increases with radius ρ, and the orientation depends on the angle θ. It is common to display the two-dimensional transform with the frequencies plotted from the center of the image, which is consequently redundant (the top and bottom or left and right halves are simply duplicates, with symmetry about the origin). In some cases, this image is shifted so that the origin is at the corners of the image and the highest frequencies are in the center. One format can be converted to the other by swapping quadrants of the display. For the purposes of image processing (removing or selecting specific frequencies, etc.) the display with the origin centered is simplest to use and has been adopted here.