Biological Image Analysis Primer

Erik Meijering* and Gert van Cappellen#

*Biomedical Imaging Group
#Applied Optical Imaging Center
Erasmus MC – University Medical Center
Rotterdam, the Netherlands

Typeset by the authors using $\LaTeX2_{\varepsilon}$.

Preface

This booklet was written as a companion to the introductory lecture on digital image analysis for biological applications, given by the first author as part of the course *In Vivo Imaging – From Molecule to Organism*, which is organized annually by the second author and other members of the applied Optical Imaging Center (aOIC) under the auspices of the postgraduate school Molecular Medicine (MolMed) together with the Medical Genetics Center (MGC) of the Erasmus MC in Rotterdam, the Netherlands. Avoiding technicalities as much as possible, the text serves as a primer to introduce those active in biological investigation to common terms and principles related to image processing, image analysis, visualization, and software tools. Ample references to the relevant literature are provided for those interested in learning more.

Acknowledgments The authors are grateful to Adriaan Houtsmuller, Niels Galjart, Jeroen Essers, Carla da Silva Almeida, Remco van Horssen, and Timo ten Hagen (Erasmus MC, Rotterdam, the Netherlands), Floyd Sarria and Harald Hirling (Swiss Federal Institute of Technology, Lausanne, Switzerland), Anne McKinney (McGill University, Montreal, Quebec, Canada), and Elisabeth Rungger-Brändle (University Eye Clinic, Geneva, Switzerland) for providing image data for illustrational purposes. The contribution of the first author was financially supported by the Netherlands Organization for Scientific Research (NWO), through VIDI-grant 639.022.401.

Contents

Preface **3**

1 Introduction **5**

1.1 Definition of Common Terms 5

1.2 Historical and Future Perspectives 7

2 Image Processing **10**

2.1 Intensity Transformation 10

2.2 Local Image Filtering 12

2.3 Geometrical Transformation 15

2.4 Image Restoration 15

3 Image Analysis **19**

3.1 Colocalization Analysis 19

3.2 Neuron Tracing and Quantification 21

3.3 Particle Detection and Tracking 23

3.4 Cell Segmentation and Tracking 25

4 Visualization **26**

4.1 Volume Rendering 26

4.2 Surface Rendering 28

5 Software Tools **29**

References **30**

Index **35**

1 Introduction

Images play an increasingly important role in many fields of science and its countless applications. Biology is without doubt one of the best examples of fields that have come to depend heavily upon images for progress. As a consequence of the ever increasing resolving power and efficiency of microscopic image acquisition hardware and the rapidly decreasing cost of mass storage and communication media, biological image data sets grow exponentially in size and carry more and more information. Extracting this information by visual inspection and manual measurement is labor intensive, and the results are potentially inaccurate and poorly reproducible. Hence there is a growing need for computerized image processing and analysis, not only to cope with the rising rate at which images are acquired, but also to reach a higher level of sensitivity, accuracy, and objectivity than can be attained by human observers.[57] It seems inevitable, therefore, that in the future biologists will increasingly resort to automated image processing and analysis technology in exploiting their precious data. In order to benefit from this technology, it is of paramount importance to have at least a basic understanding of its underlying principles: biologically highly relevant information may easily go unnoticed or get destroyed (or may even be created ex nihilo!) by improper use of image processing and analysis tools. This booklet, which summarizes and updates earlier (partial) reviews in the field,[17,23,29,31,73] was written with the aim of providing the biologist with the minimum know-how to get started. We begin by (re)defining common terminology and putting image processing and analysis into historical and future perspective.

1.1 Definition of Common Terms

Because of the rapid rise of imaging technology in the sciences as well as in everyday life, several terms have become very fashionable, even among a large percentage of the general public, but whose precise meanings appear to vary. Before we go into details, it is necessary to (re)define these terms to avoid confusion. The word “image” itself, for starters, already has at least five different meanings. In the most general sense of the word, an image is a representation of something else. Depending on the type of representation, images can be divided into several classes.[15] These include images perceivable by the human eye, such as pictures (photographs, paintings, drawings) or those formed by lenses or holograms (optical images), as well as nonvisible images, such as continuous or discrete mathematical functions, or distributions of measurable physical properties. In the remainder of this booklet, when we speak of an image, we mean a *digital* image, defined as a representation obtained by taking finitely many samples, expressed as numbers that can take on only finitely many values. In the present context of (in vivo) biological imaging, the objects we make representations of are (living) cells and molecules, and the images are usually acquired by taking samples of (fluorescent or other) light at given intervals in space and time and wavelength.

Mathematically speaking, images are matrices, or discrete functions, with the number of dimensions typically ranging from one to five and where each dimension corresponds to a parameter, a degree of freedom, or a coordinate needed to uniquely locate a sample value (Figure 1.1). A description of how, exactly, these matrices are obtained and how they relate to the physical world is outside the scope of this booklet but can be found elsewhere.[61] Each sample corresponds to what we call an image *element*. If the image is spatially 2D, the elements are usually called pixels (short for "picture elements," even though the image need not necessarily be a picture). In the case of spatially 3D images, they are called voxels ("volume elements"). However, since data sets in optical microscopy usually consist of series of 2D images (time frames or optical sections) rather than truly volumetric images, we refer to an image element of any dimensionality as a "pixel" here.

Image processing is defined as the act of subjecting an image to a series of operations that alter its form or its value. The result of these operations is again an image. This is distinct from image *analysis*, which is defined as the act of measuring (biologically) meaningful object features in an image. Measurement results can be either qualitative (categorical data) or quantitative (numerical data) and both types of results can be either subjective (dependent on the personal feelings and prejudices of the subject doing the measurements) or objective (solely dependent on the object itself and the measurement method). In many fields of research there is a tendency towards quantification and objectification, feeding the need for fully automated image analysis methods. Ultimately, image analysis results should lead to understanding the nature and interrelations of the objects being imaged. This requires not only measurement data, but also reasoning about the data and making inferences, which involves some form of intelligence and cognitive processing. Computerizing these aspects of human vision is the long-term goal of computer vision. Finally we mention computer graphics and visualization. These terms are strongly related,[78] but strictly speaking the former refers to the process of *generating* images for display of given data using a computer, while the latter

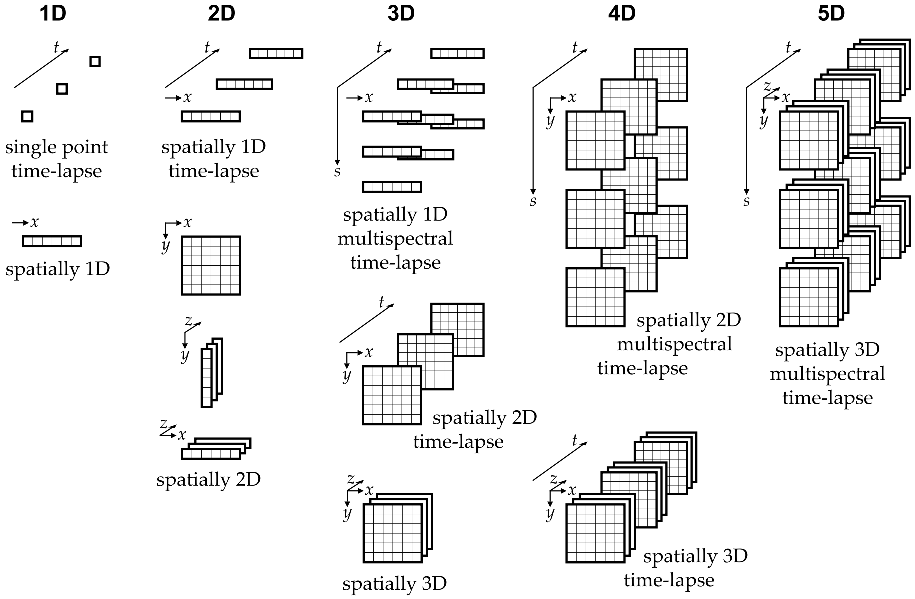


Figure 1.1 Images viewed as matrices. The overview is not meant to be exhaustive but reflects some of the more frequently used modes of image acquisition in biological and medical imaging, where the number of dimensions is typically one to five, with each dimension corresponding to an independent physical parameter: three (usually denoted x, y, z) to space, one (usually denoted t) to time, and one to wavelength, or color, or more generally to any spectral parameter (we denote this dimension s here). In other words, images are discrete functions, $I(x, y, z, t, s)$, with each set of coordinates yielding the value of a unique sample (indicated by the small squares, the number of which is obviously arbitrary here). Note that the dimensionality of an image (indicated in the top row) is given by the number of coordinates that are varied during acquisition. To avoid confusion in characterizing an image, it is advisable to add adjectives indicating which dimensions were scanned, rather than mentioning just dimensionality. For example, a 4D image may either be a spatially 2D multispectral time-lapse image, or a spatially 3D time-lapse image.

is more concerned with *transforming* data to enable rendering and exploration. An illustration of all these terms (Figure 1.2) may help memorize their meaning. In this booklet we focus mainly on image processing and image analysis and also briefly touch upon visualization.

1.2 Historical and Future Perspectives

The idea of processing images by computer was conceived in the late 1950s, and over the decades to follow was further developed and applied to such diverse fields as astronomy and space exploration, remote sensing for earth resources research, and diagnostic radiology, to mention but a few. In our present-day life, image processing and analysis technology is employed in

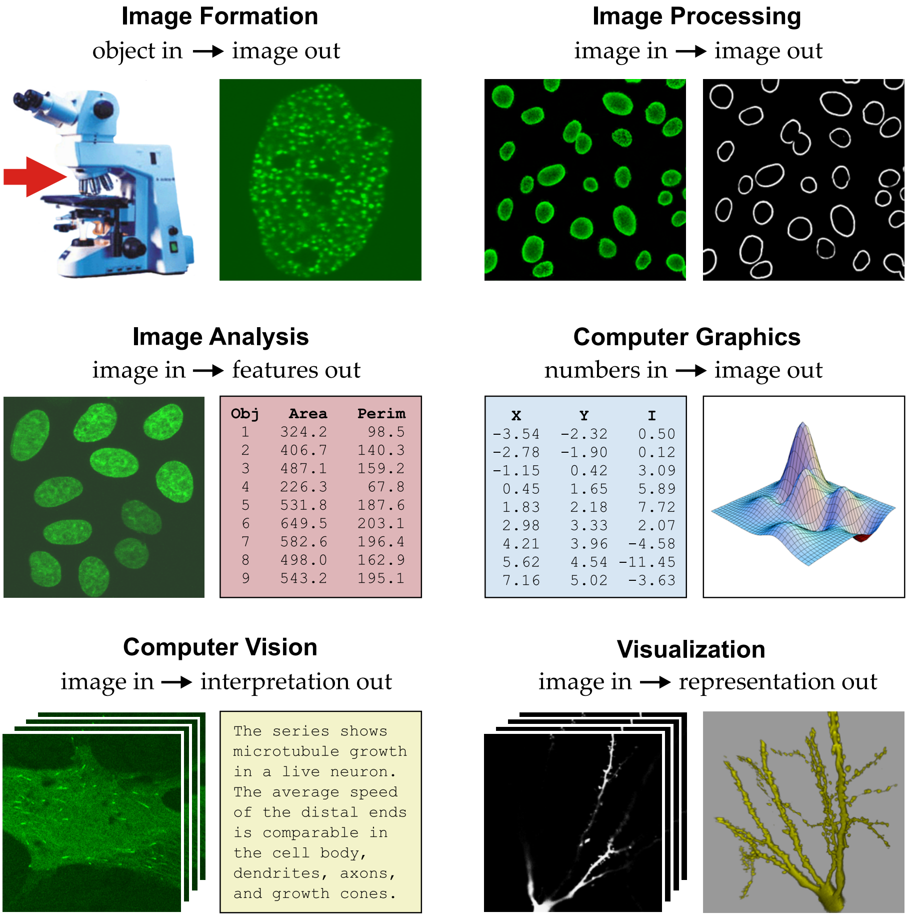


Image Formation

object in → image out

Image Processing

image in → image out

Image Analysis

image in → features out

Obj	Area	Perim
1	324.2	98.5
2	406.7	140.3
3	487.1	159.2
4	226.3	67.8
5	531.8	187.6
6	649.5	203.1
7	582.6	196.4
8	498.0	162.9
9	543.2	195.1

Computer Graphics

numbers in → image out

X	Y	I
-3.54	-2.32	0.50
-2.78	-1.90	0.12
-1.15	0.42	3.09
0.45	1.65	5.89
1.83	2.18	7.72
2.98	3.33	2.07
4.21	3.96	-4.58
5.62	4.54	-11.45
7.16	5.02	-3.63

Computer Vision

image in → interpretation out

The series shows microtubule growth in a live neuron. The average speed of the distal ends is comparable in the cell body, dendrites, axons, and growth cones.

Visualization

image in → representation out

Figure 1.2 Illustration of the meaning of commonly used terms. The process of digital image formation in microscopy is described in other books. Image processing takes an image as input and produces a modified version of it (in the case shown, the object contours are enhanced using an operation known as edge detection, described in more detail elsewhere in this booklet). Image analysis concerns the extraction of object features from an image. In some sense, computer graphics is the inverse of image analysis: it produces an image from given primitives, which could be numbers (the case shown), or parameterized shapes, or mathematical functions. Computer vision aims at producing a high-level interpretation of what is contained in an image. This is also known as image understanding. Finally, the aim of visualization is to transform higher-dimensional image data into a more primitive representation to facilitate exploring the data.

surveillance, forensics, military defense, vehicle guidance, document processing, weather prediction, quality inspection in automated manufacturing processes, et cetera. Given this enormous success, one might think that computers will soon be ready to take over most human vision tasks, also in biological investigation. This is still far from becoming a reality, however. After 50 years of research, our knowledge of the human visual system and how to excel it is still very fragmentary and mostly confined to the early stages, that is to image processing and image analysis. It seems reasonable to predict that another 50 years of multidisciplinary efforts involving vision research, psychology, mathematics, physics, computer science, and artificial intelligence will be required before we can begin to build highly sophisticated computer vision systems that outperform human observers in all respects. In the meantime, however, currently available methods may already be of great help in reducing manual labor and increasing accuracy, objectivity, and reproducibility.

2
Image Processing

Several fundamental image processing operations have been developed over the past decades that appear time and again as part of more involved image processing and analysis procedures. Here we discuss four classes of operations that are most commonly used: intensity transformation, linear and non-linear image filtering, geometrical transformation, and image restoration. For ease of illustration, examples are given for spatially 2D images, but they easily extend to higher-dimensional images. Also, the examples are confined to intensity (gray-scale) images. In the case of multispectral images, some operations may need to be applied separately to each channel, possibly with different parameter settings. A more elaborate treatment of the mentioned (and other) basic image processing operations can be found in the cited works and in a variety of textbooks.[7,15,33,38,39,72,81]

2.1 Intensity Transformation

Among the simplest image processing operations are those that pass along each image pixel and produce an output value that depends only on the corresponding input value and some mapping function. These are also called point operations. If the mapping function is the same for each pixel, we speak of a *global* intensity transformation. An infinity of mapping functions can be devised, but most often a (piecewise) linear function is used, which allows easy (interactive) adjustment of image brightness and contrast. Two extremes of this operation are intensity inversion and intensity thresholding. The latter is one of the easiest (and most error-prone!) approaches to divide an image into meaningful objects and background, a task referred to as image segmentation. Logarithmic mapping functions are also sometimes used to better match the light sensitivity of the human eye when displaying images. Another type of intensity transformation is pseudo-coloring. Since the human eye is more sensitive to changes in color than to changes in intensity, more detail may be perceived when mapping intensities to colors.

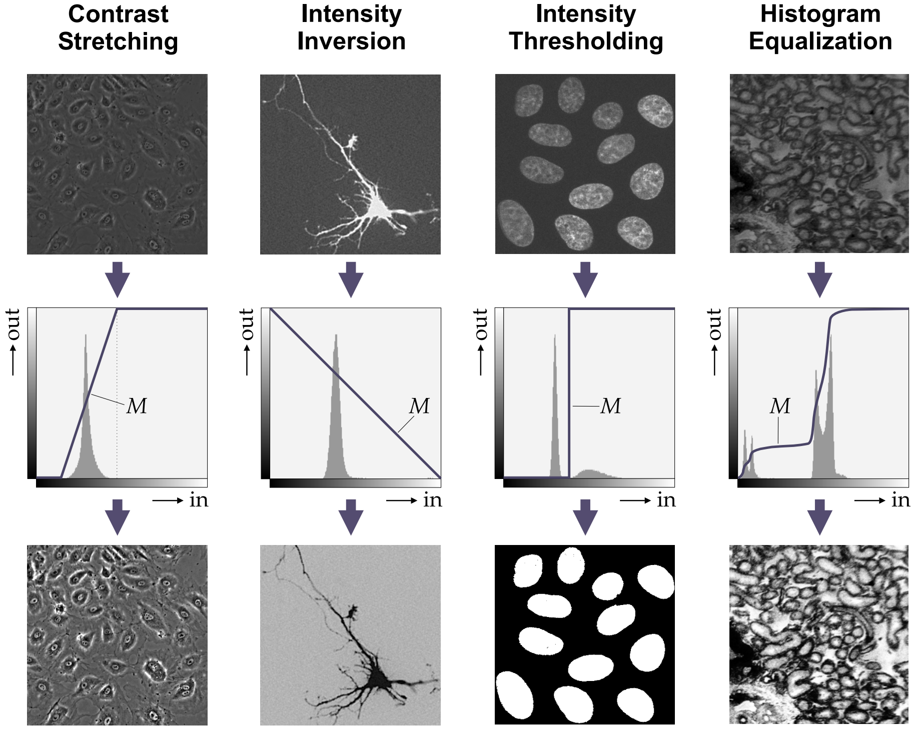


Figure 2.1 Examples of intensity transformations based on a global mapping function: contrast stretching, intensity inversion, intensity thresholding, and histogram equalization. The top row shows the images used as input for each of the transformations. The second row shows for each image the mapping function used (denoted M), with the histogram of the input image shown in the background. The bottom row shows for each input image the corresponding output image resulting from applying the mapping function: $O(x, y) = M(I(x, y))$. It is clear from the mapping functions that contrast stretching and histogram equalization both distribute the most frequently occurring intensities over a wider range of values, thereby increasing image contrast. The former transformation is suitable in the case of unimodal histograms, whereas the latter is particularly suited for images having multimodal histograms.

Mapping functions usually have one or more parameters that need to be specified. A useful tool for establishing suitable values for these parameters is the intensity histogram, which lists the frequency (number of occurrences) of each intensity value in the image (Figure 2.1). For example, if the histogram indicates that intensities occur mainly within a limited range of values, the contrast may be improved considerably by mapping this input range to the full output range (this operation is therefore called contrast stretching). Instead of being supplied by the user, mapping functions may also be computed automatically from the histogram. This is done, for example, in histogram equalization, where the mapping function is derived from the cumulative histogram of the input image, causing the histogram of the output image to be more uniformly distributed. In cases where the intensity histogram is multimodal, this operation may be more effective in improving image contrast between different types of adjacent tissues than simple contrast stretching. The histogram may also be used to determine a global threshold value,[30] for example the value corresponding to the minimum between the two major modes of the (smoothed) histogram, or the value that maximizes the interclass variance (or, equivalently, minimizes the intraclass variance) of object (above the threshold) and background (below the threshold) values.[60]

2.2 Local Image Filtering

Instead of considering just the corresponding input pixel when computing a value for each output pixel (as in intensity transformation), one could also take into account the values of adjacent input pixels. Image processing operations based on this principle are called neighborhood operations, or image filtering operations, as they are usually designed to filter out (enhance or reduce) specific image information. They can be classified into linear versus nonlinear. Linear filtering operations compute the output pixel value as a linear combination (weighing and summation) of the values of the corresponding input pixel and its neighbors. This process can be described mathematically as a convolution operation, and the mask (or filter) specifying the weight factor for each pixel value is accordingly called a convolution kernel. Examples of kernels include averaging, sharpening, smoothing, and derivative filters of varying sizes (Figure 2.2). The latter can be used, for example, to detect object edges, by a procedure known as Canny edge detection.[12] Convolution of an image with a kernel is equivalent to multiplication of the respective Fourier transformations, followed by inverse transformation of the result.[11] Certain filtering operations, for example to remove specific intensity oscillations, are better done in the Fourier domain, as the corresponding convolution kernel would be very large, requiring excessive computation times.

Nonlinear filtering operations combine neighboring input pixel values in a nonlinear fashion in producing an output pixel value. They can not be described as a convolution process. Examples include median filtering (which for each output pixel computes the value as the median of the corresponding input values in a neighborhood of given size) and min-filtering or max-filtering (where the output value is computed as, respectively, the minimum or the maximum value in a neighborhood around the corresponding input pixel). Another class of nonlinear filtering operations comes from the field of mathematical morphology[80] and deals with the processing of object shape. Of particular interest to image analysis is binary morphology, which applies to two-valued (binary) images and is often applied as postprocessing step to clean up imperfect segmentations. Morphological filtering is described in terms of the interaction of an image and a structuring element (a small

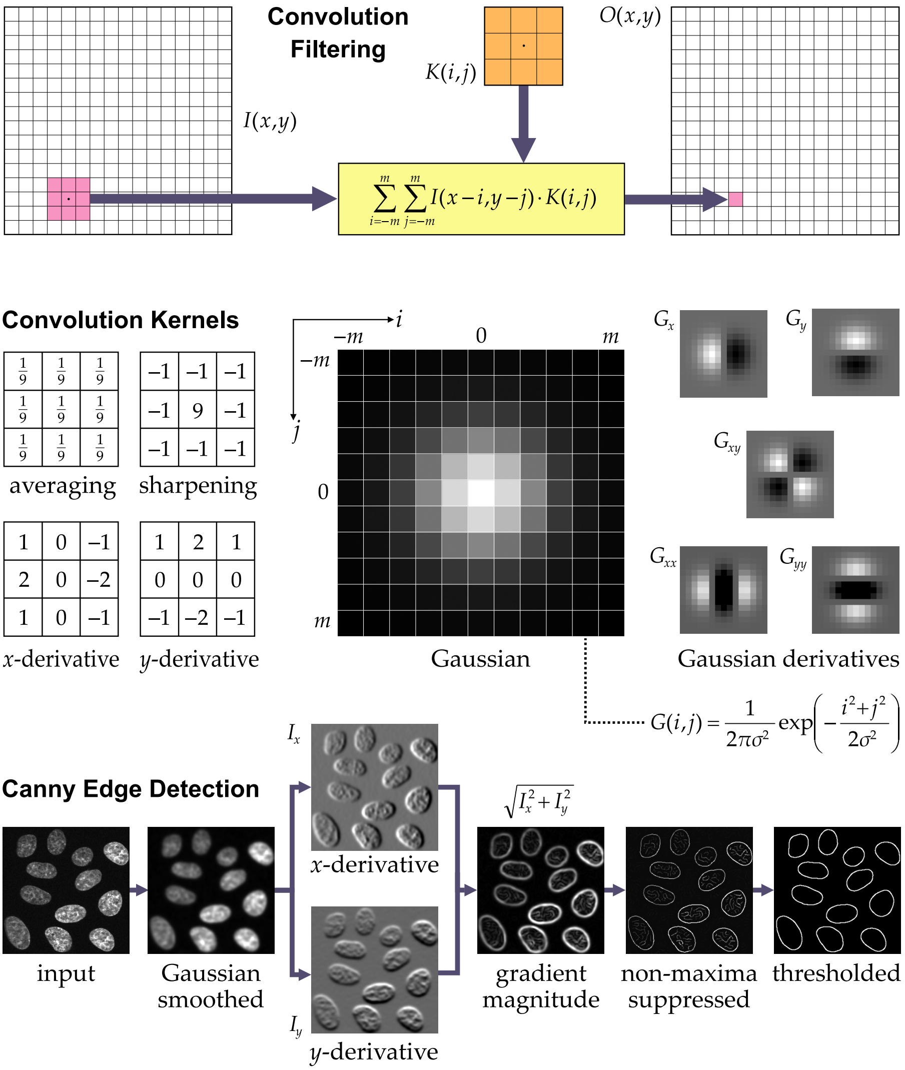


Figure 2.2 Principles and examples of convolution filtering. The value of an output pixel is computed as a linear combination (weighing and summation) of the value of the corresponding input pixel and of its neighbors. The weight factor assigned to each input pixel is given by the convolution kernel (denoted K). In principle, kernels can be of any size. Examples of commonly used kernels of size 3 × 3 pixels include the averaging filter, the sharpening filter, and the Sobel x- or y-derivative filters. The Gaussian filter is often used as a smoothing filter. It has a free parameter (standard deviation σ) which determines the size of the kernel (usually cut off at $m = 3\sigma$) and therefore the degree of smoothing. The derivatives of this kernel are often used to compute image derivatives at different scales, as for example in Canny edge detection. The scale parameter, σ, should be chosen such that the resulting kernel matches the structures to be filtered.

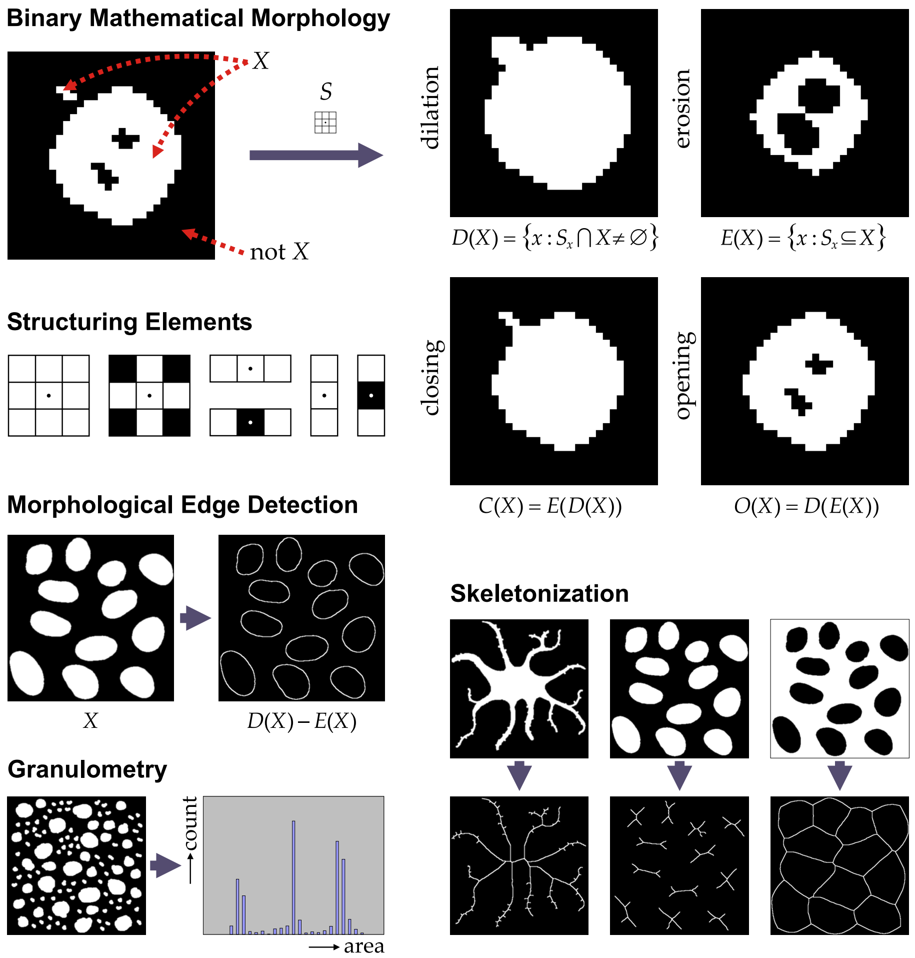


Figure 2.3 Principles and examples of binary morphological filtering. An object in the image is described as the set (denoted X) of all coordinates of pixels belonging to that object. Morphological filters process this set using a second set, known as the structuring element (denoted S). Here the discussion is limited to structuring elements that are symmetrical with respect to their center element, $s = (0, 0)$, indicated by the dot. In that case, the dilation of X is defined as the set of all coordinates x for which the cross section of S placed at x (denoted S_x) with X is not empty, and the erosion of X as the set of all x for which S_x is a subset of X. A dilation followed by an erosion (or vice versa) is called a closing (versus opening). These operations are named after the effects they produce, as illustrated. Many interesting morphological filters can be constructed by taking differences of two or more operations, such as in morphological edge detection. Other applications include skeletonization, which consists in a sequence of thinning operations producing the basic shape of objects, and granulometry, which uses a family of opening operations with increasingly larger structuring elements to compute the size distribution of objects in an image.

mask reminiscent of a convolution kernel in the case of linear filtering). Basic morphological operations include erosion, dilation, opening, and closing (Figure 2.3). By combining these we can design many interesting filters to prepare for (or even perform) image analysis. For example, subtracting the results of dilation and erosion yields object edges. Or by analyzing the results of a family of openings, using increasingly larger structuring elements, we may perform size distribution analysis or granulometry of image structures. Another operation that is frequently used in biological shape analysis[24,35,36,91] is skeletonization, which yields the basic shape of segmented objects.

2.3 Geometrical Transformation

In many situations it may occur that the images acquired by the microscope are spatially distorted or lack spatial correspondence. In colocalization experiments, for example, images of the same specimen imaged at different wavelengths may show mismatches due to chromatic aberration. Nonlinear magnification from the center to the edge of the field of view may result in deformations known as barrel distortion or pincushion distortion. In live cell experiments, one may be interested in studying specific intracellular components over time, which appear in different places in each image due to the motion of the cell itself. Such studies require image alignment, also referred to as image registration in the literature.[45,64,82] Other studies, for example karyotype analyses, require the contents of images to be reformatted to some predefined configuration. This is also known as image reformatting.

In all such cases, the images (or parts thereof) need to undergo spatial (or geometrical) transformation prior to further processing or analysis. There are two aspects to this type of operation: coordinate transformation and image resampling. The former concerns the mapping of input pixel positions to output pixel positions (and vice versa). Depending on the complexity of the problem, one commonly uses a rigid, or an affine, or a curved transformation (Figure 2.4). Image resampling concerns the issue of computing output pixel values based on the input pixel values and the coordinate transformation. This is also known as image interpolation, for which many methods exist. It is important to realize that every time an image is resampled, some information is lost. Studies in the field of medical imaging[52,85] have indicated that higher-order spline interpolation methods (for example cubic splines) are much less harmful in this regard than some standard approaches, such as nearest-neighbor interpolation and linear interpolation, although the increased computational load may be prohibitive in some applications.

2.4 Image Restoration

There are many factors in the acquisition process that cause a degradation of image quality in one way or another, resulting in a corrupted view of reality.

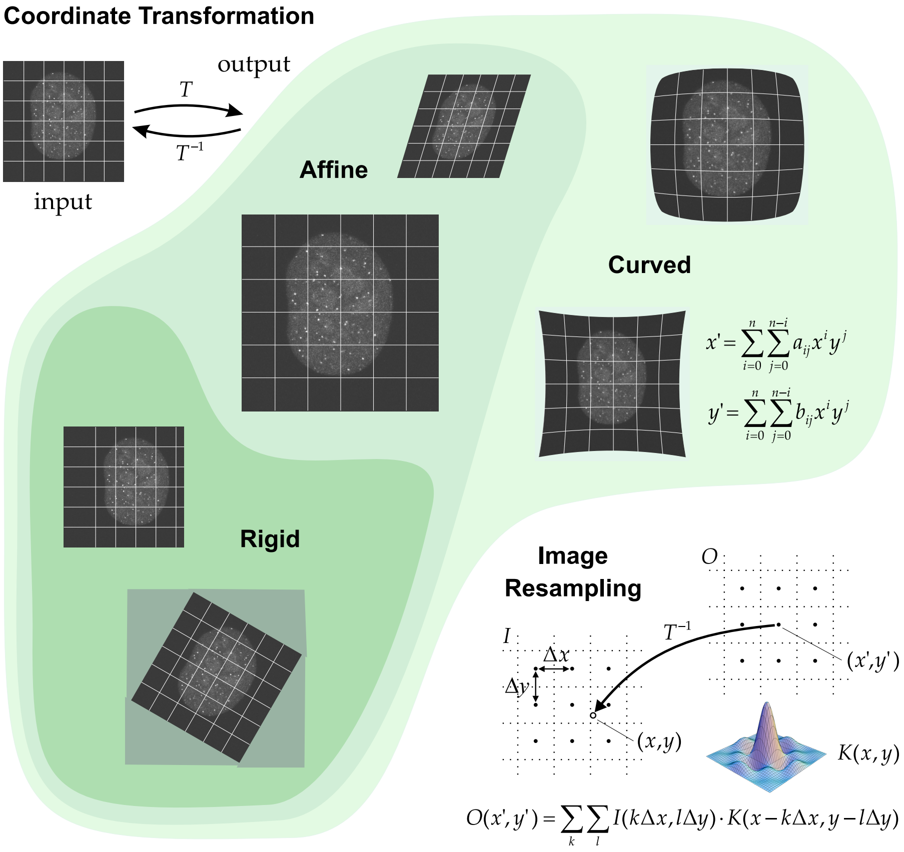

Coordinate Transformation
output
T
T^{-1}
input
Affine
Curved
$x' = \sum_{i=0}^{n}\sum_{j=0}^{n-i} a_{ij} x^i y^j$
$y' = \sum_{i=0}^{n}\sum_{j=0}^{n-i} b_{ij} x^i y^j$
Rigid
Image Resampling
O
T^{-1}
I
Δx
Δy
(x',y')
(x,y)
$K(x,y)$
$O(x',y') = \sum_k \sum_l I(k\Delta x, l\Delta y) \cdot K(x - k\Delta x, y - l\Delta y)$

Figure 2.4 Geometrical image transformation by coordinate transformation and image resampling. The former is concerned with how input pixel positions are mapped to output pixel positions. Many types of transformations (denoted T) exist. The most frequently used types are (in increasing order of complexity) rigid transformations (translations and rotations), affine transformations (rigid transformations plus scalings and skewings), and curved transformations (affine transformations plus certain nonlinear or elastic deformations). These are defined (or can be approximated) by polynomial functions (with degree n depending on the complexity of the transformation). Image resampling concerns the computation of the pixel values of the output image (denoted O) from the pixel values of the input image (denoted I). This is done by using the inverse transformation (denoted T^{-1}) to map output grid positions (x', y') to input positions (x, y). The value at this point is then computed by interpolation from the values at neighboring grid positions, using a weighing function, also known as the interpolation kernel (denoted K).

Chromatic and other aberrations in the imaging optics may result in spatial distortions (already mentioned). These may be corrected by image registration methods. Certain illumination modes result in (additive) intensity gradients or shadows, which may be corrected by subtracting an image showing only these phenomena, not the specimen. This is known as background sub-

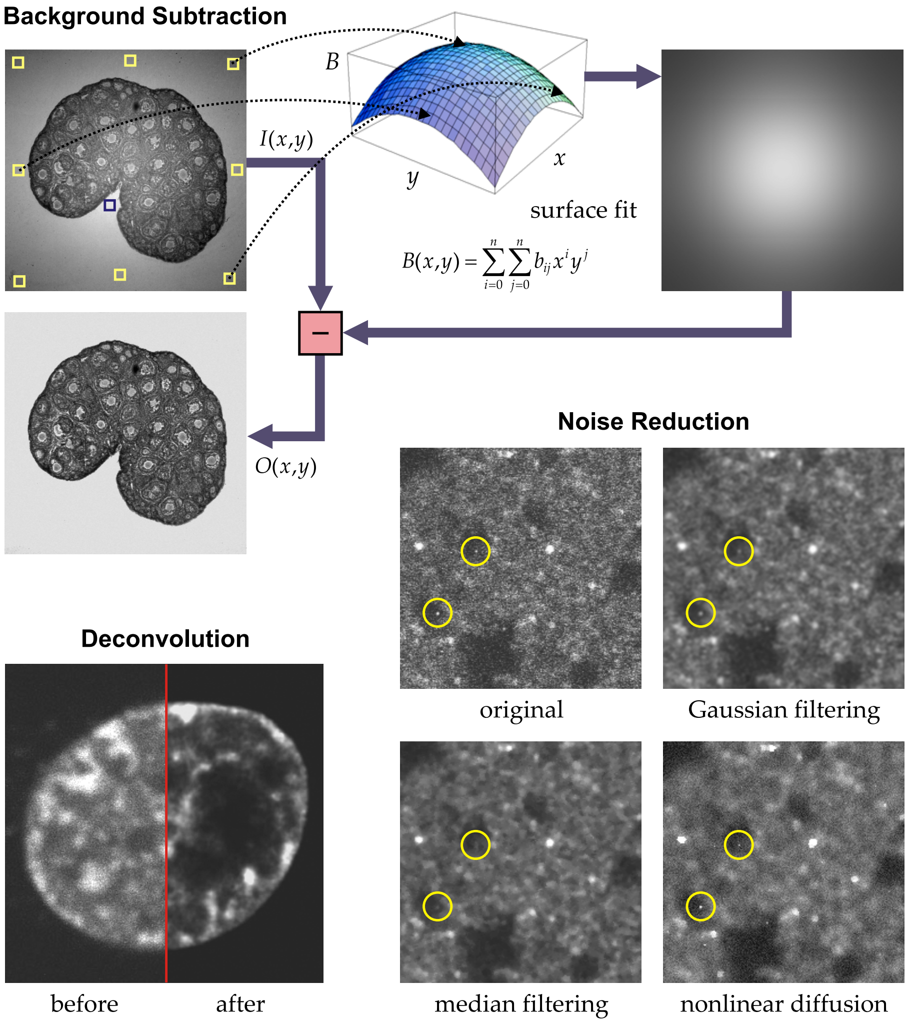


Figure 2.5 Examples of the effects of image restoration operations: background subtraction, noise reduction, and deconvolution. Intensity gradients may be removed by subtracting a background image. In some cases, this image may be obtained from the raw image itself by mathematically fitting a polynomial surface function through the intensities at selected points (indicated by the squares) corresponding to the background. Several filtering methods exist to reduce noise. Gaussian filtering blurs not only noise but all image structures. Median filtering is somewhat better at retaining object edges but has the tendency to eliminate very small objects (compare the circles in each image). Needless to say, the magnitude of these effects depends on the filter size. Nonlinear diffusion filtering was designed specifically to preserve object edges while reducing noise. Finally, deconvolution methods aim to undo the blurring effects of the microscope optics and to restore small details. More sophisticated methods are also capable of reducing noise.

traction. If it is not possible to capture a background image, it may in some cases be obtained from the image to be corrected (Figure 2.5). Another major source of intensity corruption is noise, due to the quantum nature of light (signal-dependent noise, following Poisson statistics) and imperfect electronics (mostly signal-independent, Gaussian noise). One way to reduce noise is local averaging of pixel intensities using a uniform or Gaussian convolution filter. However, while improving the overall signal-to-noise ratio (SNR), this operation also blurs other image structures. Median filtering is an effective way to remove shot noise (as caused, for example, by bright or dark pixels). It should be used with great care, however, when small objects are studied (such as in particle tracking), as these may also be (partially) filtered out. A more sophisticated technique is nonlinear diffusion filtering,[63] which smoothes noise while preserving sharpness at object edges, by taking into account local image properties (notably the gradient magnitude).

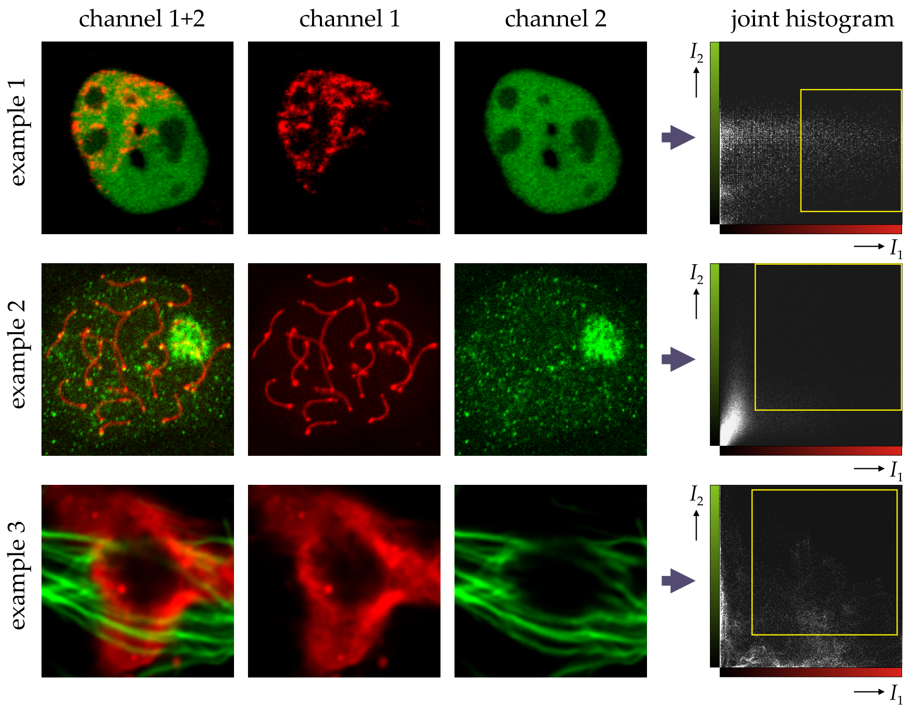
Especially widefield microscopy images may suffer from excessive blurring due to out-of-focus light. But even in confocal microscopy, where most of these effects are suppressed, images are blurred due to optical diffraction effects.[10,34] To good accuracy, these effects may be modeled mathematically as a convolution of the true optical image with the 3D point-spread function (PSF) of the microscope optics. Methods that try to undo this operation, in other words that try in every point in the image to reassign light to the proper in-focus location, are therefore called deconvolution methods.[40,61,90] Simple examples include nearest-neighbor or multi-neighbor deblurring and Fourier-based inverse filtering methods. These are computationally fast but have the tendency to amplify noise. More sophisticated methods, which also reduce noise, are based on iterative regularization and other (constrained or statistical) iterative algorithms. In principle, deconvolution preserves total signal intensity while improving contrast by restoring signal position. Therefore it is often desirable prior to quantitative image analysis.

3
Image Analysis

The image processing operations described in the previous chapter are important in enhancing or correcting image data, but by themselves do not answer any specific biological questions. Addressing such questions requires much more involved image processing and analysis algorithms, consisting of series of operations working closely together in “interrogating” the data and extracting biologically meaningful information. Because of the complexity of biological phenomena and the variability (or even ambiguity) of biological image data, many analysis tasks are difficult to automate fully and require expert user input or interaction. In contrast with most image processing operations, image analysis methods are therefore often semi-automatic. Here we briefly describe state-of-the-art methods for some of the most relevant and pressing image analysis problems in biology today: colocalization analysis, neuron tracing and quantification, and the detection or segmentation, tracking and motion analysis of particles and cells. Several technical challenges in these areas are still vigorously researched.

3.1 Colocalization Analysis

An interesting question in many biological studies is to what degree two or more molecular species (typically proteins) are active in the same specimen. This co-occurrence phenomenon can be imaged by using a different fluorescent label for each species, combined with multicolor optical microscopy imaging. A more specific question is whether or not proteins reside in the same (or proximate) physical locations in the specimen. This is the problem of colocalization. For such experiments it is of paramount importance that the emission spectra (rather than just the peak wavelengths) of the fluorophores are sufficiently well separated and that the correct filter sets are used during acquisition to reduce artifacts due to spectral bleed-through or fluorescence resonance energy transfer (FRET) as much as possible. Quantitative colocalization is perhaps the most extreme example of image analysis: it takes two images (typically containing millions of pixels) and produces only a few num-



Colocalization Measures

Pearson's correlation coefficient: $$r_P = \frac{\sum (I_1 - \bar{I}_1)(I_2 - \bar{I}_2)}{\sqrt{\sum (I_1 - \bar{I}_1)^2 \sum (I_2 - \bar{I}_2)^2}}$$

example	r_P	r	k_1	k_2	m_1	m_2
1	0.49	0.58	0.79	0.43	0.99	0.15
2	0.39	0.71	0.81	0.62	0.35	0.24
3	-0.08	0.37	0.24	0.57	0.23	0.38

overlap coefficient: $r = \sqrt{k_1 k_2}$ with

$$k_1 = \frac{\sum I_1 I_2}{\sum I_1^2} \quad \text{and} \quad k_2 = \frac{\sum I_1 I_2}{\sum I_2^2}$$

Manders' colocalization coefficients:

$$m_1 = \frac{\sum I_{1\,\mathrm{coloc}\,2}}{\sum I_1} \quad \text{and} \quad m_2 = \frac{\sum I_{2\,\mathrm{coloc}\,1}}{\sum I_2}$$

Figure 3.1 See description on page 21.

bers: the so-called colocalization measures (Figure 3.1). Pearson's correlation coefficient is often used for this purpose but may produce negative values, which is counterintuitive for a measure expressing the degree of overlap. A more intuitive measure, ranging from 0 (no colocalization) to 1 (full colocalization), is the so-called overlap coefficient , but it is appropriate only when the number of fluorescent targets is more or less equal in each channel. If this is not the case, multiple coefficients (two in the case of dual-color imaging) are required to quantify the degree of colocalization in a meaningful way.[46] These, however, tend to be rather sensitive to background offsets and noise,

Figure 3.1 Commonly used measures for quantitative colocalization analysis. The aim of all these measures is to express in numbers the degree of overlap between two fluorophores (captured in well separated channels), indicating the presence of the corresponding labeled molecules in the same or proximate physical locations (up to the optical resolution of the microscope). A visual impression of the co-occurrence of fluorophore intensities (I_1 and I_2) is given by the joint histogram (also referred to as the scatter plot or fluorogram). Some colocalization measures are computed over the entire images, while some are restricted to certain intensity ranges (indicated by the squares in the joint histograms). Among the first are Pearson's correlation coefficient (denoted r_P) and the so-called overlap coefficient (denoted r and computed from the subcoefficients k_1 and k_2). Both coefficients are insensitive to intensity scalings (due to photobleaching or a difference in signal amplification), while the former is also insensitive to intensity offsets (different background levels). The value of r_P may range from -1 to 1 and is therefore at odds with intuition. Its squared value is perhaps more valuable as it expresses the quality of a least-squares fitting of a line through the points in the scatter plot. The other measures range from 0 to 1. The value of r is meaningful only when the amount of fluorescence is approximately equal in both channels, that is when k_1 and k_2 have similar values. Manders' colocalization coefficients (denoted m_1 and m_2) are intuitively most clear but require careful separation of signal and background in both channels: the denominators are computed over the entire images, but the numerators sum only those intensities in one channel for which the corresponding intensity in the other channel is within a predefined range (the left and right and the top and bottom lines of the square region indicated in the joint histogram, for I_1 and I_2 respectively).

and require careful image restoration.[42] The most important step in colocalization analysis is the separation of signal and background, which is often done by intensity thresholding at visually determined levels.[62] The objectivity and reproducibility of this step may be improved considerably by applying statistical significance tests and automated threshold search algorithms.[19] Clearly, the resolution of colocalization is limited to the optical resolution of the microscope (in the order of 200 nm laterally and 600 nm axially), which is insufficient to determine whether two fluorescent molecules are really attached to the same target or reside within the same organelle. If colocalization or molecular interaction needs to be quantitatively studied at much higher resolutions (less than 10 nm), FRET imaging and analysis is more appropriate.[8]

3.2 Neuron Tracing and Quantification

Another biological image analysis problem, which occurs for example when studying the molecular mechanisms involved in neurite outgrowth and differentiation, is the length measurement of elongated image structures. For practical reasons, many neuronal morphology studies were and still are performed using 2D imaging. This often results in ambiguous images: at many places it is unclear whether neurites are branching or crossing. Tracing such structures and building neuritic trees for morphological analysis requires the input of human experts to resolve ambiguities. This resorting to human input is not unique to neuron tracing but is inevitable in many other complicated image analysis tasks and has led to the development of a variety of inter-

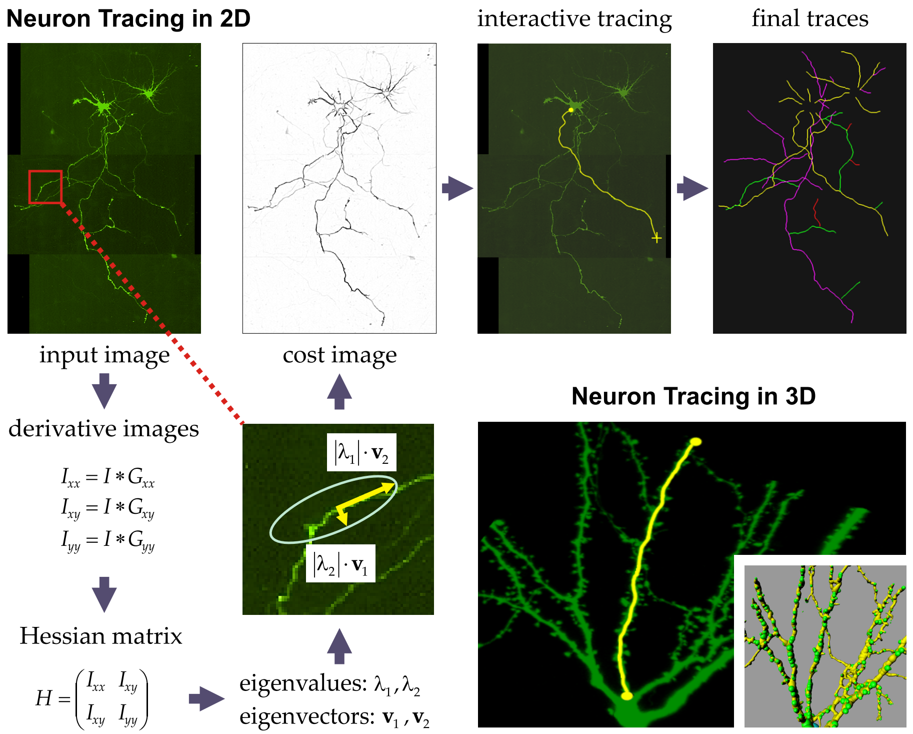

Neuron Tracing in 2D
interactive tracing
final traces
input image
cost image
derivative images
$I_{xx} = I * G_{xx}$
$I_{xy} = I * G_{xy}$
$I_{yy} = I * G_{yy}$
Hessian matrix
$H = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}$
$|\lambda_1| \cdot \mathbf{v}_2$
$|\lambda_2| \cdot \mathbf{v}_1$
eigenvalues: λ_1, λ_2
eigenvectors: $\mathbf{v}_1, \mathbf{v}_2$
Neuron Tracing in 3D

Figure 3.2 Tracing of neurite outgrowth using interactive segmentation methods. To reduce background intensity gradients (shading effects) or discontinuities (due to the stitching of scans with different background levels), the image features exploited here are the second-order derivatives, obtained by convolution with the second-order Gaussian derivative kernels (Figure 2.2) at a proper scale (to suppress noise). These constitute a so-called Hessian matrix at every pixel in the image. Its eigenvalues and eigenvectors are used to construct an ellipse (as indicated), whose size is representative of the local neurite contrast and whose orientation corresponds to the local neurite orientation. In turn, these properties are used to compute a cost image (with dark values indicating a lower cost and bright values a higher cost) and vector field (not shown), which together guide a search algorithm that finds the paths of minimum cumulative cost between a start point and all other points in the image. By using graphics routines, the path to the current cursor position (indicated by the cross) is shown at interactive speed while the user selects the optimal path based on visual judgment. Once tracing is finished, neurite lengths and statistics can be computed automatically. This is the underlying principle of the NeuronJ tracing tool, freely available as a plugin to the ImageJ program (discussed in the final chapter). The FilamentTracer tool, commercially available as part of the Imaris software package, uses similar principles for tracing in 3D images, based on volume visualization.

active segmentation methods. An example is live-wire segmentation, which was originally designed to perform computer supported delineation of object edges.[6,25] It is based on a search algorithm that finds a path from a single user-selected pixel to all other pixels in the image by minimizing the cumulative value of a predefined cost function, computed from local image features (such as gradient magnitude) along the path. The user can then interactively select the path that according to his own judgment best follows the structure of interest and fix the tracing up to some point, from where the process is iterated until the entire structure is traced. This technique has been adapted to enable tracing of neurite-like image structures in 2D and similar methods have been applied to neuron tracing in 3D (Figure 3.2).[50] More automated methods for 3D neuron tracing have also been published.[24,35,77] In the case of poor image quality, however, these may require manual postprocessing.

3.3 Particle Detection and Tracking

One of the major challenges of biomedical research in the postgenomic era is the unraveling of not just the spatial, but the *spatiotemporal* relationships of complex biomolecular systems.[89] Naturally this involves the acquisition of time-lapse image series and the tracking of objects over time. From an image analysis point of view, a distinction can be made between tracking of single molecules (or complexes) and tracking of entire cells (see next section). A number of tools are available for studying the dynamics of proteins based on fluorescent labeling and time-lapse imaging, such as fluorescence recovery after (and loss in) photobleaching (FRAP and FLIP respectively), but these yield only ensemble average measurements of properties. More detailed studies into the different modes of motion of subpopulations require single particle tracking,[65,75] which aims at motion analysis of individual proteins or microspheres. Computerized image analysis methods for this purpose have been developed since the early 1990s and are constantly being improved[5,21] to deal with increasingly sophisticated biological experimentation.

Generally speaking, particle tracking methods consist of two stages:[51] 1) the detection of individual particles per time frame, and 2) the linking of particles detected in successive frames (Figure 3.3). Regarding the former, it has been shown by theoretical as well as empirical studies[3,16,58,67,87] that the localization error can be at least one order of magnitude lower than the extension of the microscope PSF, and that the SNR is among the main factors limiting the localization accuracy. Currently, one of the best approaches to particle detection is by least-squares fitting of a Gaussian (mixture) model to the image data. In practice, the real difficulty in particle tracking is the data association problem: determining which particles as detected in one frame correspond to which particles in the next is not trivial, as the number of (real or detected) particles may not be constant over time (particles may enter or exit the field of view, they may assemble or disassemble, or limitations in the detection stage may cause varying degrees of under- or overdetection). Therefore, most current particle tracking tools should be used with care[14] and may still require manual checking and correction of the results.

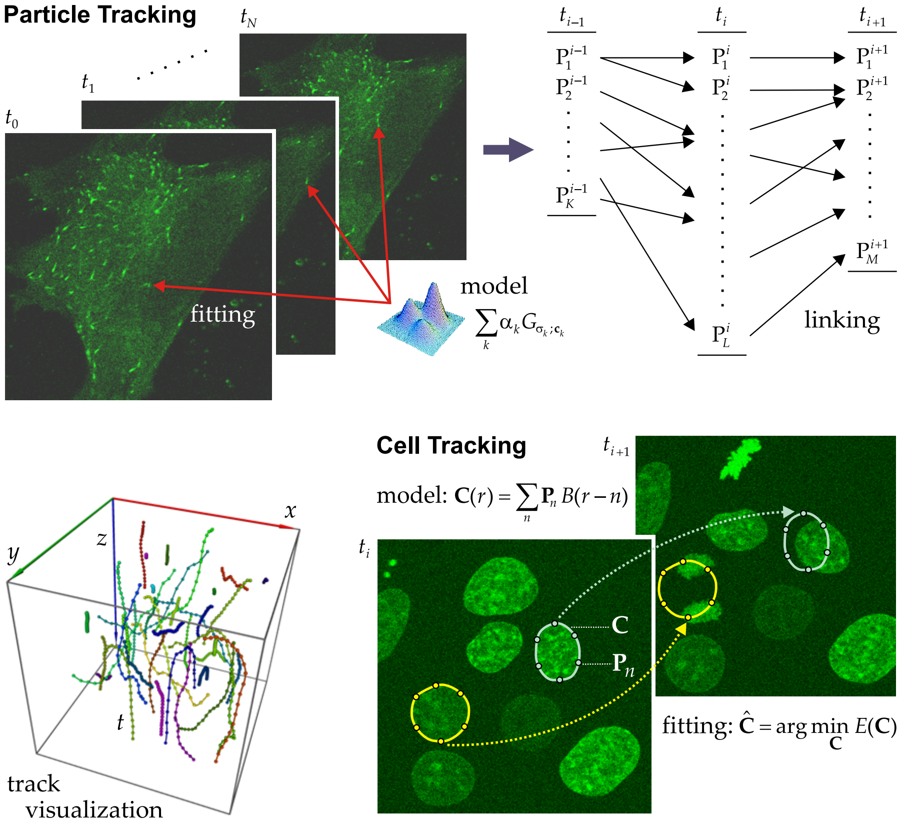


Figure 3.3 Challenges in particle and cell tracking. Regarding particle tracking, currently one of the best approaches to detection of fluorescent tags is by least-squares fitting of a model of the intensity distribution to the image data. Because the tags are subresolution particles, they appear as diffraction-limited spots in the images and therefore can be modeled well by a mixture of Gaussian functions, each with its own amplitude scaling factor, standard deviation, and center position. Usually the detection is done separately for each time step, resulting in a list of potential particle positions and corresponding features, to be linked between time steps. The linking is hampered by the fact that the number of detected particles may be different for each time step. In cell tracking, a contour model (surface model in the case of 3D time-lapse experiments) is often used for segmentation. Commonly used models consist of control points, which are interpolated using smooth basis functions (typically B-splines) to form continuous, closed curves. The model must be flexible enough to handle geometrical as well as topological shape changes (cell division). The fitting is done by (constrained) movement of the control points to minimize some predefined energy functional computed from image-dependent information (intensity distributions inside and outside the curve) as well as image-independent information (a priori knowledge about cell shape and dynamics). Finally, trajectories can be visualized by representing them as tubes (segments) and spheres (time points) and using surface rendering.

3.4 Cell Segmentation and Tracking

Motion estimation of cells is another frequently occurring problem in biological research. In particle tracking studies, for example, cell movement may muddle the motion analysis of intracellular components and needs to be corrected for. In some cases, this may be accomplished by applying (nonrigid) image registration methods.[23,29,70,82] Cell migrations and deformations are also interesting in their own right, however, because of their role in a number of biological processes, including immune response, wound healing, embryonic development, and cancer metastasis.[18] Understanding these processes is of major importance in developing drugs or therapies to combat various types of human disease. Typical 3D time-lapse data sets acquired for studies in this area consist of thousands of images and are almost impossible to analyze manually, both from a cost-efficiency perspective and because visual inspection lacks the sensitivity, accuracy, and reproducibility needed to detect subtle but potentially important phenomena. Therefore, computerized, quantitative cell tracking and motion analysis is a requisite.[22,92]

Contrary to single molecules or molecular complexes, which are subresolution objects appearing as PSF-shaped spots in the images, cells are relatively (with respect to pixel size) large objects, having a distinct shape. Detecting (or segmenting) entire cells and tracking position and shape changes requires quite different image processing methods. Due to noise and photobleaching effects, simple methods based on intensity thresholding are generally inadequate. To deal with these artifacts and with obscure boundaries in the case of touching cells, recent research has focused on the use of model-based segmentation methods,[41,48] which allow the incorporation of prior knowledge about object shape. Examples of such methods are active contours (also called snakes) and surfaces, which have been applied to a number of cell tracking problems.[20,22,69] They involve mathematical, prototypical shape descriptions having a limited number of degrees of freedom, which enable shape-constrained fitting to the image data based on data-dependent information (image properties, in particular intensity gradient information) and data-independent information (prior knowledge about the shape). Tracking is achieved by using the contour or surface obtained for one image as initialization for the next, and repeating the fitting procedure (Figure 3.3).

4 Visualization

Advances in imaging technology are rapidly turning higher-dimensional image data acquisition into the rule rather than the exception. Consequently, there is an increasing need for sophisticated visualization technology to enable efficient presentation and exploration of this data and associated image analysis results. Early systems supported browsing the data in a frame-by-frame fashion,[88] which provided only limited insight into the interrelations of objects in the images. Since visualization means generating representations of higher-dimensional data, this necessarily implies reducing dimensionality and possibly even reducing information in some sensible way. Exactly how to do this optimally depends on the application, the dimensionality of the data, and the physical nature of its respective dimensions (Figure 1.1). In any case, visualization methods usually consist of highly sophisticated information processing steps that may have a strong influence on the final result, making them very susceptible to misuse. Here we briefly explain the two main modes of visualization and point at critical steps in the process. More information can be found in textbooks on visualization and computer graphics.[27,78]

4.1 Volume Rendering

Visualization methods that produce a viewable image of higher-dimensional image data without requiring an explicit geometrical representation of that data are called volume rendering methods. A commonly used, flexible and easy to understand volume rendering method is ray casting, or ray tracing. With this method, the value of each pixel in the view image is determined by "casting a ray" into the image data and evaluating the data encountered along that ray using a predefined ray function (Figure 4.1). The direction of the rays is determined by the viewing angles and the mode of projection, which can be orthographic (the rays run parallel to each other) or perspective (the rays have a common focal point). Analogous to the real-world situation, these are called camera properties. The rays pass through the data with a certain step size, which should be smaller than the pixel size to avoid skipping impor-

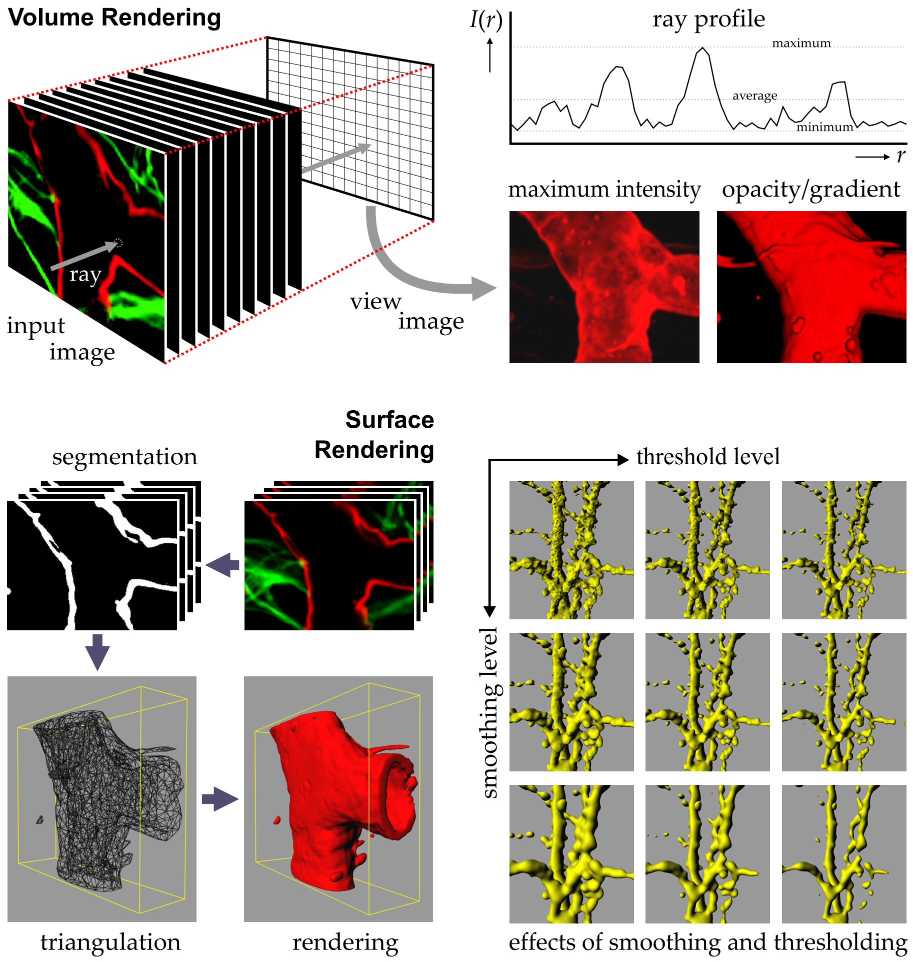


Figure 4.1 Visualization of volumetric image data using volume rendering and surface rendering methods. Volume rendering methods do not require an explicit geometrical representation of the objects of interest present in the data. A commonly used volume rendering method is ray casting: for each pixel in the view image, a ray is cast into the data, and the intensity profile along the ray is fed to a ray function, which determines the output value, such as the maximum, average, or minimum intensity, or accumulated "opacity" (derived from intensity or gradient magnitude information). By contrast, surface rendering methods require a segmentation of the objects (usually obtained by thresholding), from which a surface representation (triangulation) is derived, allowing very fast rendering by graphics hardware. To reduce the effects of noise, Gaussian smoothing is often applied as a preprocessing step prior to segmentation. As shown, both operations have a substantial influence on the final result: by slightly changing the degree of smoothing or the threshold level, objects may appear (dis)connected while in fact they are not. Therefore it is recommended to establish optimal parameter values for both steps while inspecting the effects on the original image data rather than looking directly at the renderings.

tant details. Since, as a consequence, the ray sample points will generally not coincide with grid positions, this requires data interpolation. The ray function determines what information is interpolated and evaluated and how the sample values are composited into a single output value. For example, if the function considers image intensity only and stores the maximum value found along the ray, we obtain a maximum intensity projection (MIP). Alternatively, it may sum all values and divide by the number to yield an average intensity projection. These methods are useful to obtain first impressions, even in the case of very noisy data, but the visualizations are often ambiguous due to overprojections. More complex schemes may consider gradient magnitude, color, or distance information. They may also include lighting effects to produce nicely shaded results. Each method yields a different view of the image data and may give rise to a slightly different interpretation. It is therefore often beneficial to use multiple methods.

4.2 Surface Rendering

In contrast with volume rendering methods, which in principle take into account all data along rays and therefore enable the visualization of object interiors, surface rendering methods visualize only object surfaces. Generally, this requires a mathematical description of the surfaces in terms of primitive geometrical entities: points, lines, triangles, polygons, or polynomial curves and surface patches, in particular splines. Such descriptions are derived from a segmentation of the image data into meaningful parts (objects versus background). This constitutes the most critical aspect of surface rendering: the value of the visualization depends almost entirely on the correctness of the segmentation (Figure 4.1). Once a correct segmentation is available, however, a representation of the object surfaces in terms of primitives, in particular a surface triangulation, is easily obtained by applying the so-called marching cubes algorithm.[44] Having arrived at this point, the visualization task has reduced to a pure computer graphics problem: generating an image from numbers representing primitive geometrical shapes. This could be done, again, by ray tracing: for each pixel in the view image, a ray is defined and its intersections with the surfaces are computed, at which points the effect of the light source(s) on the surfaces (based on their orientation, opacity, color, texture) are determined to yield an output pixel value. This is called image-order rendering (from pixels to surfaces). Most modern computer graphics hardwares, however, use object-order rendering (from surfaces, or primitives, to pixels). Note that using such methods, we can visualize not just segmented image data, but any information that can be converted somehow to graphics primitives. Examples of this are tracing and tracking results, which can be represented by tubes and spheres (Figures 3.2 and 3.3).

5

Software Tools

It will be clear from the previous chapters that despite the host of image processing, analysis, and visualization methods developed over the past decades, there exists no such thing as a universal method capable of solving all problems. Although it is certainly possible to categorize problems, each biological study is unique, in the sense that it is based on specific premises and hypotheses to be tested, giving rise to unique image data to be analyzed, and requiring dedicated image analysis methods in order to take full advantage of this data. As a consequence, there is a great variety of software tools. Roughly, they can be divided into four categories, spanning the entire spectrum from least to most dedicated. On the one end are tools that are mainly meant for image acquisition but that also provide basic image processing, measurement, visualization, and documentation facilities. Examples include some tools provided by microscope manufacturers, such as LSM Image Browser,[13] QWin,[43] and analySIS.[59] Next are tools that, in addition to offering basic facilities, were designed to also address a range of more complicated biological image analysis problems. Often, these tools consist of a core platform with the possibility to add modules developed for dedicated applications, such as deconvolution, colocalization, filament tracing, image registration, or particle tracking. Examples of these include Imaris,[9] AxioVision,[13] Image-Pro Plus,[49] MetaMorph,[55] and ImageJ.[68] On the other end of the spectrum are tools that are much more dedicated to specific tasks, such as Huygens[79] or AutoDeblur[4] for deconvolution, Amira[53] for visualization, Volocity[37] for tracking and motion analysis, and Neurolucida[54] for neuron tracing, et cetera.

As a fourth category we mention software packages offering researchers much greater flexibility in developing their own, dedicated image analysis algorithms. An example of this is Matlab,[47] which offers an interactive developing environment and a high-level programming language for which extensive image processing toolboxes are available, such as DIPimage.[66] It is used by engineers and scientists in many fields for rapid prototyping and validation of new algorithms but has not (yet) gained wide acceptance in biology. A software tool that is rapidly gaining popularity is ImageJ, already mentioned. It is a public-domain tool and developing environment based on the

Java programming language[83] which can be used without the need for a license, runs on any computer platform (PC/Windows, Macintosh, Linux, and other Unix variants), and its source code is openly available. The core distribution of the program supports most of the common image file formats and offers a host of facilities for manipulation and analysis of image data (up to 5D), including all basic image processing methods described in this booklet. Probably the strongest feature of the program is its extensibility: existing operations can be combined into more complex algorithms by means of macros, and new functionality can easily be added by writing plugins.[1] Hundreds of plugins are already available, considerably increasing its image file support and image processing and analysis capabilities, ranging from very basic but highly useful pixel manipulations to much more involved algorithms for image segmentation, registration,[86] transformation, mosaicking, visualization,[2,71] deconvolution, depth-of-field extension,[28] neuron tracing,[50] FRET analysis,[26] manual[56] and automated[74,76] particle tracking, colocalization, texture analysis, cell counting, granulometry, and more.

Finally we wish to make a few remarks regarding the use and development of software tools for biological image analysis. In contrast with diagnostic patient studies in clinical medical imaging practice, biological investigation is rather experimental by nature, allowing researchers to design their own experiments, including the imaging modalities to be used and how to process and analyze the resulting data. While freedom is a great virtue in science, it may also give rise to chaos. All too often, scientific publications report the use of image analysis tools without specifying which algorithms were involved and how parameters were set, making it very difficult for others to reproduce or compare results. What is worse, many software tools available on the market or in the public domain have not been thoroughly scientifically validated, at least not in the open literature, making it impossible for reviewers to verify the validity of using them under certain conditions. It requires no explanation that this situation needs to improve. Another consequence of freedom, on the engineering side of biological imaging, is the (near) total lack of standardization in image data management and information exchange. Microscope manufacturers, software companies, and sometimes even research laboratories have their own image file formats, which generally are rather rigid. As the development of new imaging technologies and analytic tools accelerates, there is an increasing need for an adaptable data model for multidimensional images, experimental metadata, and analytical results, to increase the compatibility of software tools and facilitate the sharing and exchange of information between laboratories. First steps in this direction have already been taken by the microscopy community by developing and implementing the open microscopy environment (OME), whose data model and file format, based on the extended markup language (XML), is gaining acceptance.[32,84]

References

The following overview is not meant to be exhaustive but contains references to the (mostly recent) literature where the interested reader can find (further references to) more in-depth information.

1. M. D. Abràmoff, P. J. Magalhães, and S. J. Ram. Image processing with ImageJ. *Biophotonics International* 11(7):36–42, 2004.
2. M. D. Abràmoff and M. A. Viergever. Computation and visualization of three-dimensional soft tissue motion in the orbit. *IEEE Transactions on Medical Imaging* 21(4):296–304, 2002.
3. F. Aguet, D. Van De Ville, and M. Unser. A maximum-likelihood formalism for sub-resolution axial localization of fluorescent nanoparticles. *Optics Express* 13(26):10503–10522, 2005.
4. AutoQuant Imaging, Inc., Troy, USA.
5. C. P. Bacher, M. Reichenzeller, C. Athale, H. Herrmann, and R. Eils. 4-D single particle tracking of synthetic and proteinaceous microspheres reveals preferential movement of nuclear particles along chromatin-poor tracks. *BMC Cell Biology* 5(45):1–14, 2004.
6. W. A. Barrett and E. N. Mortensen. Interactive live-wire boundary extraction. *Medical Image Analysis* 1(4):331–341, 1997.
7. G. A. Baxes. *Digital Image Processing: Principles and Applications*. John Wiley & Sons, New York, 1994.
8. C. Berney and G. Danuser. FRET or no FRET: A quantitative comparison. *Biophysical Journal* 84(6):3992–4010, 2003.
9. Bitplane AG, Zürich, Switzerland.
10. M. Born and E. Wolf. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. Pergamon, Oxford, 6th edition, 1980.
11. R. N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill, New York, 3rd edition, 2000.
12. J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8(6):679–698, 1986.
13. Carl Zeiss MicroImaging GmbH, Göttingen, Germany.
14. B. C. Carter, G. T. Shubeita, and S. P. Gross. Tracking single particles: A user-friendly quantitative evaluation. *Physical Biology* 2(1):60–72, 2005.
15. K. R. Castleman. *Digital Image Processing*. Prentice Hall, Englewood Cliffs, 1996.
16. M. K. Cheezum, W. F. Walker, and W. H. Guilford. Quantitative comparison of algorithms for tracking single fluorescent particles. *Biophysical Journal* 81(4):2378–2388, 2001.

17. H. Chen, J. R. Swedlow, M. Grote, J. W. Sedat, and D. A. Agard. The collection, processing, and display of digital three-dimensional images of biological specimens. In J. B. Pawley (editor), *Handbook of Biological Confocal Microscopy*, chapter 13, pages 197–210. Plenum Press, London, 2nd edition, 1995.

18. M. Chicurel. Cell migration research is on the move. *Science* 295(5555):606–609, 2002.

19. S. V. Costes, D. Daelemans, E. H. Cho, Z. Dobbin, G. Pavlakis, and S. Lockett. Automatic and quantitative measurement of protein-protein colocalization in live cells. *Biophysical Journal* 86(6):3993–4003, 2004.

20. O. Debeir, I. Camby, R. Kiss, P. Van Ham, and C. Decaestecker. A model-based approach for automated in vitro cell tracking and chemotaxis analyses. *Cytometry Part A* 60(1):29–40, 2004.

21. J. F. Dorn, K. Jaqaman, D. R. Rines, G. S. Jelson, P. K. Sorger, and G. Danuser. Yeast kinetochore microtubule dynamics analyzed by high-resolution three-dimensional microscopy. *Biophysical Journal* 89(4):2835–2854, 2005.

22. A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillen-Aghion, J.-C. Olivo-Marin, and C. Zimmer. Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces. *IEEE Transactions on Image Processing* 14(9):1396–1410, 2005.

23. R. Eils and C. Athale. Computational imaging in cell biology. *Journal of Cell Biology* 161(3):477–481, 2003.

24. J. F. Evers, S. Schmitt, M. Sibila, and C. Duch. Progress in functional neuroanatomy: Precise automatic geometric reconstruction of neuronal morphology from confocal image stacks. *Journal of Neurophysiology* 93(4):2331–2342, 2005.

25. A. X. Falcão, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. de A. Lotufo. User-steered image segmentation paradigms: Live wire and live lane. *Graphical Models and Image Processing* 60(4):233–260, 1998.

26. J. N. Feige, D. Sage, W. Wahli, B. Desvergne, and L. Gelman. PixFRET, an ImageJ plug-in for FRET calculation that can accommodate variations in spectral bleed-throughs. *Microscopy Research and Technique* 68(1):51–58, 2005.

27. J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, 2nd edition, 1997.

28. B. Forster, D. Van De Ville, J. Berent, D. Sage, and M. Unser. Complex wavelets for extended depth-of-field: A new method for the fusion of multichannel microscopy images. *Microscopy Research and Technique* 65(1-2):33–42, 2004.

29. D. Gerlich, J. Mattes, and R. Eils. Quantitative motion analysis and visualization of cellular structures. *Methods* 29(1):3–13, 2003.

30. C. A. Glasbey. An analysis of histogram-based thresholding algorithms. *CVGIP: Graphical Models and Image Processing* 55(6):532–537, 1993.

31. C. A. Glasbey and G. W. Horgan. *Image Analysis for the Biological Sciences*. John Wiley & Sons, New York, 1995.

32. I. G. Goldberg, C. Allan, J.-M. Burel, D. Creager, A. Falconi, H. Hochheiser, J. Johnston, J. Mellen, P. K. Sorger, and J. R. Swedlow. The Open Microscopy Environment (OME) Data Model and XML file: Open tools for informatics and quantitative analysis in biological imaging. *Genome Biology* 6(5):R47, 2005.

33. R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, Upper Saddle River, 2nd edition, 2002.

34. M. Gu. *Advanced Optical Imaging Theory*. Springer-Verlag, Berlin, 2000.

35. W. He, T. A. Hamilton, A. R. Cohen, T. J. Holmes, C. Pace, D. H. Szarowski, J. N. Turner, and B. Roysam. Automated three-dimensional tracing of neurons in confocal and brightfield images. *Microscopy and Microanalysis* 9(4):296–310, 2003.

36. D. Houle, J. Mezey, P. Galpern, and A. Carter. Automated measurement of Drosophila wings. *BMC Evolutionary Biology* 3(25):1–13, 2003.

37. Improvision, Inc., Lexington, MA, USA.

38. B. Jähne. *Practical Handbook on Image Processing for Scientific Applications*. CRC Press, Boca Raton, 2nd edition, 2004.

39. A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, 1989.

40. P. A. Jansson (editor). *Deconvolution of Images and Spectra*. Academic Press, San Diego, 1997.

41. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision* 1(4):321–331, 1988.

42. L. Landmann and P. Marbet. Colocalization analysis yields superior results after image restoration. *Microscopy Research and Technique* 64(2):103–112, 2004.

43. Leica Microsystems GmbH, Wetzlar, Germany.

44. W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics* 21(3):163–169, 1987.

45. J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis* 2(1):1–36, 1998.

46. E. M. M. Manders, F. J. Verbeek, and J. A. Aten. Measurement of colocalization of objects in dual-colour confocal images. *Journal of Microscopy* 169(3):375–382, 1993.

47. The MathWorks, Inc., Natick, MA, USA.

48. T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis* 1(2):91–108, 1996.

49. Media Cybernetics, Inc., Silver Spring, MD, USA.

50. E. Meijering, M. Jacob, J.-C. F. Sarria, P. Steiner, H. Hirling, and M. Unser. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry Part A* 58(2):167–176, 2004.

51. E. Meijering, I. Smal, and G. Danuser. Tracking in molecular bioimaging. *IEEE Signal Processing Magazine* 23(3):46–53, 2006.

52. E. H. W. Meijering, W. J. Niessen, and M. A. Viergever. Quantitative evaluation of convolution-based methods for medical image interpolation. *Medical Image Analysis* 5(2):111–126, 2001.

53. Mercury Computer Systems, Chelmsford, MA, USA.

54. MicroBrightField, Inc., Williston, VT, USA.

55. Molecular Devices Corporation, Downingtown, PA, USA.

56. MTrackJ, Biomedical Imaging Group Rotterdam, Erasmus MC, The Netherlands.

57. R. F. Murphy, E. Meijering, and G. Danuser. Special issue on molecular and cellular bioimaging. *IEEE Transactions on Image Processing* 14(9):1233–1236, 2005.

58. R. J. Ober, S. Ram, and E. S. Ward. Localization accuracy in single-molecule microscopy. *Biophysical Journal* 86(2):1185–1200, 2004.

59. Olympus Soft Imaging Solutions GmbH, Berlin, Germany.

60. N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9(1):62–66, 1979.

61. J. B. Pawley (editor). *Handbook of Biological Confocal Microscopy*. Plenum Press, London, 2nd edition, 1995.

62. P. G. Peñarrubia, X. F. Ruiz, and J. Gálvez. Quantitative analysis of the factors that affect the determination of colocalization coefficients in dual-color confocal images. *IEEE Transactions on Image Processing* 14(8):1151–1158, 2005.

63. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(7):629–639, 1990.

64. J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: A survey. *IEEE Transactions on Medical Imaging* 22(8):986–1004, 2003.

65. H. Qian, M. P. Sheetz, and E. L. Elson. Single particle tracking: Analysis of diffusion and flow in two-dimensional systems. *Biophysical Journal* 60(4):910–921, 1991.

66. Quantitative Imaging Group, Delft University of Technology, The Netherlands.

67. S. Ram, E. S. Ward, and R. J. Ober. Beyond Rayleigh's criterion: A resolution measure with application to single-molecule microscopy. *Proceedings of the National Academy of Science USA* 103(12):4457–4462, 2006.

68. W. S. Rasband, National Institutes of Health, Bethesda, MD, USA.

69. N. Ray, S. T. Acton, and K. Ley. Tracking leukocytes in vivo with shape and size contrained active contours. *IEEE Transactions on Medical Imaging* 21(10):1222–1235, 2002.

70. B. Rieger, C. Molenaar, R. W. Dirks, and L. J. van Vliet. Alignment of the cell nucleus from labeled proteins only for 4D in vivo imaging. *Microscopy Research and Technique* 64(2):142–150, 2004.

71. C. Rueden, K. W. Eliceiri, and J. G. White. VisBio: A computational tool for visualization of multidimensional biological image data. *Traffic* 5(6):411–417, 2004.

72. J. C. Russ. *The Image Processing Handbook*. CRC Press, Boca Raton, 4th edition, 2002.

73. S. Sabri, F. Richelme, A. Pierres, A.-M. Benoliel, and P. Bongrand. Interest of image processing in cell biology and immunology. *Journal of Immunological Methods* 208(1):1–27, 1997.

74. D. Sage, F. R. Neumann, F. Hediger, S. M. Gasser, and M. Unser. Automatic tracking of individual fluorescence particles: Application to the study of chromosome dynamics. *IEEE Transactions on Image Processing* 14(9):1372–1383, 2005.

75. M. J. Saxton and K. Jacobson. Single-particle tracking: Applications to membrane dynamics. *Annual Review of Biophysics and Biomolecular Structure* 26:373–399, 1997.

76. I. F. Sbalzarini and P. Koumoutsakos. Feature point tracking and trajectory analysis for video imaging in cell biology. *Journal of Structural Biology* 151(2):182–195, 2005.

77. S. Schmitt, J. F. Evers, C. Duch, M. Scholz, and K. Obermayer. New methods for the computer-assisted 3-D reconstruction of neurons from confocal image stacks. *NeuroImage* 23(4):1283–1298, 2004.

78. W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Kitware, New York, 3rd edition, 2002.

79. Scientific Volume Imaging BV, Hilversum, The Netherlands.

80. J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.

81. M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS Publishing, Pacific Grove, 2nd edition, 1999.

82. C. Ó. S. Sorzano, P. Thévenaz, and M. Unser. Elastic registration of biological images using vector-spline regularization. *IEEE Transactions on Biomedical Engineering* 52(4):652–663, 2005.

83. Sun Microsystems, Inc., Santa Clara, CA, USA.

84. J. R. Swedlow, I. Goldberg, E. Brauner, and P. K. Sorger. Informatics and quantitative analysis in biological imaging. *Science* 300(5616):100–102, 2003.

85. P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE Transactions on Medical Imaging* 19(7):739–758, 2000.

86. P. Thévenaz, U. E. Ruttimann, and M. Unser. A pyramid approach to subpixel registration based on intensity. *IEEE Transactions on Image Processing* 7(1):27–41, 1998.

87. D. Thomann, D. R. Rines, P. K. Sorger, and G. Danuser. Automatic fluorescent tag detection in 3D with super-resolution: Application to the analysis of chromosome movement. *Journal of Microscopy* 208(1):49–64, 2002.

88. C. Thomas, P. DeVries, J. Hardin, and J. White. Four-dimensional imaging: Computer visualization of 3D movements in living specimens. *Science* 273(5275):603–607, 1996.

89. R. Y. Tsien. Imagining imaging's future. *Nature Reviews Molecular Cell Biology* 4:S16–S21, 2003.

90. H. T. M. van der Voort and K. C. Strasters. Restoration of confocal images for quantitative image analysis. *Journal of Microscopy* 178(2):165–181, 1995.

91. S. L. Wearne, A. Rodriguez, D. B. Ehlenberger, A. B. Rocher, S. C. Henderson, and P. R. Hof. New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales. *Neuroscience* 136(3):661–680, 2005.

92. C. Zimmer, B. Zhang, A. Dufour, A. Thébaud, S. Berlemont, V. Meas-Yedid, and J.-C. Olivo-Marin. On the digital trail of mobile cells. *IEEE Signal Processing Magazine* 23(3):54–62, 2006.

Index

The page numbers refer to places in the booklet where the reader can find either a brief description of the corresponding index term or a reference to the relevant literature for that term.

1D, 7
2D, 6, 7
3D, 6, 7
4D, 7
5D, 7

Active contour, 25
Averaging, 12, 13, 18

Background subtraction, 17, 18
Blurring, *see* Smoothing

Canny edge detection, 12, 13
Cell tracking, 24, 25
Closing, 14, 15
Colocalization, 19*ff*
Computer graphics, 6, 8
Computer vision, 6, 8
Contrast stretching, 11, 12
Convolution, 12, 13
Coordinate, 6, 7

Deblurring, *see* Deconvolution
Deconvolution, 17, 18
Denoising, *see* Noise reduction
Derivative, 12, 13, 22
Dilation, 14, 15
Dimension, 6, 7
  spatial, 7
  spectral, 7
  temporal, 7

Edge detection
  Canny, 12, 13
  morphological, 14, 15

Erosion, 14, 15

Filter, 12
  averaging, 12, 13
  derivative, 12, 13, 22
  Gaussian, 13, 17
  linear, 12
  maximum, 12
  median, 12, 17, 18
  minimum, 12
  morphological, 12, 14
  nonlinear, 12, 17
  sharpening, 12, 13
  smoothing, 12, 13
Fluorogram, 21

Gaussian
  derivative, 22
  filtering, 13, 17
  mixture model, 23, 24
  smoothing, 13, 27
Granulometry, 14, 15

Histogram, 11
  cumulative, 11
  equalization, 11
  multimodal, 11, 12
  thresholding, 12
  unimodal, 11

Image, 5
  analysis, 6, 8, 19*ff*
    objective, 6
    qualitative, 6
    quantitative, 6
    subjective, 6

binary, 12
brightness, 10
classes, 6
contrast, 10
derivative, 13, 22
digital, 6
element, 6
filtering, 12
formation, 8
interpolation, *see* Interpolation
matrix, 6, 7
multispectral, 7, 10
processing, 6, 8, 10*ff*
reformatting, 15
registration, 15, 25
resampling, 15, 16
restoration, 15*ff*
segmentation, *see* Segmentation
time-lapse, 7
understanding, 6, 8
Interpolation, 15, 16
kernel, 16
linear, 15
nearest-neighbor, 15
spline, 15

Kernel, 12, 13, *see also* Filter

Manders' coefficients, 21
Maximum intensity projection, 27, 28
Median filtering, 12, 17, 18
Morphology
binary, 12, 14
mathematical, 12
neuronal, 21

Neighborhood operation, 12
Neuron tracing, 21*ff*
Noise reduction, 17, 18
Nonlinear diffusion filtering, 17, 18

Opening, 14, 15
Overlap coefficient, 20, 21

Particle tracking, 23, 24
Pearson's correlation coefficient, 20, 21
Picture, 6
Pixel, 6
Point operation, 10
Projection
accumulated opacity, 27
average intensity, 27, 28
maximum intensity, 27, 28
minimum intensity, 27
orthographic, 26
perspective, 26

Ray tracing, 26, 27
Reformatting, 15
Registration, 15, 25
Rendering
image-order, 28
object-order, 28
surface, 27, 28
volume, 26, 27
Resampling, 15, 16

Sample, 6
Segmentation, 10
interactive, 22
live-wire, 22
model-based, 25
Sharpening, 12, 13
Skeletonization, 14, 15
Smoothing, 12, 27
Snake, *see* Active contour
Software tools, 29*ff*
Structuring element, 12, 14
Surface rendering, 27, 28

Thresholding, 10–12, 27
Tracking, 23*ff*
Transformation
affine, 15, 16
coordinate, 15, 16
curved, 15, 16
Fourier, 12
geometrical, 15, 16
intensity, 10*ff*
rigid, 15, 16
spatial, 15

Visualization, 6, 8, 26*ff*
Volume rendering, 26, 27
Voxel, 6