Microscopic Image Analysis for Life Science Applications

September 8, 2007

Chapter Number	3
Chapter Title	Overview of Image Analysis Tools and Tasks for Microscopy
Author 1	Jelena Kovačević
Author 1 - Address	Departments of BME and ECE, Center for Bioimage Informatics
Author 1 - Address	Carnegie Mellon University, Pittsburgh, PA 15213
Author 1 - Email	jelenak@cmu.edu
Author 2	Gustavo K. Rohde
Author 2 - Address	Department of BME and Center for Bioimage Informatics
Author 2 - Address	Carnegie Mellon University, Pittsburgh, PA 15213
Author 2 - Email	gustavor@cmu.edu
Corresponding Author	Author 1

Chapter 3

Overview of Image Analysis Tools and Tasks for Microscopy

The advent of fluorescent proteins, together with the recent development of advanced high-resolution microscopes, has enabled scientists to probe the intricate structure and function of cells and subcellular structures with unprecedented accuracy and specificity. Imaging experiments have become the main source of data for biologists to test and validate hypotheses related to basic cellular and molecular phenomena. Computational methods for automated interpretation and information extraction from biological images have augmented the impact of imaging experiments and are quickly becoming a valuable extension of the microscope. Such studies have long been a major topic of biomedical research (see, for example, [1]), and the recent advances in microscopic image acquisition systems as well as sophisticated image processing algorithms indicate that this is a trend likely to continue [2, 3].

We present a brief overview of computer-aided image analysis tools and tasks for microscopy. While the tasks and applications we discuss are those currently needed in the field, the algorithms used to perform them are not often state-of-the-art. Thus, our aim is to introduce the reader to the modern tools of signal processing and data mining for microscopy, highlighting opportunities for innovative research in the area. Some recent accomplishments can be found in the special issue of the Signal Processing Magazine on molecular and cellular bioimaging [4].

The overview of the system we discuss is given in Fig. 3.1. We begin by reviewing a system's level approach that summarizes the physical aspects of acquisition described in Chapter 1 (the first two blocks in the figure, PSF and A/D conversion). Important features and artifacts of imaging acquisition systems such as sampling, blurring, and noise are briefly discussed in connection to traditional and modern techniques for automated artifact removal and information extraction (the blocks of denoising, deconvolution and restoration in the figure). The chapter is structured around mathematical tools currently used as well as some that are not, but have the potential for high impact in automated biological image analysis. We conclude the chapter by covering various image processing and analysis tasks needed in microscopy in the order they appear in a real system, starting with registration and mosaicing, followed by segmentation, tracing and tracking, and finally data modeling, analysis and simulation (the last three blocks in the figure). The main aim of this chapter is to provide a concise overview of the state-of-the-art tools and algorithms for computer-aided extraction of quantitative information from images, in the hope that they become state-of-the-art tools in microscopy as well.

3.1 Image Analysis Framework

We first set up the framework in which we will examine digital images arising from typical microscopes (widefield, confocal, and others). As terms themselves can lead to confusion, we note the following: The term “digital image” typically refers to a continuous-domain image that has been both discretized (that is, sampled on a discrete grid), as well as digitized (that is, the intensity has been quantized for digital representation). Further, in the signal processing literature, “time” is often used to denote the image domain, although strictly speaking the domain of images can be space, time, channels, etc. Therefore, to avoid confusion, when speaking of the types of domains, we will refer to them as continuous domain and discrete domain (we will, however, keep the standard nomenclature, such as discrete-time Fourier transform, time-frequency, etc).

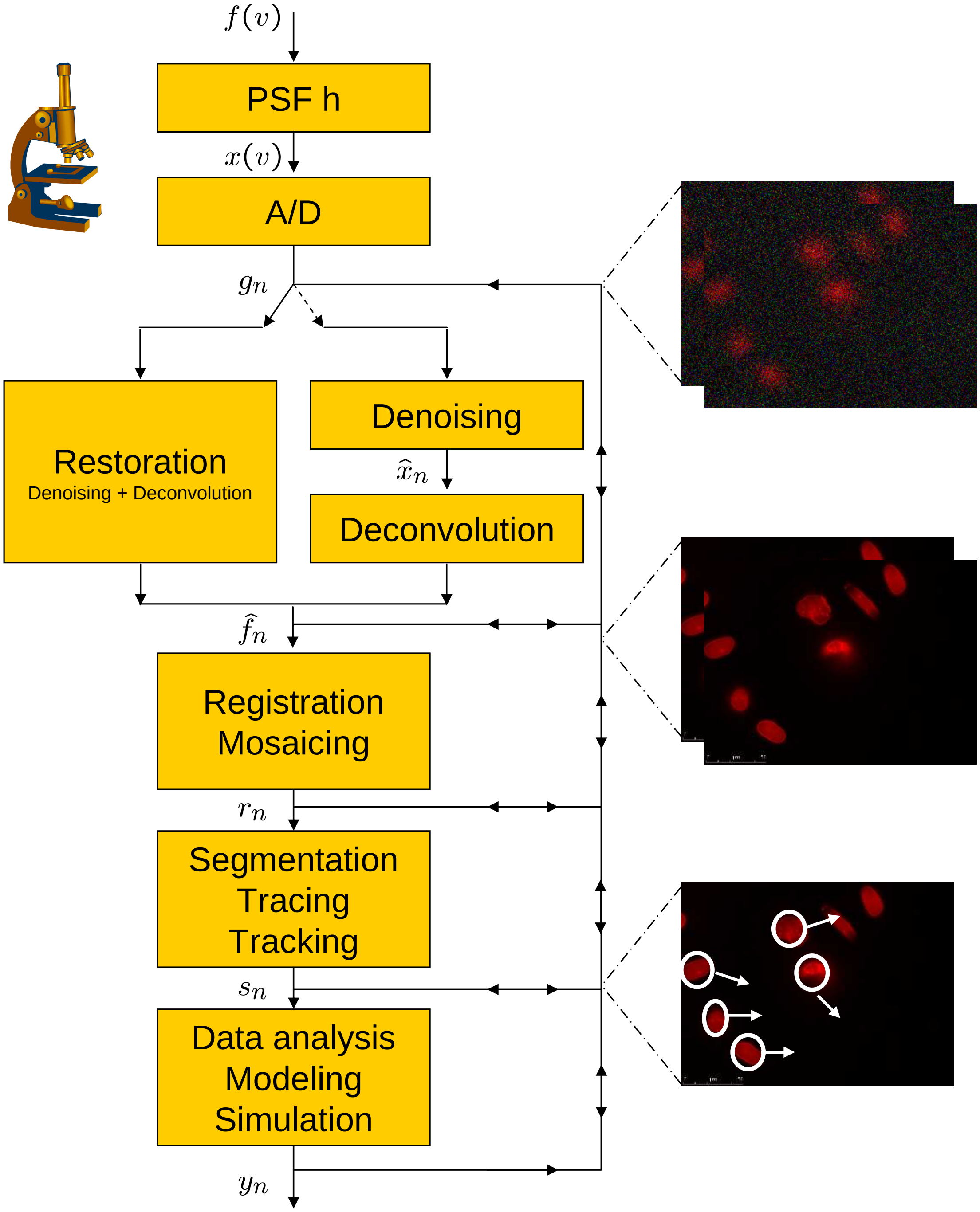

Figure 3.1: A conceptual pipeline. The specimen is imaged using any of today's microscopes, modeled by the input image $f(v)$ passing through the blocks of PSF (properties of the microscope, described by convolution with $h(v)$) and A/D conversion (analog-to-digital conversion, effects of sampling and digitization together with uncertainty introduced by various sources of noise), producing a digital image g_n. That digital image is then restored either via a denoising followed by deconvolution, or via joint denoising/deconvolution, producing a digital image $\hat{f}_n$. Various options are possible: the image could go through a registration/mosaicing processing producing r_n, segmentation/tracing/tracking producing s_n and finally a data analysis/modeling/simulations block with the output y_n. At the input/output of each block, one can join the pathway to either skip a block or send feedback to previous block(s) in the system. (Images courtesy of K. N. Dahl.)

Without discretization, the result of imaging an object (specimen) would be a continuous-domain image. We introduce images (both continuous-domain as well as discrete-domain) as vectors in Hilbert spaces and acquisition/imaging systems as transformations from one Hilbert space into another. Most of these will be linear and can transform one continuous-domain image into another (continuous-domain filtering), a continuous-domain image into a discrete-domain one (analog-to-digital conversion (A/D)), or, a discrete-domain image into a discrete-domain one (discrete-domain filtering). We will discuss all three of the above in what follows. A conceptual view of the whole system is given in Fig. 3.1. An excellent reference text on foundations of image science is [5].

3.1.1 Continuous-Domain Image Processing

What we usually call a *microscope image* is an already acquired, and thus discrete-domain data set, which we refer to from now on as a digital image. Since ultimately, what we really want to understand, analyze and interpret, are the biological characteristics of the underlying specimen, we must first understand the continuous-domain image at its root. While we should distinguish between the specimen and the image of the specimen, from now on, we will assume that the continuous-domain image of the specimen is the ground truth.

Framework. We will denote such an image as $f(x, y, z, t, c)$, where x, y are the two spatial dimensions in the focal plane of the objective, z is the third spatial dimension perpendicular to the focal plane, t is the time dimension in case of time-lapse imaging and c denotes multiple channels. (Not all the dimensions are present in every case.) To make matters even simpler, we introduce a vector $v = (x, y, z, t, c)$ which will allow us to add other dimensions to the vector if needed. The number of dimensions is denoted by d (five in this case). The image $f(v)$ appears as the input to the system in Fig. 3.1.

Very often, the value of $f(v)$ is scalar, for example, $f(v)$ could denote the intensity. It is also possible for $f(v)$ to be vector valued, as in the case of color. Thus, at any point in space and time, $f(v)$ could be given in one of the standard, three-variable color models such as RGB, YUV, HSV, and others. We will keep the notation the same regardless of whether $f(v)$ is scalar or vector valued; its meaning will be obvious from the context. For example, $f(x, y)$ is typically called a 2D slice, $f(x, y, z)$ a z-stack, $f(x, y, t)$ a 2D time series, while $f(x, y, z, t)$ is a 3D time series. In all these cases, the value can be either scalar or vector. For example,$(r(x, y), g(x, y), b(x, y)) = f(x, y)$ is an RGB 2D slice.

While in reality the domain of f is finite, that is, the domains of x, y, z, t, c are all of finite support, the domain of such images is often assumed to be infinite for ease of explanation. For example, while we might be imaging a specimen of finite size $100\mu m \times 100\mu m$ in the focal plane, we might use the tools which assume that the domain is infinite. This is also done to avoid more cumbersome tools operating on a finite domain (such as the circular convolution), although these tools have become standard as well.

Deterministic Vs Stochastic Description. For now, we limit ourselves to the deterministic treatment of images and transformations between them. The issues of randomness and uncertainty, such as those found when noise is present, will be discussed later.

Linear-Shift Invariant Systems and Point Spread Functions. Given an image $f(v)$, we may model the effect of the acquisition system on it as the operator $H[\cdot]$, resulting in $x(v) = H[f(v)]$ (see Fig. 3.1), where H is in general nonlinear. In image processing, H is termed a *filter*, and in practice, we most often assume that H is *linear*. In other words, an imaging, or, acquisition system, is a linear transformation from one Hilbert space into another.

Another important property of filters is shift invariance. A system/filter is called *shift invariant* if, given the input/output pair $(f(v), x(v))$, the output of the system with the shifted input $f(v - v_0)$ is a shifted output $x(v - v_0)$. Combining the properties of linearity and shift invariance gives rise to *linear shift-invariant (LSI)* systems; analysis is the simplest when the system is LSI. For an LSI system, the operation of the filter on the image is described by *convolution*, denoted by $*$; given that each image $f(v)$ can be written as

$$f(v) \;=\; \int f(r)\delta(v-r)dr, \tag{3.1}$$

where $\delta(v)$ is the Dirac pulse, the filtered image is given by

$$x(v) \;=\; H\left[\int f(r)\delta(v-r)dr\right] \;=\; \int f(r)H\left[\delta(v-r)dr\right] \;=\; \int f(r)h(v-r)dr \;=\; (h*f)(v). \tag{3.2}$$

In the above, we have defined the *impulse response* or *point spread function (PSF)* of an LSI system $h(v)$, as the output of the system to the input which is a Dirac pulse $\delta(v)$. Note that the PSF is typically assumed to be positive, while this is not necessarily true for the impulse response.

The Fourier View. Because of the cumbersome form of the convolution, when analyzing images, one often resorts to using the *Fourier transform (FT)*, as the convolution becomes multiplication in the Fourier domain. For certain classes of images, the FT finds the harmonic content of the image, or, in other words, it tries to represent the image as a sum of sinusoids. Fig. 3.2 shows a few examples of simple and more complex harmonic signals.

The FT of $f(v)$ is given by (we follow the convention of using lower-case letters for the image data in the original domain, and upper-case ones for the same in the Fourier domain):

$$F(\omega) \;=\; \int_{\mathbb{R}^d} f(v) e^{-j\omega^T v} dv, \tag{3.3}$$

where ω, v are vectors in general. For a 2D signal, (3.3) reads

$$F(\omega_1, \omega_2) \;=\; \int_{\mathbb{R}^2} f(v_1, v_2) e^{-j(\omega_1 v_1 + \omega_2 v_2)} dv_1 dv_2.$$

Using (3.3) on the convolution expression (3.2), we obtain that the FT of $x(v) = (h * f)(v)$ is $X(\omega) = H(\omega)F(\omega)$. The FT is valid assuming our signals belong to the space of square-integrable functions, $\mathcal{L}^2(\mathbb{R})$. If a signal is periodic, that is, if it lives on a circle, then the appropriate FT is called *Fourier series (FS)*.

What This Means in Practice. In practice, the system we refer to in the impulse response of the system or PSF, is the microscope itself; thus, h represents the optical properties of the microscope (with h positive). Knowing these properties allows us to determine the PSF and thus, if necessary, reverse the effects of the imaging system on the image itself (remember, our goal is to have access to as "pure" a version of our specimen image $f(v)$ as possible). This is depicted in Fig. 3.1, where the input is $f(v)$, and the output is the filtered version $x(v)$.

3.1.2 A/D Conversion

While it is clear that if it were possible, the underlying image would have most of its parameters continuous in nature (space and time at least), it is also intuitively clear that this is not the output of a microscope. The reason for this is that there is a minimum distance at which the responses from two separate points in the sample can be distinguished. This minimum distance leads to the notion of *resolution* in microscopy, somewhat different from the same term applied in signal processing (where it denotes the amount of information present in a signal).

Sampling, or, discretization, can be performed without loss of information, if enough samples are taken to represent the continuous-domain image faithfully. Intuitively, this means that the continuous-domain image cannot "vary" too much as it would require too many samples to represent it well, that is, it must be somewhat "smooth". This can be mathematically made precise by saying that, if the continuous-domain image $x(v)$ is bandlimited, that is, if its FT is zero above a certain frequency, $|X(\omega)| = 0$, for $|\omega| > \omega_m$, then it can be reconstructed from its samples taken at twice the maximum frequency $\omega_s = 2\omega_m$. This is known under various names, Shannon sampling theorem among others [6]. The resulting samples are $x_n = x(\mu n)$, where $n \in \Omega_s \subset \mathbb{Z}^d$ is a set of index coordinates (for example, pixels when $d = 2$ or voxels when $d = 3$), and $\mu \in \mathbb{R}^d$ collects the sampling steps in each dimension. The original image $x(v)$ can then be reconstructed from x_n via

$$x(v) \;=\; \sum_{n \in \Omega_s} x_n \prod_{i=1}^{d} \mathrm{sinc}_{\mu_i}(v_i - n_i \mu_i), \tag{3.4}$$

where the sinc function is given as $\mathrm{sinc}_{\mu_i}(v_i) = sin(\pi v_i/\mu_i)/(\pi v_i/\mu_i)$.

Continuous-domain images are rarely truly bandlimited, and thus, filtering is first performed to bandlimit the image before sampling, resulting in $x(v) = (h * f)(v)$. This, in turn means that the filtered image $x(v)$ can be recovered perfectly from its samples x_n, though not the original one, $f(v)$. In practice, however, several issues (for example, noise, limited field of view, etc.) complicate the procedure and algorithms to recover a good estimate of $x(v)$ depend on many more assumptions to be detailed later in this chapter.

One way to account for various sources of uncertainty is to introduce the probability of observing the value α_n at coordinate n; this probability can be expressed as $\mathcal{P}(\alpha_n; x_n)$, where $\mathcal{P}(\cdot)$ is a probability model that accounts for the uncertainty introduced due to noise in several steps during discretization. As is the case with most photon detection-based instruments, the primary source of uncertainty introduced is due to photon counting noise, where $\mathcal{P}$ is given by a Poisson distribution:

$$\mathcal{P}(\alpha_n; x_n) \;=\; \frac{x_n^{\alpha_n} e^{-x_n}}{s_n!}. \tag{3.5}$$

Although photon detection instruments are inherently limited by Poisson noise, photon counting effect is not the only source of uncertainty in the image. Other artifacts such as thermal noise and digitization noise (uncertainty introduced by storing only a certain number of bits to describe each pixel value) also play a role and the overall probability model is more likely to be a mixture of several sources of uncertainty. This is modeled in Fig. 3.1 by the A/D block, where the input is the image $x(v)$, and the output, instead of being just a simple sampled version of the input x_n, is given as $g_n = \beta(x(v), \text{noise})$.

3.1.3 Discrete-Domain Image Processing

Framework. As we have seen, after A/D (discretization/digitization), the resulting image is g_n, with $n = (n_1, \ldots, n_d)$. For example, a gray-scale 2D time series with N_3 2D slices of size $N_1 \times N_2$, will be denoted as g_{n_1,n_2,n_3}, where n_1 and n_2 are two spatial coordinates with domains $n_1 = 0, \ldots, N_1 - 1$, $n_2 = 0, \ldots, N_2 - 1$ and n_3 is the time coordinate with domain $n_3 = 0, \ldots, N_3 - 1$.

We stress an important point: Assume, for simplicity, we deal with a digital image g_{n_1,n_2}, that is, just a single 2D slice. One can look at this image in two ways:

1. The domain of $n = (n_1, n_2)$ is infinite, that is both $n \in \mathbb{Z}^2$. We then assume that the image is of finite energy, written as $g_n \in \ell^2(\mathbb{Z}^2)$, and is a set of $N_1 \times N_2$ nonzero values in the plane. In this case, the discrete-time FT will be used (introduced in a moment) and the result of a convolution gives support larger than the one with which we started.

2. The domain of $n = (n_1, n_2)$ is finite, that is, $n_1 = 0, \ldots, N_1 - 1$, and $n_2 = 0, \ldots, N_2 - 1$. The image is of finite energy by construction. In other words, we assume that the digital image exists only on a discrete grid of size $N_1 \times N_2$. That also means that the convolution used is the circular convolution which preserves the domain, and the appropriate Fourier transform is the discrete Fourier transform (also introduced in a moment).

While the above distinction might seem unimportant, it has great consequences on the machinery we use.

Linear Shift-Invariant System and Digital Filters. Similarly to what was shown in continuous domain, most of the time we deal with LSI systems—filters. For the discrete domain, these filters can model the effects after A/D (discretization/digitization), or can be additional digital filters applied to achieve a certain effect. Their operation on a discretized signal g_n can be described via convolution again as

$$(a * g)_n \;=\; \sum_m g_m a_{n-m}. \tag{3.6}$$

Note that we use the same symbol $*$ for both continuous-domain as well as the discrete-domain convolution; the meaning is clear from the context. In signal processing, when the the domain of the digital image is finite (as it is in practice), one often uses the so-called *circular convolution*, which preserves the domain. In other words, while the convolution as defined in (3.6) produces a larger image as the result of filtering (how much larger depends on the filter), the circular convolution produces the image of the same size. These various convolutions may seem confusing and cumbersome; in fact, mathematically, one is able to deal with this problem in a unified way by defining the spaces of signals and filters, and then deriving the appropriate forms of convolution as well as the FT [7]. While that treatment is beyond the scope of this introduction, the interested reader is encouraged to look up [7] for more details.

The Fourier View. Similarly to what was done in continuous domain, we often use the Fourier machinery to analyze digital images. As explained previously, one must distinguish between images for which we assume the domain is infinite (yet with only a finite number of nonzero values), from those whose domain is finite (which is the case in practice). As we have seen, the convolution in these two instances is different; so is the FT. Without going into mathematical details, we just give those two versions of the FT. Again, here we use the standard nomenclature for the FT using time as the domain although the domain of images is space; this is done to avoid confusion when looking up relevant literature. To make matters simple, we give below 1D versions:

1. *Discrete-Time Fourier Transform (DTFT)*

$$F(e^{j\omega}) \;=\; \sum_{n\in\mathbb{Z}} f_n e^{-j\mu\omega n} \qquad \leftrightarrow \qquad f_n \;=\; \frac{\mu}{2\pi}\int_{-\pi/\mu}^{\pi/\mu} F(e^{j\omega}) e^{j\mu\omega n} d\omega \tag{3.7}$$

2. *Discrete Fourier Transform (DFT)*

$$F_k \;=\; \sum_{n=0}^{N-1} f_n W_N^{-nk} \qquad \leftrightarrow \qquad f_n \;=\; \frac{1}{N}\sum_{k=0}^{N-1} F_k W_N^{nk} \tag{3.8}$$

with $W_N = e^{j2\pi/N}$. For example, for images, the 2D DFT would have the form

$$f_{n_1,n_2} \;=\; \frac{1}{N_1 N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} F_{k_1,k_2} W_{N_1}^{n_1 k_1} W_{N_2}^{n_2 k_2}.$$

3.2 Image Analysis Tools for Microscopy

The aim in this section is to give an overview of tools, both those which are currently used as well as those we believe would be useful for solving imaging tasks in microscopy (see Section 3.3). As a plethora of such tools exist, we give a brief overview of only a subset of those; well established methods for image analysis and synthesis using both predetermined vectors (functions) as well as vectors automatically learned from data. Other important image processing and data analysis tools applicable to microscopic image analysis include pattern recognition and machine learning methods for clustering and classification [8], as well as statistical signal processing methods for detection and estimation [9, 10].

3.2.1 Signal and Image Representations

Signal representations (signal being a microscope image in this case) aim to represent a signal in a different domain so that its salient characteristics might become obvious in that domain. For example, a simple image which is a sum of two different frequencies in horizontal and vertical directions might be difficult to interpret in the original/image domain (see Fig. 3.2(e)), while its harmonic content is immediately obvious in the Fourier domain (having two single nonzero coefficients).

It is useful to consider signal/image representations in a more general framework; then, many of the tools we discuss are just specific instantiations of that framework (see [11, 12] for more details). Signals are usually assumed to "live" in a Hilbert space. Hilbert spaces are those spaces in which we are allowed to manipulate signals (add, multiply by a constant), measure them, compute inner products, etc. We are also allowed to represent our signals in terms of *bases* (nonredundant representations) or *frames* (redundant representations). These representations are typically written as

$$g \;=\; \sum_{i\in I} G_i \varphi_i, \tag{3.9}$$

where I is some index set, φ_i are basis/frame elements belonging to Φ, and G_i are called transform coefficients (subband, Fourier or wavelet, in case of specific transforms). Mathematically, $G_i\varphi_i$ is the projection of the signal onto the space spanned by φ_i and G_i is the value of that projection.[1] That projection is computed as

$$G_i \;=\; \langle \tilde{\varphi}_i, g\rangle, \tag{3.10}$$

where $\tilde{\varphi}_i$ form what is usually called a *dual basis/frame* $\tilde{\Phi}$.

The difference between bases and frames is that in the case of frames, there are more vectors than needed to represent a particular signal, making it a redundant representation. For example, any vector in a plane can be uniquely represented by two projections onto two noncolinear lines (subspaces). Projecting onto yet another line will not destroy the representation property; it will just make it redundant.

[1] We are not being entirely rigorous here as for infinite-dimensional spaces, bases have to be defined with some care; we gloss over those subtleties in this treatment.

Matrix View. It is possible to look at signal expansions using linear operators/matrices, offering a more compact, and very often, a more easily understood way of looking at representations. Equation (3.10) can be rewritten as

$$G \;=\; \tilde{\Phi}^* g, \tag{3.11}$$

where g is now a vector containing the input signal, G is the vector of transform coefficients and $\tilde{\Phi}$ is the matrix containing the dual basis/frame vectors as its columns. Then, (3.9) can be written as:

$$g \;=\; \Phi G \;=\; \Phi \tilde{\Phi}^* g, \tag{3.12}$$

from where it is obvious that for this to be a valid representation, $\Phi\tilde{\Phi}^*$ must be an identity. For bases, Φ and $\tilde{\Phi}$ are square and $\tilde{\Phi} = (\Phi^*)^{-1}$, while for frames, the frame Φ and the dual frame $\tilde{\Phi}$ are both rectangular.

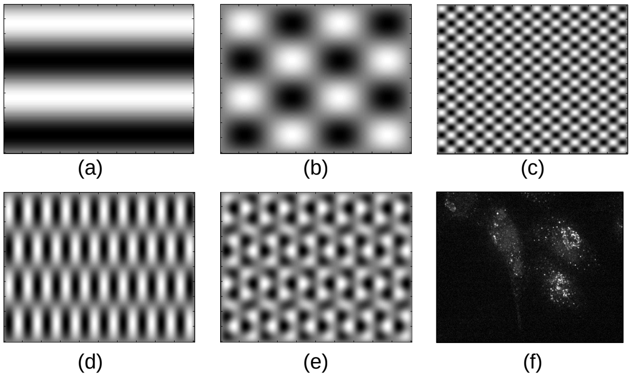

Figure 3.2: Example images. (a) Simple 1D sinusoidal image: $f(x, y) = \sin(\pi/50)x$. (b) 2D sinusoidal image of low frequency: $f(x, y) = \sin(\pi/50)x \sin(\pi/50)y$. (c) 2D sinusoidal image of high frequency: $f(x, y) = \sin(\pi/10)x \sin(\pi/10)y$. (d) 2D sinusoidal image of different frequencies in the two directions: $f(x, y) = \sin(\pi/50)x \sin(\pi/10)y$. (e) 2D image obtained as a sum of two frequencies in two directions: $f(x, y) = \sin(\pi/50)x \sin(\pi/10)y + \sin(\pi/14)x \sin(\pi/20)y$. (f) General image.

Spaces We Consider. For ease of notation, we will keep our discussion in 1D; extensions to more than one dimension are most often obtained by separate application of 1D concepts to each dimension in turn (Kronecker product).

Standard and most intuitive spaces with which we deal are real/complex Euclidean spaces $\mathbb{R}^N$ and $\mathbb{C}^N$. In these spaces, signals are considered to be vectors of finite length N and the inner product is the standard component-wise product. Then the index set I above is just the set $I = \{1, \dots, N\}$.

In discrete-time signal processing we deal almost exclusively with sequences g having finite square sum or finite energy, where $g = (\dots, g_{-1}, g_0, g_1, \dots)$ is, in general, complex-valued. Such a sequence g is a vector in the Hilbert space $\ell^2(\mathbb{Z})$.

Orthonormal Bases. In case of nonredundant representations, when $\tilde{\Phi} = \Phi$, and thus $\Phi\Phi^* = I$, we deal with *orthonormal bases (ONBs).* Both projection and reconstruction are performed using the same set of basis vectors Φ. Many properties hold for orthonormal bases:

Orthonormality of Basis Vectors. Since $\Phi^* = \Phi$, $\langle \varphi_i, \varphi_j \rangle = \delta_{i-j}$, that is, basis vectors are orthonormal.

Projections. A characteristic of orthonormal bases allowing us to approximate signals is that an orthogonal projection onto a subspace spanned by a subset of basis vectors, $\{\varphi_i\}_{i \in J}$, where J is the index set of that subset is $Pg = \sum_{i \in J} \langle \varphi_i, g \rangle \varphi_i$, that is, it is a sum of projections onto individual one-dimensional subspaces spanned by each φ_i. Beware that this is not true when $\{\varphi_i\}_{i \in J}$ do not form an orthonormal system.

Parseval's Equality. The *Parseval's equality* is simply the *norm-preserving* property of ONBs. In other words,

$$\|g\|^2 \;=\; \sum_{i \in I} |\langle \varphi_i, g \rangle|^2 \;=\; \sum_{i \in I} |G_i|^2. \tag{3.13}$$

Least-Squares Approximation. Suppose that we want to approximate a vector from a Hilbert space by a vector lying in the subspace $S = \{\varphi_i\}_{i \in J}$. The orthogonal projection of g onto S is given earlier by Pg. The difference vector $d = g - \hat{g}$ satisfies $d \;\perp\; S$. This approximation is best in the least-squares sense, that is, $\min \|g - y\|$ for y in S is attained for $y = \sum_i G_i \varphi_i$ with $G_i = \langle \varphi_i, g \rangle$ being the expansion coefficients. In other words, the best approximation is our $\hat{g} = Pg$. An immediate consequence of this result is the successive approximation property of orthogonal expansions. Call $\hat{g}^{(k)}$ the best approximation of g on the subspace spanned by $\{\varphi_1, \varphi_2, \dots, \varphi_k\}$. Then the approximation $\hat{g}^{(k+1)}$ is given by $\hat{g}^{(k+1)} = \hat{g}^{(k)} + \langle \varphi_{k+1}, g \rangle \varphi_{k+1}$, that is, the previous approximation plus the projection along the added vector φ_{k+1}.

Note that the successive approximation property does not hold for nonorthogonal bases. When calculating the approximation $\hat{g}^{(k+1)}$, one cannot simply add one term to the previous approximation, but has to recalculate the whole approximation.

Tight Frames. In case of redundant representations, when $\tilde{\Phi} = \Phi$, and thus $\Phi\Phi^* = I$, we deal with *Parseval tight frames (PTFs)* (when $\Phi\Phi^* = cI$, the frame is tight). Both projection and reconstruction are performed using the same set of frame vectors Φ (albeit possibly scaled). Note, however, that while for bases, $\Phi^*\Phi = I$ as well, this is not the case for frames. Orthonormality of frame vectors does not hold anymore as the vectors are not linearly independent, but a generalized sort of Parseval's equality still holds ($\|g\|^2 = \sum_{i \in I} c|G_i|^2$, with $c = 1$ for PTFs). More details on frames can be found in [11, 12].

How to Choose the Right Representation. From what we have described above, we have the option of choosing the nonredundant/redundant representation, and then, orthonormal/general for bases and tight/general for frames. Even once one of these four options is chosen, we are still left with the task of choosing the specific basis/frame elements. Fortunately, choices abound, and we now present some of those.

Time-Frequency Considerations. One of the often used criteria when choosing basis functions is how well they can represent certain local events in the original domain (space for images) as well as frequency. For example, a pure sinusoid is considered a perfectly local event in frequency, while a Dirac pulse is a perfectly local event in time (original domain). To see how a specific representation localizes in time (original domain) or frequency, one typically uses a conceptual tool called *time-frequency plane* (see Fig. 3.3 for examples). Consider any of the three bottom diagrams. Each has time (original domain) on the horizontal axis and frequency on the vertical. The tiles represent the time-frequency distribution of the energy of the basis functions of the corresponding transform. For example, in the DFT, the tiles are all rectangles splitting the frequency into pieces, but going over all time. This is because the basis functions for the DFT cover all time but only pieces of frequency. Thus, one can see that the DFT will work well to isolate events local in frequency but will not isolate those local in time.

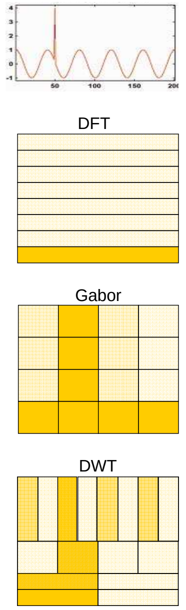

Figure 3.3: Energy distribution of a simple low-frequency signal with a Dirac pulse using different bases: DFT, Gabor and DWT.

3.2.2 Fourier Analysis

When one is interested in the harmonic content of a signal, Fourier analysis is the tool of choice. The reason for this is that the representation functions are sinusoids (complex exponentials).

Fourier Bases. Let us see how the Fourier view from the previous section fits into the representation framework we just introduced. We do this by looking at the DFT as a signal expansion (this is a short version of the same example in [11]), which, while ubiquitous, is rarely looked upon as a signal expansion or written in matrix form. The easiest way to do that is to write the reconstruction expression (3.8) in matrix notation as

$$g = \frac{1}{N} \underbrace{\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N & \cdots & W_N^{N-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & W_N^{N-1} & \cdots & W_N \end{pmatrix}}_{\Phi = \mathrm{DFT}_N} \underbrace{\begin{pmatrix} G_0 \\ G_1 \\ \vdots \\ G_{N-1} \end{pmatrix}}_{G} = \mathrm{DFT}_N \, G. \qquad (3.14)$$

The DFT matrix defined as above is not normalized, that is, $(1/N)(\mathrm{DFT}_N)(\mathrm{DFT}_N^*) = I$. If we normalized the above matrix by $1/\sqrt{N}$, the DFT would exactly implement an orthonormal basis. The decomposition formula (3.8) in matrix notation becomes:

$$G = \mathrm{DFT}_N^* \, g. \qquad (3.15)$$

The advantage of looking at the DFT in matrix form is that its operation now becomes "visual" and projections onto the corresponding subspaces are easily seen. For example, for $N = 2$, the DFT would project onto a space with "smooth" signals (averaging) as well as the space of "detail" signals (differencing). One can also see that each of the coefficients will extract a specific range of "frequencies" from the signal, that is, it will measure how "wiggly" the signal is. An important point about the Fourier analysis is that while it extract perfectly the frequency content of the signal, the local events in the original domain (for example, a spike in 1D or edge in 2D) are diluted over the whole frequency domain as each basis function will capture those local events. This is one of the reasons other techniques, such as wavelets, are used, where a trade-off between frequency and original domain localization can be achieved.

Fourier Frames. Similarly to Fourier bases, Fourier frames exist. They are known under the name *harmonic tight frames (HTFs)* and are obtained by projecting a Fourier basis from a larger space onto a smaller one. For example, given a DFT of size $M \times M$, we can obtain an HTF of size $N \times M$, with $N < M$, by deleting columns of the DFT. These representations possess similar properties to the DFT, except that they are redundant. As HTFs are possible for any combination of M, N, any redundancy can be achieved and the choice is governed by the application. More details on these frames can be found in [12].

3.2.3 Gabor Analysis

While the Fourier-type tools are best used on signals which are reasonably smooth as that global behavior will be captured in the frequency domain, they are not appropriate for nonstationary signals. A classic example is that of a signal consisting of a sinusoid and a Dirac pulse. The sinusoid is captured by the Fourier tools while the Dirac pulse is spread over all frequencies and its location is lost (see Fig. 3.3, DFT). One way to deal with the problem is to window the Fourier basis functions so that just those with the support over the Dirac pulse will contain information about it; all the other basis functions will not overlap and will thus contain no information about it. This leads to the *Gabor transform (GT)* (also called *short-time Fourier transform (STFT)*), and is pictorially shown in Fig. 3.3, Gabor. Thus, upon observing that the coefficients corresponding to the basis functions with support coinciding with the support of the Dirac pulse contain more energy, one can conclude that the location of the Dirac pulse is somewhere within the range of that window (the window is shifted along the original axis to cover the entire original domain). This method still suffers from the problem that the "zoom" factor of the analysis, or how finely we can pinpoint the Dirac pulse, depends on the choice of the window. One can write the same equations as (3.14)-(3.15):

$$g = \mathrm{GT}\, G, \qquad \mathrm{GT} = W\, \mathrm{DFT}_N, \tag{3.16}$$

$$G = \mathrm{GT}^*\, g, \tag{3.17}$$

where the GT is obtained by windowing the DFT by W. The window W is a diagonal matrix with window elements along the diagonal. Since the window multiplies the DFT_N on the left, it multiplies each DFT basis function in turn. The window W is typically chosen to be a symmetric lowpass filter.

3.2.4 Multiresolution Analysis

The *wavelet transform (WT)* and the *discrete wavelet transform (DWT)* arose in response to the problems the GT could not solve. The answer came as a tool with short basis functions at high frequencies together with long basis functions at low frequencies. As such, the WT is just the tool to tackle the above example (see Fig. 3.3, DWT). Unlike the DFT and GT, which both have fixed basis functions, the WT (or DWT) allows for different families with the same time-frequency tiling as in the figure. These are obtained by choosing different template wavelets and then forming the whole basis from the template by shifts and dilations. While the DWT is an appropriate tool for the above problem, it does not do so well if the frequency of the sinusoid is high, as then the position of the Dirac pulse cannot be distinguished in the high frequencies. *Multiresolution (MR)* techniques encompass a whole set of tools allowing for a rather arbitrary division of the time-frequency plane to match the signal at hand, including the one we just mentioned. As such, the DFT as well as the GT become subcases of it, with specific tilings.

Implementing MR Transforms Using Filter Banks. These various MR transforms are implemented using filter banks (FBs), signal processing devices which split the frequency spectrum into parts. The basic frequency splitting is achieved through filtering and sampling, while reconstruction is done in the reverse direction, upsampling and filtering again. The analysis filter bank maps the signal into transform coefficients (also called subbands) as in (3.10) or (3.11), while the synthesis filter bank reconstructs the original signal from those, as in (3.9) or (3.12).

Discrete Wavelet Transform. The DWT achieves the tiling as in Fig. 3.3, DWT, by iterating two-channel FBs on the lowpass (smoothing, coarse) channel. The number of iterations determines the number of leaves of the DWT (or any

MR transform) as well as the level of coarseness at the last level (the more levels, the coarser). The actual filters used determine the basis functions are many: Well-known families are Haar, Daubechies, Coiflets, and others [6, 13, 14]. Depending on the basis functions used, we can have ONBs or general bases (Haar and Daubechies are ONBs, Coiflets are general bases with symmetries). For example, the 3-level Haar DWT is given by the following matrix:

$$\mathrm{DWT}_{3,\mathrm{Haar}} \;=\; C \cdot \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \tag{3.18}$$

where C is a diagonal matrix of constants, $C = \mathrm{diag}(1/\sqrt{2}, 1/\sqrt{2}, 1/\sqrt{2}, 1/\sqrt{2}, 1/2, 1/2, 1/4, 1/4)$, needed to make the DWT an ONB. Many others are possible leading to different values in the above matrix (assuming three levels).

Wavelet Packets and Variations. As one could have split the coarse channel again, one could have done it with the highpass as well. Thus, one can obtain arbitrary trees, anywhere from a full tree with J levels corresponding to a 2^J GT, to a DWT with J levels, to an arbitrary tree with J levels (called *wavelet packet (WP)*).

One can also split the frequency range into N channels immediately, getting the DFT as a subcase. By splitting the N-channel filter bank further in any of the above ways, one obtains an even more arbitrary tiling of the time-frequency plane.

Recommendations. The first goal in choosing one of the above transforms is getting the one to represent the signal the best. In other words, the first concern should be matching the transform to the time-frequency distribution of the underlying signal (image). After that has been done successfully, the choice of particular filters and such may be attempted. However, choosing the appropriate transform is not easy; the idea behind WP is to grow a full tree and then prune back given a cost function, which might not be readily available. For example, in compression, people use the number of bits; thus, if pruning the tree at a particular branch leaves us with fewer bits, we do it; otherwise we do not. Most often, we do not have an appropriate cost function, as was the case in [15], where the authors solved the problem by assigning weights to all the nodes in the tree. Then, the cost function is implemented implicitly; low weights mean pruning of the branch, while high weight means the branch is kept.

3.2.5 Unsupervised, Data-Driven Representation and Analysis Methods

Except for the WP, the techniques described above use a predetermined set of vectors (functions) to represent signals and images. Alternatively, representation schemes can be built based on any available training data using automated learning approaches, an area of intense research. We now briefly review examples of data-driven representation and analysis methods, beginning with the *principal component analysis (PCA)* technique. Many more works for dimensionality reduction and data analysis exist (see, for example, [16, 17]), but for brevity these are not reviewed here.

PCA. PCA is a method for reducing the dimensionality of a data set by finding a projection of the original set of vectors $\{g_1, \ldots, g_M\}$, onto a lower-dimensional space, optimal in a mean-square (MS) sense. Assume that the dimensionality of the original space is N, that is, each $g_i \in \mathbb{R}^N$, and that we want to reduce the dimensionality to L, with $L \leq N$. PCA accomplishes this task by finding an L-dimensional ONB $\Phi = \{\varphi_1, \ldots, \varphi_L\}$, such that the error of the approximation between the original vectors and their approximations is minimized in the MS sense. In Φ, the projections of the original vectors are expressed as:

$$\hat{g}_i \;=\; m + \sum_{k=1}^{L} \alpha_{k,i}\varphi_k, \qquad i = 1, \ldots, M, \tag{3.19}$$

where $m = (1/M)\sum_{i=1}^{M} g_i$ is the sample mean. We now find $\varphi_k, k = 1, \ldots, L$, such that

$$\mathcal{E}(\varphi) \;=\; \sum_{i=1}^{M} \| \underbrace{m + \sum_{k=1}^{L} \alpha_{k,i}\varphi_k}_{\hat{g}_i} - g_i \|^2 \tag{3.20}$$

is minimized. It is fairly simple to prove [18], that the solution is found as the L eigenvectors φ_k, $k = 1, \ldots, L$, corresponding to the L largest eigenvalues of the so-called scatter matrix (sample covariance):

$$S \;=\; \frac{1}{M}\sum_{i=1}^{M}(g_i - m)(g_i - m)^T. \tag{3.21}$$

The coefficients $\alpha_{k,i}$ are the values of the projections, that is, they are given by

$$\alpha_{k,i} \;=\; \langle g_i, \varphi_k \rangle. \tag{3.22}$$

The eigenvectors φ_k can provide meaningful geometric insight into the distribution from which the samples $g_i, i = 1, \ldots, M$, were drawn. The first eigenvector φ_1 is the one for which the squared energy of the projection is maximum. The second eigenvector φ_2 is the one whose energy is also maximum, but it is constrained to be orthogonal to the first, and so on. Although PCA was originally developed as a statistical data analysis technique, modern uses of it include finding optimal linear subspaces for representing image data.

It is worth noting that while the terms Karhunen-Loève (KL) and PCA are often used interchangeably in the literature, a distinction can be made in that PCA refers to the diagonalization of the sample covariance matrix, while KL refers to the diagonalization of an ensemble covariance matrix.

ICA. The PCA framework above is simple, efficient to compute, and extensively used in signal and image processing as well as general data analysis applications. While PCA finds orthogonal directions which best represent the data in the MS sense, *independent component analysis (ICA)* [19, 20] finds directions which are most independent from each other. When the underlying distribution for the data is a multivariate Gaussian one, the coefficients α_k are uncorrelated, and therefore independent. If the underlying data does not originate from a multivariate Gaussian distribution, correlation does not imply statistical independence and approximately independent coefficients may be obtained using ICA. Thus, the goal in ICA is to find vectors $\varphi_k, k = 1, \ldots, L$ which produce coefficients α_k that are not only uncorrelated but statistically independent as well. The process of computing independent components involves gradient-based nonlinear searches, normally initialized with results obtained from PCA. Implementation of this technique also depends on a proper definition of independence. The Kullback-Leibler distance [21] (not a true distance per se, but nonnegative nonetheless) is often used as a measure of statistical independence for ICA. However, because of issues related to computational complexity, practitioners are restricted to using surrogates such as kurtosis coefficients [19]. ICA is most used for deciphering the components, which, through addition, form a specific signal of interest, that is, they physically come from multiple sources (see [22] as well as Hyvärinen's web site for examples and demos [23]).

KPCA. Other extensions of the PCA framework include *kernel PCA (KPCA)* and *generalized PCA (GPCA)*. The idea in KPCA is to search for structure in the data by embedding it into a higher-dimensional (possibly infinite) space through a function $\gamma(g_i)$ taking values in the feature space Γ. The inner product in Γ is defined as $\langle g_i, g_j \rangle_\Gamma = \langle \gamma(g_i), \gamma(g_j) \rangle$, and defines a kernel $\Lambda(g_i, g_j)$ which can also be decomposed into a set of eigenvalues and eigenvectors to be used to estimate a low-dimensional representation of the original data (see [24] for more details about the computations involved). While KPCA can be used to extract interesting information from a set of data points, there is no general theory for choosing the kernel functions Λ. Polynomials and Gaussian kernels are often used, although their optimality for different applications is difficult to ascertain. Many of the modern methods for nonlinear dimensionality reduction can be interpreted as KPCA but with different kernel matrices [25].

GPCA. GPCA is a fairly recent contribution [26] with the same goal as the PCA, except that GPCA does not restrict the subspaces to be orthonormal with respect to each other, and each can be of a different dimension. In addition, GPCA identifies the membership of each data point. It starts by estimating a collection of polynomials from the data. Although the polynomials themselves are nonlinear, their coefficients can be estimated linearly from the data. Next, one point per subspace is segmented with clustering techniques, and a basis for the subspace passing through that point is obtained by evaluating appropriate derivatives.

ISOMAP. With the exception of KPCA, the approaches above provide tools for finding linear subspaces, or a collection of linear subspaces, to analyze and represent image data. Given sufficient data, a larger class of nonlinear manifolds can be recovered using the *ISOMAP* algorithm [27], which can be described in two relatively simple steps. First, the ”geodesic” distance between data points is estimated, by constructing a neighborhood graph and then finding the shortest path through the graph that connects the two points. (This distance is not the usual Euclidean distance, but rather a distance that is adapted to the underlying geometry of the data.) The classical multidimensional scaling method [28] is used to compute low-dimensional coordinates for each point. The decay of the variance of the residual of the approximations as a function of the number of approximating dimensions can then be used to estimate the true dimensionality (free parameters) of the data.

LLE. Another class of nonlinear manifold learning algorithms involves searching for low-dimensional representations that best preserve local structures of the data. The *linear local embedding (LLE)* [29] algorithm is an example of such an approach. Like ISOMAP, the first step in the LLE algorithm is to compute the P nearest neighbors of each data point g_i. The second step is to assign weights $\alpha_{k,i}$ to each nearest neighbor of each point such that

$$\mathcal{E}(\alpha) \;=\; \sum_{i=1}^{M} \left\| g_i - \sum_{k=1}^{P} \alpha_{k,i} g_k \right\|^2 \tag{3.23}$$

is minimized subject to the constraints that $\alpha_{k,i} = 0$ if g_k is not amongst the P nearest neighbors of g_i, and that $\sum_{k=1}^{P} \alpha_{k,i} = 1$ for all i. In the final step, LLE computes data points $\hat{g}_i \in \mathbb{R}^L$, with $L < N$, that best preserve the local properties of the data as represented by the sets of weights computed in the previous step, by minimizing

$$\mathcal{E}(\hat{g}) \;=\; \sum_{i=1}^{M} \left\| \hat{g}_i - \sum_{k=1}^{P} \alpha_{k,i} \hat{g}_k \right\|^2, \tag{3.24}$$

and computing the bottom $M + 1$ eigenvectors of the matrix $(I - A)^T (I - A)$, where $A_{k,i} = \alpha_{k,i}$.

3.2.6 Statistical Estimation

The random variations introduced by system noise, artifacts, as well as uncertainty originating from the biological phenomena under observation, require stochastic image processing methods. In fact, several tasks and applications for microscopy we are about to review can be understood as statistical estimation problems [9], where the goal is to seek the solution to the problem at hand optimal in some probabilistic sense, requiring one to adopt some optimality criterion.

MSE Estimators. The mean-squared error (MSE) between the estimate $\hat{\theta}$ of some quantity θ, denoted as $\mathrm{MSE}(\hat{\theta}) = \mathrm{E}\left\{(\hat{\theta} - \theta)^2\right\}$, where E is the expectation operator, is a desirable estimate since it minimizes the error of the estimate (on average). The Wiener filtering method given in (3.29) is an example of such an estimate. However, MSE estimators often depend on unknown parameters, leading to other desirable estimators, such as minimum-variance unbiased ones.

MAP Estimators. An interesting alternative is provided by the Bayesian framework. Let g represent the observed image data and let b represent some pattern or hidden feature(s). In the Bayesian framework, an estimate of b is computed by maximizing the a posteriori probability (MAP)

$$p(b|g) \;=\; \frac{p(g|b)}{p(g)} p(b). \tag{3.25}$$

Since $p(g)$ is constant with respect to b, maximizing (3.25) is equivalent to maximizing

$$p(g|b)p(b), \tag{3.26}$$

where $p(g|b)$ is the so-called probabilistic data model, and $p(b)$ is interpreted as a prior bias on b. Taking logarithms, the problem in (3.26) is equivalent to minimizing the following cost function

$$\Psi(b) \;=\; \log p(g|b) + \log p(b), \tag{3.27}$$

whose solution is the estimate $\hat{b}$ we want to find:

$$\hat{b} \;=\; \arg\max_{b} \Psi(b). \tag{3.28}$$

The MAP problem is often recast as a minimization instead of maximization, by taking the negative of the quantities above. When no prior knowledge on b is available (for example, $p(b)$ is a uniform distribution), (3.28) is equivalent to a Maximum Likelihood (ML) estimate of b.

Closed-form solutions for computing MAP estimates are rarely available due to the nonlinearity present in nearly all but the most trivial problems. Instead, one is often required to solve (3.27) numerically. When b and g are interpreted to belong to a Hilbert space of continuous functions, the Bayesian framework reduces to solving a variational calculus problem. The forms of $p(g|b)$ and $p(b)$ specify the required function spaces to which g and b must belong (for example, square integrable, Sobolev, bounded variation, etc.). A common approach is to derive maximization (minimization) approaches based on Euler-Lagrange equations from the maximization problem. Many algorithms in image restoration, segmentation, and registration discussed later in the chapter can be viewed within this framework.

Gradient-Based Estimators. When b is a finite-dimensional vector, standard gradient-based optimization approaches can be used, the most common of which is the steepest descent method. The cost function $\Psi(b)$ is given by (3.27), where b is now a finite-dimensional vector. The steepest descent method computes $\hat{b}$ by solving $\nabla_b \Psi(b) = 0$ through the following three-step iterative approach: The estimate $\hat{b}_k$ is iteratively updated by (1) computing the gradient of the objective function $\nabla_b \Psi(b) \mid_{b=\hat{b}_k}$, (2) finding τ such that $\Psi(\hat{b}_k - \tau \nabla_b \Psi(b) \mid_{b=\hat{b}_k})$ is minimized, and (3) updating $\hat{b}_{k+1} = \hat{b}_k - \tau \nabla_b \Psi(b) \mid_{b=\hat{b}_k}$. Often, step (2) is bypassed and τ is fixed to a small value.

3.3 Imaging Tasks in Microscopy

The goal in an imaging-based biological experiment is to extract structural, spatial, and functional quantitative information about some biological phenomenon accurately and, if possible, automatically. We now briefly review some of the canonical problems in microscopic image analysis for extracting such information, such as restoration, registration, segmentation and others. A general view of the system, which this section follows, is given in Fig. 3.1.

3.3.1 Intelligent Acquisition

Although the general process of acquisition was described in Section 3.1, a new set of techniques aiming at providing intelligence during the acquisition process has emerged.

The first motivation for these approaches is to enhance resolution. In laser scanning confocal microscopy, images are acquired line-by-line, pixel-by-pixel [30]. We can achieve significant time savings by only imaging those regions where we expect to find an object. These time savings could then be used to increase the frame rate, or to acquire the selected regions at a higher spatial resolution.

The second motivation is to reduce photobleaching and phototoxicity. In fluorescence microscopy, images are acquired by shining excitation light on the specimen to activate fluorescence. However, this can damage the fluorescent signal (photobleaching) [31], as well as the cell itself (phototoxicity) [32], thus limiting the duration over which we can view a cellular process. By reducing the total area acquired in each frame, we reduce the overall exposure to excitation light, hence reducing both photobleaching and phototoxicity.

Intelligent acquisition of microscope images has not been studied until recently. In [33], the authors designed an algorithm to reduce the number of pixels sampled in a 2D or 3D image when using a laser scanning confocal microscope. They observed that a large portion of scanning time is spent on low fluorescence regions, which presumably contain little useful information. The approach is then to begin by scanning the field at a low resolution. Each scanned value is examined, and if found to be significant, the area around it is scanned at a higher resolution. The process is repeated iteratively.

In [34] instead, the authors provide intelligence while learning the data model. They study a large number of tiny moving objects over a sustained period of time. To assist with efficient acquisition, they develop and continually refine a model to describe the objects' motion. In [35], the authors provide algorithms for modeling objects' motions for the purposes of tracking, and although not used directly, their work helped inspire the approach in [34].

3.3.2 Deconvolution, Denoising and Restoration

Microscope images typically contain artifacts that, if accentuated, may prevent reliable information extraction and interpretation of the image data. Two main sources of artifacts can be identified: blurring caused by the PSF h (see (3.2)), and noise arising from the electronics of A/D conversion (see Fig. 3.1). The combined tasks of deconvolution (deblurring) and denoising are generally referred to as image enhancement or restoration, as shown in Fig. 3.1. In the figure, two parallel paths are possible: (1) Joint denoising and deconvolution known as restoration, which as input has the output of the microscope g_n, and as the output, the estimate $\hat{f}_n$ of the input image $f(v)$. Note that while the output of restoration is another digital image $\hat{f}_n$, the problem itself is posed in terms of its continuous-domain version, that is, finding the best estimate $\hat{f}(v)$. Our discussion in this section focuses mostly on this path. (2) Separate tasks of denoising, having as input has the output of the microscope g_n, and as the output, the estimate $\hat{x}_n$, followed by deconvolution.

The problem of deconvolution in the presence of noise dates back many decades and has been applied to a variety of imaging problems related to astronomy, medicine, and many others, in addition to microscopy (for recent reviews, see [36–38]). Our purpose here is not to provide an extensive review of existing methodology, but rather an overview of important concepts often used, their relationship to Fourier analysis, as well as more modern ideas based on wavelets and sparse representations.

The plethora of restoration methods available can be classified according to different criteria. Some of the terminology associated with different methods available include linear versus nonlinear, least squares, maximum likelihood, expectation maximization, blind versus model-based, and others. Here we describe two different optimization criteria based on which several different methods have been designed, beginning with the minimum MSE estimation. All of the methods being described assume an LSI degradation model (PSF), and can be applied in two or three dimensions.

MSE Estimation. The approach used here was described in Section 3.2.6. Let $f(v)$ represent an image one wishes to reconstruct, ideally, by undoing the effects of the PSF by inverse filtering operation h_{in} on some measured data $g(v)$, $\hat{f}(v) = (h_{\text{in}} * g)(v)$. The measured data $g(v)$, the original image $f(v)$, as well as its estimate $\hat{f}(v)$, can all be viewed as random variables due to the uncertainty introduced by noise sources. Thus, a reasonable criterion to minimize is the MSE between the linear estimate $\hat{f}(v)$ and the real image $f(v)$, that is, $\mathrm{E}\{(f - \hat{f})^2\}$. Under assumptions explained below, it is possible to derive the well-known Wiener filter [39] solution to the problem, expressed in the Fourier domain as:

$$\hat{F}(\omega) \;=\; \underbrace{\frac{H(\omega)}{|H(\omega)|^2 + S_e(\omega)/S_g(\omega)}}_{H_{\text{in}}(\omega)} G(\omega), \tag{3.29}$$

where $H(\omega)$ is the Fourier transform of the PSF, $S_e(\omega)$ is the power spectral density of the noise source, and $S_f(\omega)$ is the power spectral density of the image being measured. The derivation above assumes that both the image and the noise source are well modeled by ergodic random variables, as well as that the noise is additive and white (uncorrelated). These assumptions are often violated in microscope imaging experiments. Moreover, the quantities $S_e(\omega)$ and $S_g(\omega)$ are seldom known and practitioners often replace the ratio $S_e(\omega)/S_g(\omega)$ by some constant c.

MAP Estimation. The approach used here was also described in Section 3.2.6. Here, we denote the ensemble of possible images by b, while the output estimate will be called $\hat{f}$. That is, we seek to maximize the posterior probability $p(b|g) = (p(g|b)/p(g))p(b)$, where $p(g)$ does not depend on b, and $p(b)$ represents the prior knowledge one may have about the image b. The cost function is the following functional (a version of (3.27)):

$$\Psi(b) \;=\; \phi(g,b) + cP(b), \tag{3.30}$$

where the logarithms have been subsumed into the above terms, $P(b)$ is a regularization function derived based on *a priori* knowledge, and c is an arbitrary constant. Its role is to prevent the solution to the algorithm from containing certain undesired characteristics such as excessive oscillations. Restoration algorithms are normally set up as minimization problems by defining the terms above as the negative of the log of the probabilities.

Modeling the Minimization Function. When the Poisson distribution is used to model $p(g|b)$, the minimization term ϕ is defined as:

$$\phi_{\mathcal{P}}(g,b) \;=\; \int \left[(h * b)(v) - g(v)\log(h * b)(v))\right] dv, \tag{3.31}$$

while the equivalent for a Gaussian distribution model for $p(g|b)$ is:

$$\phi_{\mathcal{G}}(g,b) \;=\; \int |(h*b)(v)-g(v)|^2\,dv \;=\; \|h*b-g\|_{L_2}^2. \tag{3.32}$$

Several algorithms can be derived to optimize these functionals. A classical example is the Richardson-Lucy (RL) algorithm: an iterative, unregularized algorithm that minimizes $\phi_{\mathcal{P}}(g,b)$, one out of a class of methods known to produce result images $\hat{f}$ dominated by noise as the number of iterations increase. The addition of different priors on b, together with other constraints such as positiveness, can help overcome these difficulties.

Modeling the Prior. Many different priors on b, $P(b)$, have been used in the literature. One of the most often used is the L_p-type regularization: $P_{\mathcal{Q}}(b) = \|Db\|_{L_p}^p$, where D is a linear operator (often of a differential type) and $p = 1$ or 2. When $p = 2$ this regularizer tends to minimize the energy of either the image itself ($D = 1$) or properties of the image (for example, derivatives $D = \nabla$). This leads to a Tikhonov-type regularizer [40], which tends to produce blurred estimates $\hat{b}$, countering the effect of the $\phi(g,b)$ minimization term. In contrast, $p = 1$ leads to so-called ”sparsity” maximizing solutions which tend to preserve edge structures better. Examples include total variation regularization terms ($D = \nabla$) [41] and wavelet-based sparse representations (when Db refers to coefficients of the WT of the image) [42].

Other regularization methods include entropy-type priors where one uses a model $m(v)$ which represents the prior knowledge about the image. The entropy prior is defined as

$$P_{\mathcal{E}}(b) \;=\; \int \left[b(v)-m(v)-b(v)\log\frac{b(v)}{m(v)}\right]dv. \tag{3.33}$$

This entropy functional is an optimal distance measure for positive functions [43]. However, since a precise model for the image to be measured is often not available, researchers typically use a constant function leading to a preference for smooth functions [37].

Other regularization approaches exist, such as the Good’s roughness penalty [44] which often yields estimates of good quality in comparison with other methods [37]. In addition, for situations when a precise model for the PSF is unknown, there exist so called ”blind” restoration algorithms that seek to estimate both the image as well as the PSF. One notable example is based on the RL iterative algorithm described above [45].

Finally, in actual implementation, the overall functional $\Psi(b)$ (consisting of any combination of the terms $\phi_{\mathcal{P}}$, $\phi_{\mathcal{G}}$, $P_{\mathcal{E}}$, $P_{\mathcal{Q}}$) is discretized at the measured image grid of pixels and the functional is minimized using standard approaches such as the steepest-gradient descent (see Section 3.2.6), conjugate gradient descent, Newton-type as well as multiplicative methods [36, 37, 41]. Computational complexity is often a concern. Linear methods, such as the Wiener filter explained earlier, can be implemented in real time. More recent, nonlinear methods, such as the one described in [42], are more computationally demanding, although recent advances [46] may help overcome these difficulties.

3.3.3 Registration and Mosaicing

Image registration refers to the task of finding the spatial relationship and alignment between two or more images. Registration methods are useful for combining the information contained in multiple images of the same object, acquired by different instruments, and at perhaps different resolutions. Other applications include mosaicing as well as tracking objects in image time series. Here we provide an overview of image registration methods applicable to microscope images beginning with semi-automatic landmark-based registration, followed by fully automated intensity-based methods.

Registration. Let $\Omega_{1,i} \in \mathbb{R}^d$, $i = 1, \cdots, N_1$, and $\Omega_{2,k} \in \mathbb{R}^d$, $k = 1, \cdots, N_2$, be the pixel coordinates (in units of meters) of two images $\hat{f}_1(\Omega_{1,i})$ and $\hat{f}_2(\Omega_{2,k})$, respectively. The goal in a registration (alignment) problem is to find a spatial transformation β that relates the coordinates of the source image to the coordinates of the target image: $\tilde{\Omega}_{1,k} = \beta(\Omega_{2,k})$. The value of the image $\hat{f}_1$ at position $\tilde{\Omega}_{1,k}$ does not exist in general, since the image $\hat{f}_1$ is only defined at coordinates $\Omega_{1,i} \in \mathbb{R}^d$, $i = 1, \cdots, N_1$. However, a reasonable guess may be computed by using standard, continuous, representations of the image $\hat{f}_1(\Omega_1) = \sum_i \hat{F}_{1,i}\varphi_i(\Omega_1)$, as described in Section 3.1, to compute $\hat{f}_1(\tilde{\Omega}_{1,k})$.

With the knowledge of a set of N corresponding points $p_{1,k} = p_{2,k}$, $k = 1, \cdots, N$, one is often able to compute a spatial transformation β by solving the following minimization problem:

$$\beta^{op} \;=\; \arg\min_{\beta \in \mathcal{C}} \frac{1}{N} \sum_{k=1}^{N} \|\beta(p_{1,k}) - p_{2,k}\|^2, \tag{3.34}$$

where $\mathcal{C}$ defines a certain class for the spatial transformation β and $\|\cdot\|$ is the standard vector norm. The problem above is akin to the well-known Procrustes problem, and for the class of rigid-body transformations (rotations plus translations), the closed-form solution is known [47]. One first removes the mean from the coordinates $p_{1,k} - \bar{p}_1$ and $p_{2,k} - \bar{p}_2$, with $\bar{p}_1 = \frac{1}{N}\sum_{k=1}^{N} p_{1,k}$ and $\bar{p}_2 = \frac{1}{N}\sum_{k=1}^{N} p_{2,k}$ (then these mean-zero values are assigned to $p_{1,k}$ and $p_{2,k}$). Define the matrix $K = P_1^T P_2$, where P_1 and P_2 are matrices with each row comprising of a vector $p_{1,k}$ and $p_{2,k}$, respectively. The singular value decomposition $K = UDV^T$ can be used to compute the rotation matrix R that aligns the two point clouds:

$$R \;=\; V \triangle U^T, \tag{3.35}$$

with $\triangle \;=\; \mathrm{diag}(1, 1, \det(VU^T))$ as an example in three dimensions [48]. The translation component is given by $a = \bar{p}_2 - R\bar{p}_1$ and the final transformation is given by $\beta(p_1) = Rp_1 + a$.

The framework above can be extended to include other classes of spatial transformations, such as the set of transformations composed of linear combination of radial basis functions. If at least N basis functions are used, with minor assumptions, two point clouds can be matched exactly (that is, the error in (3.34) is 0). One often used class of radial basis functions are thin-plate splines (see [49] for an example).

The methodology above is not used universally as the corresponding landmark points are often hard to obtain. Automated landmark extraction methods are difficult to implement while manual landmark selection is cumbersome, time consuming, and often imprecise. Another important class of image registration methods are those that operate directly on the intensity values of the images $\hat{f}_1$ and $\hat{f}_2$ by solving a different optimization problem

$$\beta^{op} \;=\; \arg\min_{\beta \in \mathcal{C}} \Upsilon\left(\hat{f}_1, \hat{f}_2, \beta\right), \tag{3.36}$$

where $\Upsilon(\cdot)$ refers to an objective function normally composed of (1) a (dis)similarity measure between the intensity values of the target image $\hat{f}_2(\Omega_1)$ and warped source image $\hat{f}_1(\beta(\Omega_1))$ and (2) a constraint, or, regularization term so as to "bias" β towards any available prior information. Many different methods for intensity-based image registration exist and can be classified according to the type of spatial transformation β, the objective function Υ, and optimization method chosen (for comprehensive reviews, see [50,51]). Spatial transformations often used include rigid body, affine, polynomial, linear combination of B-splines or radial basis functions, as well as elastic and fluid deformation models. Objective functions currently in use in the literature include the sum of squared differences between the intensity values of the target image $\hat{f}_1(\Omega)$ and warped source image $\hat{f}_2(\beta(\Omega))$, their correlation coefficient, as well as their mutual information [52]. Optimization methods used include Powel's direction set method [53], gradient descent, conjugate gradients, as well as Newton-type methods.

The landmark and intensity-based methods are not mutually exclusive and can be combined by adding the two terms into a single optimization problem:

$$\beta^{op} \;=\; \arg\min_{\beta \in \mathcal{C}} \frac{c_1}{N} \sum_{k=1}^{N} \|\beta(p_{1,k}) - p_{2,k}\|^2 + c_2 \Upsilon\left(\hat{f}_1, \hat{f}_2, \beta\right), \tag{3.37}$$

where c_1 and c_2 are arbitrary constants. For an example of such an approach, see [54].

Mosaicing. Automated and semi-automated image registration methods are essential for building image mosaics. Due to the limited field of view of magnification objectives, this operation is often necessary for obtaining a more global view of the object, but with sufficient resolution for intricate analysis. Automated stages for mosaicing exist, but are often not accurate enough [55]. Large mosaics may be built with the aid of image registration methods given slightly overlapping images (a recent example can be found in [56]). Other applications of image registration to microscopic image analysis include electrophoresis image alignment for protein analysis [57], as well as studies of tissue differentiation during the evolution of Drosophila melanogaster embryos [54].

3.3.4 Segmentation, Tracing, and Tracking

Broadly speaking, segmentation and tracing refer to the detection of relevant contiguous objects within an image as well as the determination of their positions. Segmentation and tracing methods allow for localized analysis within an image and are essential for extracting information pertaining to one or more specific objects (for example, cells, organelles) in an image, while tracking refers to the detection or estimation of objects as a function of time in time series data sets. Naturally, manual image segmentation, tracing and tracking are all possible with the aid of modern computer display systems. However, given the enormous quantities of data produced by modern imaging systems, manual image interpretation and information extraction is not only costly, but also inaccurate and has poor reproducibility. We now briefly review some of the automatic and semi-automatic methods for cell and nuclear segmentation and neuronal tracing.

Segmentation, tracing and tracking methods are typically the first step in many imaging applications and have been applied to the classification and clustering of cellular shape [58], cell biomass computation [59], leukocyte detection and tracking [60], neurite tracing [61], cell migration [62], subcellular analysis [63] and studies of the influence of Golgi-protein expression on the size of the Golgi apparatus [64], among others.

Segmentation. The simplest and most widely used automated image segmentation method available is that of thresholding. This method consist of assigning the label of background to every pixel in the image whose value falls below a chosen threshold, while the label of foreground is assigned to each pixel that matches or exceeds the value of the threshold. More advanced thresholding methods choose the value of the threshold adaptively, through computation of global, or at times local, image histograms. However, thresholding methods alone are seldom effective as microscope images contain noise and are not illuminated uniformly, as well as because they neglect to account for geometric information in the data. These methods are thus commonly used only as initialization to other, more elaborate ones.

Voronoi-Based Segmentation. One such, relatively simple, technique is known as Voronoi diagram-based segmentation, often used when more than one relevant object (for example, cells, nuclei) are present. The idea is to use coordinates within each object (one coordinate per object to be segmented, obtained with the help of thresholding techniques) as representative coordinates for the entire object, and to ”draw” edges and vertices so that each coordinate is enclosed by a single polygon. For example, this technique is often used to segment cell nuclei in images of DNA probes [65]. However, it does not perform well when the objects being segmented are asymmetric or are close in space. In these situations, a single point is too simplistic a description for the geometry of the object, and the edges computed may not respect the boundary of the objects being segmented.

Watershed Segmentation. A more attractive technique is watershed segmentation, which can capture intricate object boundaries without overwhelming computational cost. The basic idea is to view a 2D image, for example, as a topographical surface, and ”flood” the surface from its local minima by considering its intensity level sets. When two regions are merging, a dam is built to represent an edge and boundary in the segmentation result. This technique can be applied directly to the raw image data as well as to processed images, such as edge-enhanced ones or distance-transformed ones. However, it is known to oversegment images and careful seeding (initialization) must be used [66, 67].

Active-Contour Segmentation Methods. The segmentation methods described above rely on a discrete interpretation of the image data in that any continuous properties of the objects being imaged are disregarded. In the past couple of decades, a different class of segmentation algorithms that explicitly include continuous information, such as curvature, has emerged, and is generally denoted as deformable models or active contours [68]. An active contour is a closed curve $C(l)$, with l some parameterization $l \in [0, 1]$, and $C(0) = C(1)$ (in two dimensions, such a curve is represented by $C(l) = (C_x(l), C_y(l))$). Active-contour segmentation methods seek to find the contours which best delineate different objects in an image. Kass et al. [69] formulate this as a variational optimization problem, with a cost function given by:

$$\Psi(C) \;=\; c_1 \int_0^1 \left|\frac{dC(l)}{dl}\right|^2 dl + c_2 \int_0^1 \left|\frac{d^2C(l)}{dl^2}\right|^2 dl + c_3 \int_0^1 \left|\nabla r(C(l))\right|^2 dl, \tag{3.38}$$

where c_1, c_2, c_3 are arbitrary constants, and $\nabla r(v)$ represents the gradient of the image r. The above can be a sum of other applicable forces as well (see [70] for an example that includes a stochastic term). The cost function is minimized through variational methods [68] and leads to a partial differential equation-type solution to the problem, $dC/dt = F(C, r)$, with t being a time variable artificially included, and $F(C, r)$ representing a force (or sum of forces) term derived based on (3.38). This equation can be solved with Euler integration methods. Another interesting aspect of such methods is that the curve $C(l)$ is normally parameterized using interpolating splines, for example. Parametric contours can be difficult to generalize to multiple object and changes in topology, as a reparameterization of the curve(s) is necessary. Again, as in the tasks described previously, the problem is posed in a continuous-domain setting and later discretized.

An alternative technology for propagating contours is based on the level-set methods [71], which avoid parameterization of the curve C by defining it implicitly as a level set (normally the zero level set) of a function $\phi(v)$ ($\{v|\phi(v) = 0\}$). For convenience, $\phi(v)$ is normally defined over the same domain as the image data $r(v)$. The level-set function ϕ is usually initialized as a signed distance function to an initial contour drawn over the image, and a typical implementation involves the following differential equation:

$$\frac{d\phi}{dt} = V(\kappa)|\nabla\phi|, \tag{3.39}$$

where $V(\kappa)$ is the so called speed function (normally involving image edges), while κ is the curvature of the contour, which can be computed directly from ϕ. As mentioned previously for the active-contour equation (3.38), the corresponding level-set equation can be given as the sum of applicable forces (see [70], for example). The solution is obtained in steady state (the contours do not evolve anymore) using Euler-type integration methods. There are several advantages to the level-set formulation. As mentioned above, changes in topology are handled automatically without extra effort. In addition, it is an Eulerian formulation, meaning that all computations are done with respect to a fixed grid (as opposed to tracking a determined amount of propagating ”particles” defining a contour). Therefore the technique can easily be extended to three or more dimensions (parameterized front propagation methods are complicated in three or more dimensions, especially if changes in topology are involved). The disadvantage of the level-set method is that the level-set function ϕ needs to be computed throughout the domain of the image, even though only a contour (or sets of contours) is desired. Methods for computing ϕ only in the neighborhood of its zero level set exist [71], although the speed-up is not substantial when the images consist of many contours close to each other. A novel approach to this computational problem is presented in [64]. The authors combine the power of active contours with the multiresolution (MR)/multiscale (MS) framework to compute forces using convolutions at different scales. This approach eliminates the need for the level-set framework and extensive computations, leading to a new class of active-contour methods dubbed multiscale active contours (MSAC) [64], with computational savings of 1-2 orders of magnitude. Modern image segmentation methods based on level sets can be generalized to include many different force terms and may not require image gradients, allowing for robust segmentation of objects without strong edges (see, for example, [72]). Other methods are based on the STACS algorithm we mentioned earlier [70], as well as follow-ups to that work, such as TPSTACS [73] as well as MSAC described above [64].

Tracing. The problem of tracing elongated pathways in image data is essentially a segmentation problem. The overall goal is the same as in segmentation: to detect and determine the location of a structure of interest. The significant geometric differences between neurons (normally elongated and not necessarily closed contours) and cells, for example, call for the development of a different methodology. Methods based on skeletonization [74] often fail because of the presence of noise, out-of-focus noise, and illumination artifacts. Those techniques that do not depend on skeletonization rely on a local path following approach. A starting point is given and a step is taken along a direction computed from the local intensity values of the image data. The approach can be summarized as an ordinary differential equation problem, where one seeks to find a path $C(l)$ by solving

$$\frac{dC(l)}{dl} = t(l), \tag{3.40}$$

where $t(l)$ is a vector that represents the tangential direction of the axon at location $C(l)$ in the image. The tangential direction can be computed by matching, locally, a cylindrical model for the axon, as in [75]. Alternatively, $t(l)$ can be computed from the eigenvectors of the Hessian (matrix of second derivatives) of the image data locally as in [61]. Such path following approaches often fail in the presence of noise and other artifacts (imperfect illumination, etc.), and thus typically require significant user interaction [61].

Tracking. Image-based tracking refers to the detection of relevant objects in image time series as well as the determination of their spatial positions in time. Tracking has been applied to modeling and understanding of biological molecular dynamics [76, 77], as well as to understanding cell migration [60, 62]. In a broad sense, tracking can be thought of as a time-dependent segmentation problem, and as in static image segmentation, a variety of methods exist. A comparison of methods for tracking single fluorescent particles is given in [78]. The simplest tracking algorithm consists of identifying potential molecules or cells by thresholding (possibly followed by morphological operations to remove holes and spurious artifacts) and then performing nearest neighbor matching. While this approach may be fruitful for tracking fluorescence particles in image data, it performs poorly when the objects being tracked change shape as time progresses [77]. For this reason, time-dependent active contours and deformable models are preferred for studies of cell migration, for example.

3.3.5 Classification and Clustering

The advent of modern, automated, digital microscopes, together with target specific fluorescent probes has enabled the collection of large amounts of image data whose impact can be vastly augmented through the use of high-throughput image screening and analysis methods. The image processing tasks described previously (restoration, registration, segmentation, tracking, etc.) can be combined with traditional machine learning methodology to deliver powerful tools to aid discovery and validation for life sciences applications [79].

A prominent example is the application of image-based clustering and classification methods to the problem of subcellular localization of proteins within cellular compartments. The development of advanced protein tagging methods, with the aid of biological image database systems [80, 81] and advanced feature extraction, classification, and clustering methods have enabled scientists to address the problem of analyzing subcellular location patterns on a proteome-wide basis, providing valuable information on the molecular mechanisms that dictate cell structure and function [15, 65]. Introduction of more advanced tools such as MR, has been proposed in [15] for subcellular analysis, as well as in [82] for detection of developmental stages *Drosophila* embryos.

Yet another common use of classification methods in bioimaging is in cell cycle analysis and determination. Common applications include studying of the effects of gene suppression [83] as well as drug therapies [84]. The steps used to implement an image processing system capable of performing automated classification and analysis of cell cycle response are normally the same as for other applications and include, but are not limited to, image restoration, registration to remove translation and rotation dependency in numerical feature extraction, as well as training of the classifier.

3.3.6 Modeling

Computational imaging methods also have a place in modeling and simulation of biological phenomena at the cellular and subcellular scales. Quantitative information automatically extracted from images can be used for model selection, calculating model parameters, as well as for validating different models. We describe two applications of image-based modeling in cells and subcellular structures: computation of material parameters describing force distribution in cells, as well as modeling the dynamical properties of microtubules.

The precise determination of the mechanical properties of cells, under different environments, can be used to gain a better understanding of a variety of biological phenomena. Finite-element models derived based on constitutive laws can be used to estimate stress-strain relationships, as well as other physical parameters, based on boundary conditions extracted from image data. Modeling and simulation of uniaxial cell stretching experiments were performed in [85] where cell boundaries were extracted (segmented) as a function of time from image of cells undergoing stretching.

Modeling has also been performed in an effort to understand the motion of subcellular structures with the aid of time-lapse microscopy [86]. One of the simplest and most used models for describing particle random motion is the auto regressive moving average (ARMA) model, where the goal is to describe the value of an observed variable as a linear combination of past values of that variable as well as past values of a white noise random variable [87]. In [86] localization information obtained from automated tracking algorithms [88] were used to estimate the parameters of an ARMA model for studying kinetochore microtubule dynamics in yeast. Another new area for modeling is that of intelligent acquisition, discussed at the beginning of this section, where data set models, as opposed to the data itself, are learned and acquired [34].

Bibliography

[1] J. Prewitt and M. Mendelsohn, "The analysis of cell images," *Ann. N Y Acad. Sci.*, vol. 128, no. 3, pp. 1035–1053, 1966.

[2] R. Eils and C. Athale, "Computational imaging in cell biology," *Journ. Cell Biol.*, vol. 161, no. 3, pp. 477–81, 2003.

[3] Y. Wang, K. Hahn, R. Murphy, and A. Horwitz, "From imaging to understanding: Frontiers in live cell imaging," *Journ. Cell Biol.*, vol. 174, no. 4, pp. 481–484, 2006.

[4] "IEEE Signal Proc. Mag., sp. iss. Molec. Cellular Bioimaging," May 2006.

[5] H. Barrett and K. Myers, *Foundations of Image Science*. Hoboken, NJ: John Wiley & Sons, 2004.

[6] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*. Signal Processing, Englewood Cliffs, NJ: Prentice Hall, 1995.

[7] M. PÜSchel and J. Moura, "Algebraic signal processing theory: Foundation and 1-D time." Submitted.

[8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning; Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.

[9] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, vol. I. Englewood Cliffs, NJ: Prentice Hall, 1993.

[10] S. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*, vol. II. Englewood Cliffs, NJ: Prentice Hall, 1993.

[11] J. Kovačević and A. Chebira, "Life beyond bases: The advent of frames (Part I)," *IEEE Signal Proc. Mag.*, vol. 24, pp. 86–104, Jul. 2007.

[12] J. Kovačević and A. Chebira, "Life beyond bases: The advent of frames (Part II)," *IEEE Signal Proc. Mag.*, vol. 24, Sep. 2007.

[13] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[14] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. Pure and Appl. Math.*, vol. 41, pp. 909–996, Nov. 1988.

[15] A. Chebira, Y. Barbotin, C. Jackson, T. Merryman, G. Srinivasa, R. Murphy, and J. Kovačević, "A multiresolution approach to automated classification of protein subcellular location images," *BMC Bioinformatics*, vol. 8, no. 210, 2007.

[16] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neur. Comp.*, vol. 13, pp. 1373–1396, 2003.

[17] D. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," in *Natl. Acad. Sci.*, vol. 100, pp. 5591–5596, 2003.

[18] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Englewood Cliffs, NJ: John Wiley & Sons, 2001.

[19] P. Comon, “Independent component analysis, A new concept ?,” *Signal Proc.*, vol. 36, pp. 287–314, 1994.

[20] A. Hyvärinen, “Survey on independent component analysis,” *Neural Computing Surveys*, vol. 2, pp. 94–128, 1999.

[21] S. Kullback, *Information Theory and Statistics*. Dover Publications, 1968.

[22] A. Bell and T. Sejnowski, “The ‘independent components’ of natural scenes are edge filters,” *Vis. Res.*, vol. 37, no. 23, pp. 3327–3338, 1997.

[23] A. Hyvärinen, “ICA Web Site.” `http://www.cs.helsinki.fi/u/ahyvarin/whatisica.shtml`.

[24] L. Saul, K. Weinberger, J. Ham, F. Sha, and D. Lee, *Semisupervised Learning*, ch. Spectral Methods for Dimensionality Reduction. MIT Press, 2006.

[25] J. Ham, D. Lee, S. Mika, and B. Schölkopf, “A kernel view of the dimensionality reduction of manifolds,” tech. rep., Max Planck Institute for Biological Cybernectics, 2003.

[26] R. Vidal, Y. Ma, and S. Sastry, “Generalized principal component analysis (GPCA),” *IEEE Trans. Patt. Anal. and Mach. Intelligence*, vol. 27, no. 12, pp. 1–15, 2005.

[27] J. Tenenbaum, V. de Silva, and J. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, pp. 2319–2323, 2000.

[28] T. Cox and M. Cox, *Multidimensional Scaling*. Chapman and Hall/CRC, 2001.

[29] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, pp. 2323–2326, 2000.

[30] S. Inoué, *Handbook of Biological Confocal Microscopy*, ch. Foundations of Confocal Scanned Imaging in Light Microscopy. Springer-Verlag, 2005.

[31] K. König, *Handbook of Biological Confocal Microscopy*, ch. Cell Damage During Multi-Photon Microscopy. Springer-Verlag, 2005.

[32] A. Diaspro, G. Chirico, C. Usai, P. Ramoino, and J. Dobrucki, *Handbook of Biological Confocal Microscopy*, ch. Photobleaching. Springer-Verlag, 2005.

[33] T. E. Merryman and J. Kovačević, “Adaptive multiresolution acquisition of fluorescence microscopy data sets,” *IEEE Trans. Image Proc., sp. iss. Molecular and Cellular Bioimaging*, vol. 14, pp. 1246–1253, Sep. 2005.

[34] C. Jackson, R. Murphy, and J. Kovačević, “Efficient acquisition and learning of fluorescence microscopy data models,” in *Proc. IEEE Int. Conf. Image Proc.*, (San Antonio, TX), Sep. 2007.

[35] A. Genovesio, T. Liedl, V. Emiliani, W. Parak, M. Coppey-Moisan, and J. Olivo-Marin, “Multiple particle tracking in 3-D+t microscopy: Method and application to the tracking of endocytosed quantum dots,” *IEEE Trans. Image Proc.*, vol. 15, pp. 1062–1070, May 2006.

[36] P. Sarder and A. Nehorai, “Deconvolution methods for 3-D fluorescence microscopy images,” *IEEE Signal Proc. Mag., sp. iss. Molec. Cellular Bioimaging*, vol. 23, pp. 32–45, May 2006.

[37] P. Verveer, M. Gemkow, and T. Jovin, “A comparison of image restoration approaches applied to three-dimensional confocal and wide-field florescence microscopy,” *Journ. Microscopy*, vol. 193, no. 1, pp. 50–61, 1999.

[38] J. Markham and J. Conchello, “Fast maximum-likelihood image-restoration algorithms for three-dimensional fluorescence microscopy,” *Journ. Opt. Soc. Am. A*, vol. 18, no. 5, pp. 1062–1071, 2001.

[39] K. Castleman, *Digital Image Processing*, ch. Image Restoration. Prentice Hall, 1996.

[40] A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems*. John Wiley & Sons, 1977.

[41] N. Dey, L. Blanc-Feraud, C. Zimmer, Z. Kam, J. Olivo-Marin, and J. Zerubia, “A deconvolution method for confocal microscopy with total variation regularization,” in *Proc. IEEE Int. Symp. Biomed. Imaging*, (Arlington, VA), pp. 1223–1226, Apr. 2004.

[42] M. Figueiredo and R. Nowak, “An EM algorithm for wavelet-based image restoration,” *IEEE Trans. Image Proc.*, vol. 12, no. 8, pp. 906–916, 2003.

[43] J. Skilling, *Classic maximum entropy*, ch. Maximum Entropy and Bayesian Methods. Kluwer Academic Publishers, 1989.

[44] S. Joshi and M. Miller, “Maximum a posteriori estimation with Good’s roughness for three-dimensional optical-sectioning microscopy,” *Journ. Opt. Soc. Am. A*, vol. 10, no. 5, pp. 1078–1085, 1993.

[45] D. Fish, A. Brinicombe, E. Pike, and J. Walker, “Blind deconvolution by means of the Richardson-Lucy algorithm,” *Journ. Opt. Soc. Am. A*, vol. 12, no. 1, pp. 58–65, 1995.

[46] C. Vonesch and M. Unser, “Fast wavelet-regularized image deconvolution,” in *Proc. IEEE Int. Symp. Biomed. Imaging*, (Arlington, VA), pp. 608–611, Apr. 2007.

[47] P. Schönemann, “A generalized solution of the orthogonal procrustes problem,” *Psychometrika*, vol. 31, pp. 1–10, 1966.

[48] J. Hajnal, D. Hill, and D. Hawkes, *Medical Image Registration*. New York, NY: CRC Press, 2001.

[49] F. Bookstein, “Principal warps: Thin-plate splines and the decomposition of transformations,” *IEEE Trans. Patt. Anal. and Mach. Intelligence*, vol. 11, pp. 567–585, 1989.

[50] J. Maintz and M. Viergever, “A survey of medical image registration,” *Med. Image Anal.*, vol. 2, pp. 1–36, 1998.

[51] B. Zitova and J. Flusser, “Image registration methods: A survey,” *Image and Vis. Comp.*, vol. 21, pp. 977–1000, 2003.

[52] J. Pluim, J. Mainz, and M. Vergever, “Mutual-information-based registration of medical images: A survey,” *IEEE Trans. Med. Imag.*, vol. 22, pp. 986–1004, 2003.

[53] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge Univ. Press, 2nd ed., 1992.

[54] C. Sorzano, P. Thévenaz, and M. Unser, “Elastic registration of biological images using vector-spline regularization,” *IEEE Trans. Biomed. Eng.*, vol. 52, no. 4, pp. 652–663, 2005.

[55] C. Vonesch, F. Aguet, J. Vonesch, and M. Unser, “The colored revolution of bioimaging,” *IEEE Signal Proc. Mag., sp. iss. Molec. Cellular Bioimaging*, vol. 23, pp. 20–31, May 2006.

[56] T. Vercauteren, A. Perchant, G. Maladain, X. Pennec, and N. Ayache, “Robust mosaicing with correction of motion distortions and tissue deformations for in vivo fibered microscopy,” *Med. Image Anal.*, vol. 10, pp. 673–692, 2006.

[57] K. Rohr, P. Cathier, and S. Worz, “Elastic registration of electrophoresis images using intensity information and point landmarks,” *Pattern Recogn.*, vol. 37, no. 5, pp. 1035–1048, 2004.

[58] A. Olson, N. Larson, and C. Heckman, “Classification of cultured mammalian cells by shape analysis and pattern recognition,” in *Natl. Acad. Sci.*, vol. 77, pp. 1516–1520, 1980.

[59] A. Gray, D. Young, N. Martin, and C. Glasbey, “Cell identification and sizing using digital image analysis for estimation of cell biomass in high rate algal ponds,” *Journ. Appl. Psychology*, vol. 14, pp. 193–204, 2002.

[60] D. Mukherjee, N. Ray, and S. Acton, “Level set analysis for leukocyte detection and tracking,” *IEEE Trans. Image Proc.*, vol. 13, pp. 562–572, 2004.

[61] E. Meijering, M. Jacob, J. Sarria, P. Steiner, H. Hirling, and M. Unser, “Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images,” *Cytometry*, vol. 58A, pp. 167–176, Apr. 2004.

[62] C. Zimmer, B. Zhang, A. Dufour, A. Thebaud, S. Berlemont, V. Meas-Yedid, and J. O. Marin, “On the digital trail of mobile cells,” *IEEE Signal Proc. Mag., sp. iss. Molec. Cellular Bioimaging*, vol. 23, pp. 54–62, May 2006.

[63] S. Chen, T. Zhao, G. Gordon, and R. Murphy, “A novel graphical model approach to segmenting cell images,” in *Proc. IEEE Symp. Comp. Intelligence Bioinform. Comp. Biol.*, 2006.

[64] G. Srinivasa, M. Fickus, and J. Kovačević, “Multiscale active contour transformations for the segmentation of fluorescence microscope images,” in *Proc. SPIE Conf. Wavelet Appl. in Signal and Image Proc.*, (San Diego, CA), Aug. 2007.

[65] E. Glory and R. Murphy, “Automated subcellular location determination and high throughput microscopy,” *Developmental Cell*, vol. 12, pp. 7–16, 2007.

[66] M. Velliste and R. Murphy, “Automated determination of protein subcellular locations from 3D fluorescence microscope images,” in *Proc. IEEE Int. Symp. Biomed. Imaging*, (Washnigton, DC), pp. 867–870, 2002.

[67] E. Bengtsson, C. Wählby, and J. Lindblad, “Robust cell image segmentation methods,” *Pattern Recogn. and Image Anal.*, vol. 14, no. 2, pp. 157–167, 2004.

[68] C. Xu, D. Pham, and J. Prince, “Medical image segmentation using deformable models,” in *Handbook of Medical Imaging* (J. Fitzpatric and M. Sonka, eds.), SPIE Press, 2000.

[69] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *Int. Journ. Comp. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.

[70] C. Pluempitiwiriyawej, J. Moura, Y. Wu, and C. Ho, “STACS: A new active contour scheme for cardiac MR image segmentation,” *IEEE Trans. Med. Imag.*, vol. 24, pp. 593–603, May 2005.

[71] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2003.

[72] T. Chan and L. Vese, “Active contours without edges,” *IEEE Trans. Image Proc.*, vol. 10, pp. 266–277, Feb. 2001.

[73] L. Coulot, H. Kischner, A. Chebira, J. Moura, J. Kovačević, E. Osuna, and R. Murphy, “Topology preserving STACS segmentation of protein subcellular location images,” in *Proc. IEEE Int. Symp. Biomed. Imaging*, (Arlington, VA), pp. 566–569, Apr. 2006.

[74] A. Cohen, B. Roysam, and J. Turner, “Automated tracing and volume measurements of neurons from 3-D confocal fluorescence microscopy data,” *Journ. Microscopy*, vol. 173, no. 2, pp. 103–114, 1994.

[75] K. Al-Kofahi, S. Lasek, D. Szarowski, C. Page, G. Nagy, J. Turner, and B. Roysam, “Rapid automated three-dimensional tracing of neurons from confocal image stacks,” *IEEE Trans. Inform. Tech. in Biomed.*, vol. 6, pp. 171–187, Jun. 2006.

[76] D. Sage, F. Neumann, F. Hediger, S. Gasser, and M. Unser, “Automatic tracking of individual fluorescence particles: Application to the study of chromosome dynamics,” *IEEE Trans. Image Proc.*, vol. 14, no. 3, pp. 1372–1383, 2005.

[77] E. Meijering, I. Smal, and G. Danuser, “Tracking in molecular imaging,” *IEEE Signal Proc. Mag., sp. iss. Molec. Cellular Bioimaging*, vol. 23, pp. 46–53, May 2006.

[78] M. Cheezum, W. Walker, and W. Guilford, “Quantitative comparison of algorithms for tracking single fluorescent particles,” *Biophys. Journ.*, vol. 81, no. 4, pp. 2378–2388, 2001.

[79] X. Zhou and S. Wong, “Informatics challenges of high-throughput microscopy,” *IEEE Signal Proc. Mag., sp. iss. Molec. Cellular Bioimaging*, vol. 23, no. 3, pp. 64–71, 2006.

[80] K. Huang and R. Murphy, “From quantitative microscopy to automated image understanding,” *Journ. Biomed. Optics*, vol. 9, pp. 893–912, 2004.

[81] I. Goldberg, C. Alland, J. Burel, D. Creager, A. Falconi, H. Hochheiser, J. Johnston, J. Mellen, P. Sorger, and J. Swedlow, “The open microscopy environment (OME) data model and XML file: Open tools for informatics and quantitative analysis in biological imaging,” *Genome Biol.*, vol. 6, no. 5, p. R47, 2005.

[82] R. Kellogg, A. Chebira, A. Goyal, P. Cuadra, S. Zappe, J. Minden, and J. Kovačević, “Towards an image analysis toolbox for high-throughput Drosophila embryo RNAi screens,” in *Proc. IEEE Int. Symp. Biomed. Imaging*, (Arlington, VA), pp. 288–291, Apr. 2007.

[83] N. Harder, B. Neumann, M. Held, U. Liebel, H. Erfle, J. Ellenberg, R. Eils, and K. Rohr, “Automated recognition of mitotic patterns in fluorescence microscopy images of human cells,” in *Proc. IEEE Int. Symp. Biomed. Imaging*, (Arlington, VA), pp. 1016–1019, Apr. 2006.

[84] X. Chen, X. Zhou, and S. Wong, “Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy,” *IEEE Trans. Biomed. Eng.*, vol. 53, no. 4, pp. 762–766, 2006.

[85] E. Gladilin, A. Micoulet, B. Hosseini, K. Rohr, J. Spatz, and R. Eils, “3D finite element analysis of uniaxial cell stretching: from image to insight,” *Phys. Biol.*, vol. 4, pp. 104–113, Jun. 2007.

[86] K. Jaqaman, J. Dorn, G. Jelson, J. Tytell, P. K. Sorger, and G. Danuser, “Comparative autoregressive moving average analysis of kinetochore microtubule dynamics in yeast,” *Biophys. Journ.*, vol. 91, pp. 2312–2325, 2006.

[87] M. Priestley, *Spectral Analysis and Time Series*, ch. Stationary Random Processes. Academic Press, 1981.

[88] J. Dorn, K. Jaqaman, D. Rines, G. Jelson, P. Sorger, and G. Danuser, “Yeast kinetochore microtubule dynamics analyzed by high-resolution three-dimensional microscopy,” *Biophys. Journ.*, vol. 89, pp. 2835–2854, 2005.