Contrast invariant image analysis and PDE's

Frédéric Guichard   Jean-Michel Morel   Robert Ryan

Contents

Introduction

This book addresses the problem of low-level image analysis and, as such, is a contribution to image processing and imaging science. While the material touches on several aspects of image analysis—and peripherally on other parts of image processing—the main subject is image smoothing using partial differential equations (PDEs). The rational for a book devoted to smoothing is the assumption that a digital image must be smoothed before reliable features can be extracted.

The purpose of this introduction is to establish some of the language, conventions, and assumptions that are used throughout the book, to review part of the history of PDEs in image processing, and to introduce notation and background material.

I.1 Images

Since the objects of our study are ultimately digital images, we begin by defining what we mean by "digital image" and by describing some of the ways these images are obtained and some current assumptions about the "original images" from which the digital images are derived.

Most of the images dealt with will be natural images, that is, images from nature (people, landscapes, cityscapes, etc.). We include medical images and astronomical images, and we do not exclude drawings, paintings, and other man-made images. All of the images we consider will be grayscale images. Thus, mathematically, an image is a real-valued function u defined on some subset Ω of the plane $\mathbb{R}^2$. The value $u(\boldsymbol{x})$, $\boldsymbol{x} = (x, y) \in \Omega$, represents the gray level of the image at the point $\boldsymbol{x}$. If u is a digital image, then its domain of definition is a finite grid with evenly spaced points. It is often square with $2^n \times 2^n$ points. The gray levels $u(\boldsymbol{x})$ are typically coded with the integers 0–255, where 0 represents black and 255 represents white. If h is the distance between grid lines, then the squares with sides of length h centered at the points $u(\boldsymbol{x})$ are called *pixels*, where "pix" is slang for "picture" and "el" stands for "element."

The mathematical development in this book proceeds along two parallel lines. The first is theoretical and deals with images u that belong to function spaces, generally spaces of continuous functions that are defined on domains of $\mathbb{R}^2$. The second line concerns numerical algorithms, and for this the images are digital images. To understand the relations between the digital and continuous images, it is useful to consider some examples of how images are obtained and some of the assumptions we make about the processes and the images. Perhaps the simplest example is that of taking a picture of a natural scene with a digital camera. The scene—call it S—is focused at the focal plane of the camera forming a representation of S that we denote by u_f. When we take the picture, the image u_f is sampled, or captured, by an array of charged coupled devices (CCDs) producing the digital image u_d. This image, u_d, is the only representation of S that is directly available to us; the image u_f is not directly available to us. Even more elusive is the completely hypothetical image that we call u_S. This is the representation of S that would be formed at the focal plane of an ideal camera having perfect optics. A variation on this example is to capture u_f on film as the image u_p. Then u_p can be sampled (scanned) to produce a digital image u_d. For example, before the advent of CCDs, astronomical images were captured on Schmidt plates. Many of these plates have been scanned recently, and the digital images have been made available to astronomers via the Internet.

Aspects of the photographic example could be recast for medical imaging. Although photography plays an important role in medicine, images for diagnostic use are often obtained using other kinds of radiation. X-rays are perhaps closest to our photographic example. In this case, there is an image corresponding to u_p that can be scanned to produce a digital image u_d. Other medical imaging processes, such as scintigraphy and nuclear magnetic resonance, are more complicated, but these processes yield digital images. The images examined by the experts are often "negatives" produced from an original digital images. Irrespective of the process, digital images captured by some technology all have one characteristic in common: They are all noisy.

One way to relate the different representations of S, is to write

$$u_d = Tu_S + n,$$

where T is a hypothetical operator representing some technology and n is noise. In the case of photography, we might write this in two steps,

$$\begin{cases} u_f = P * u_S + n_1, \\ u_d = Ru_f + n_2, \end{cases}$$

where P represents the optics and R represents the sampling. This is a useful model in optical astronomy, since astronomers have considerable knowledge about the operators P and R and about the noises n_1 and n_2. Similarly, experts in other technologies know a great deal about the processes and noise sources. Noise and pixels are illustrated in Figure I.1

In the photographic example, the image u_f is a smoothed version of u_S. Furthermore, $Ru_f(\boldsymbol{x})$ is not exactly $u_f(\boldsymbol{x})$ but rather an average of values of u_f in a small neighborhood of $\boldsymbol{x}$, which is to say that the operator R does some smoothing. Thus, in this example, u_d is sampled from a smoothed version of S. We are going to assume that this is the case for the digital images considered in the book, except for digital images that are artificially generated. This is realistic, since all of the processes T that we can imagine for capturing images, smooth the original photon flux. In fact, this is more of an observation about technology than it is an assumption. We are also going to assume that, for any technology considered, the sampling rate used to produce u_d is high enough so that u_d is a "good" representation of the smoothed version of S, call it u_f, from which it was derived. Here, "good" means that the parallel development in the book mentioned above make sense; it means that, from a practical point of view, the theoretical development that uses smooth functions to model the images u_f is indeed related to the algorithmic development that uses the digital images u_d. We will say more about smoothing and sampling in section I.2.

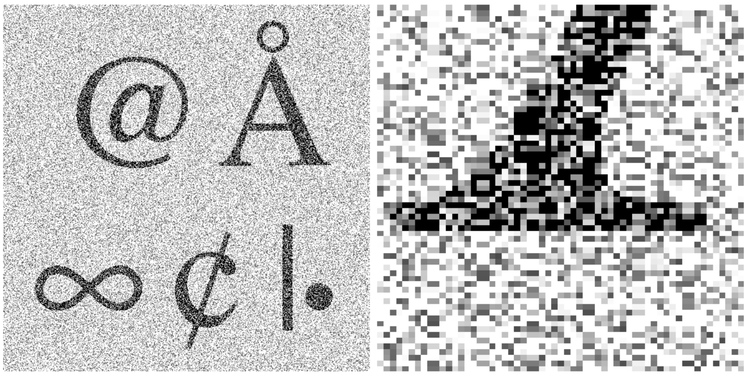

Figure I.1: A noisy image magnified to show the pixels.

It is widely assumed that the underlying "real image" u_S is either a measure or, for more optimistic authors, a function that has strong discontinuities. Rudin in 1987 [214] and De Giorgi and Ambrosio in 1988 [92] proposed independently the space $BV(\mathbb{R}^2)$ of functions with bounded variation as the correct function space for modeling the images u_S. A function f is in $BV(\mathbb{R}^2)$ if its partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$, taken as distributions, are Radon measures with finite total mass. $BV(\mathbb{R}^2)$ looked at first well adapted to modeling digital images because it contains functions having step discontinuities. In fact, the characteristic functions of smooth domains in $\mathbb{R}^2$ belong to $BV(\mathbb{R}^2)$. However, in 1999, Alvarez, Gousseau, and Morel used a statistical device on digital images u_d to estimate how the corresponding images u_S oscillate [6]. They deduced by geometric-measure arguments, that the u_S have, in fact, unbounded variation. We may therefore accept the idea that these high-resolution images contain very strong oscillations. Although the images u_f are smoothed versions of the u_S, and hence the oscillations have been averaged, common sense tells us that they also have large derivatives at transitions between different observed objects, that is, on the apparent contours of physical objects. Furthermore, we expect that these large derivatives (along with noise) are passed to the digital images u_d.

I.2 Image processing

For the convenience of exposition, we divide image processing into separate disciplines. These are distinguished not so much by their techniques, which often overlap, as they are by their goals. We will briefly describe two of these areas: compression and restoration. The third area, image analysis, is the main subject of the book and will be discussed in more detail.

Image compression

Compression is based on the discrete nature of digital images, and it is motivated by economic necessity: Each form of storage and transmission has an associated cost, and hence one wishes to represent an image with the least number of bits that is compatible with end usage. There are two kinds of compression: lossless compression and lossy compression. Lossless compression algorithms are used to compress digital files where the decompressed file must agree bit-by-bit with the original file. Perhaps the best known example of lossless compression is the zip format. Lossless algorithms can be used on any digital file, including digital images. These algorithms take advantage of the structure of the file itself and have nothing to do with what the file represents. On the other hand, lossy compression algorithms take advantage of redundancies in natural images and subtleties of the human visual system. Done correctly, one can throw away information contained in an image without impairing its usefulness. The goal is to develop algorithms that provide high compression factors without objectionable visible alterations. Naturally, what is visually objectionable depends on how the decompressed image is used. This is nicely illustrated with our photographic example. Suppose that we capture the image u_f at our camera's highest resolution. If we are going to send u_d over the Internet to a publisher to be printed in a high-quality publication, then we want no loss of information and will probably send the entire file in the zip format. If, however, we just want the publisher to have a quick look at the image, then we would probably send u_d compressed as a .jpg file, using the Joint Photographic Expert Group (JPEG) standard for still image compression. This kind of compression is illustrated in Figure I.2.

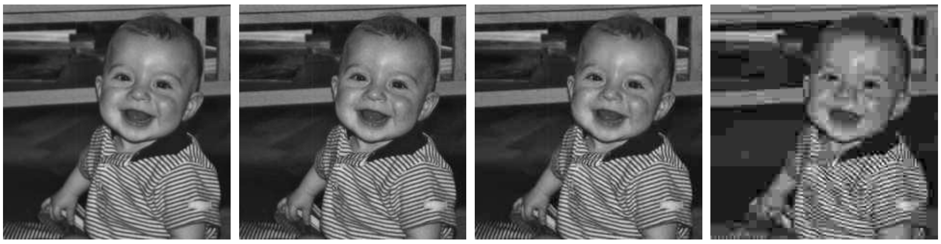

Figure I.2: Compression. Left to right: the original image and its increasingly compressed versions. The compression factors are roughly 7, 10, and 25. Up too a 10 factor, alterations are hardly visible.

Image restoration

A second area is restoration or denoising. Restoring digital images is much like restoring dirty or damaged paintings or photographs. Beginning with a digital image that contains blurs or other perturbations (all of which may be considered as noise), one wishes to produce a better version of the image; one wishes to enhance aspects of the image that have been attenuated or degraded. Image restoration plays an important role in law enforcement and legal proceedings. For example, surveillance cameras generally produce rather poor images that must often be denoised and enhanced as needed. Image restoration is also important in science. When the Hubble Space Telescope was first launched in 1990, and until it was repaired in 1993, the images it returned were all blurred due to a spherical aberration in the telescope's primary mirror. Elaborate (and costly) algorithms were developed to restore these poor images, and indeed useful images were obtained during this period. Restoration is illustrated in Figure I.3 with an artificial example. The image on the left has been ostensibly destroyed by introducing random-valued pixels amounting to 75% of the total pixel count. Nevertheless, the image can be significantly restored, and a restored version is shown on the right, by using a Vincent and Serra operator which we will study in Chapter **??**, the "area opening".

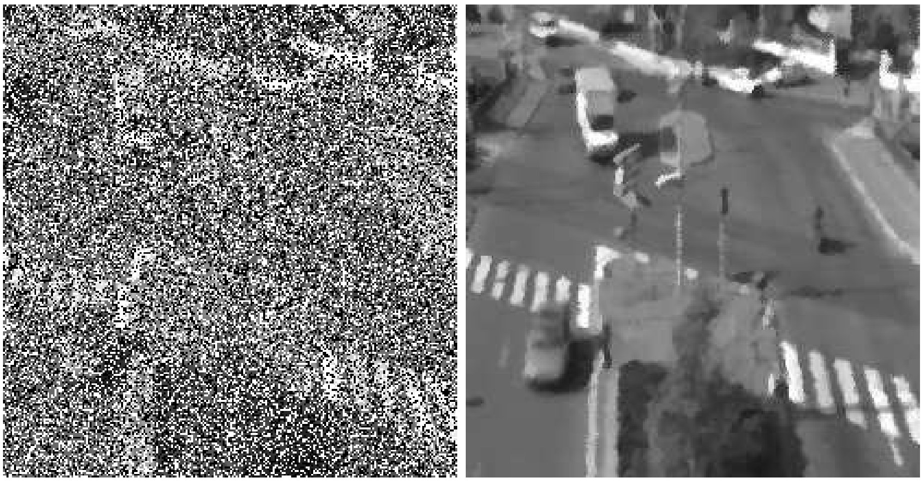

Figure I.3: Denoising. Left: an image with up to 75% of its pixels contaminated by simulated noise. Right: a denoised version by the Vincent-Serra algorithm (area opening).

Image analysis

A third area of image processing is low-level image analysis, and since this is the main topic of the book, it is important to explain what we mean by "low-level" and "analysis." "Analysis" is widely used in mathematics, with various shades of meaning. Our use of "analyze," and thus of "analysis," is very close to its common meaning, which is to decompose a whole into its constituent parts, to study the parts, and to study their relation to the whole. For our purposes, the constituent parts are, for the most part, the "edges" and "shapes" in an image. These objects, which are often called features, are things that we could, for a given image, point to and outline, although for a complex natural image this would be a tedious process. The goal of image analysis is to create algorithms that do this automatically.

The term "low-level" comes from the study of human vision and means extracting reliable, local geometric information from an image. At the same time, we would like the information to be minimal but rich enough to characterize the image. The goal here is not compression, although some of the techniques may provide a compressed representation of the image. Our goal is rather to answer questions like, Does a feature extracted from image A exist in image B? We are also interested in comparing features extracted from an image with features stored in a database. As an example, consider the level set at the left of Figure I.4. It consists of major features (roughly, the seven appendages) and noise. The noise, which is highly variable, prevents us from comparing the image directly with other images having similar shapes. Thus we ask for a sketchy version, where, however, all essential features are kept. The images on the right are such a sketchy versions, where most of the spurious details (or noise) have disappeared, but the main structures are maintained. These sketchy versions may lead to concise invariant encoding of the shape. Notice how the number of inflexion points of the shape has decreased in the simplification process. This is an example of what we mean by image analysis. The aim is not denoising or compression. The aim is to construct an invariant code that puts in evidence the "main parts" of an image (in this case, the appendages) and that facilitates fast recognition in a large database of shapes.

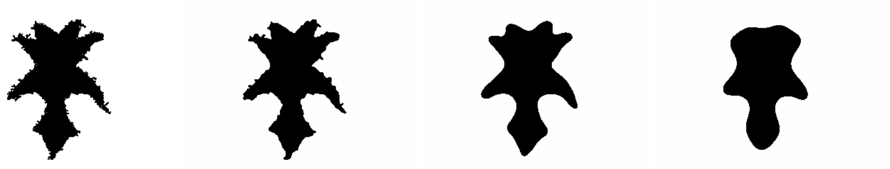

Figure I.4: Analysis of a shape. The original scanned shape is on the left. Simplified versions are to the right.

Edge detection and scale space

Since the earliest work in the 1960s, one of the goals of image analysis has been to locate the strong discontinuities in an image. This search is called *edge detection*, and it derives from early research that involved working with images of cubes. This seemingly simple goal turned out to be exceedingly difficult. Here is what David Marr wrote about the problem in the early 1980s ([163], p. 16):

> The first great revelation was that the problems are difficult. Of course, these days this fact is a commonplace. But in the 1960s almost no one realized that machine vision was difficult. The field had to go through the same experience as the machine translation field did in its fiascoes of the 1950s before it was at last realized that here were some problems that had to be taken seriously. The reason for this misconception is that we humans are ourselves so good at vision. The notion of a feature detector was well established by Barlow and by Hubel and Wiesel, and the idea that extracting edges and lines from images might be at all difficult simply did not occur to those who had not tried to do it. It turned out to be an elusive problem: Edges that are of critical importance from a three-dimensional point of view often cannot be found at all by looking at the intensity changes in an image. Any kind of textured image gives a multitude of noisy edge segments; variations in reflectance

> and illumination cause no end of trouble; and even if an edge has a clear existence at one point, it is as likely as not to fade out quite soon, appearing only in patches along its length in the image. The common and almost despairing feeling of the early investigators like B.K.P. Horn and T.O. Binford was that practically anything could happen in an image and furthermore that practically everything did.

The point we wish to emphasize is that textures and noise (which are often lumped together in image analysis) produce unwanted edges. The challenge was to separate the "true edges" from the noise. For example, one did not want to extract all of the small edges in a textured wall paper; one wanted the outline of the wall. The response was to blur out the textures and noise in a way that left the "true edges" intact, and then to extract these features. More formally, image analysis was reformulated as two processes: smoothing followed by edge detection. At the same time, a new doctrine, the *scale space*, was proposed. Scale space means that instead of speaking of features of an image at a given location, we speak of them at a given location and at a given scale, where the scale quantifies the amount of smoothing performed on the image before computing the features. We will see in experiments that "edges at scale 4" and "edges at scale 7" are different outputs of an edge detector.

Three requirements for image smoothing operators

We have advertised that this book is about image analysis, which we have just defined to be smoothing followed by edge detection, or feature extraction. In fact, the text focuses on smoothing and particularly on discussing and answering the question, What kind of smoothing should be used? To approach this problem, we need to introduce three concepts associated with image analysis operators. These concepts will be used to narrow the field of smoothing operators. We introduce them informally at first; more precise meanings will follow.

Localization

The first notion is *localization.* Roughly speaking, to say that an operator T is localized means it essentially uses information from a small neighborhood of $\boldsymbol{x}$ to compute the output $Tu(\boldsymbol{x})$. Recall that the sampling operator R in the photographic example was well localized. As another example, consider the classic Gaussian smoothing operators $\mathcal{G}_t$ defined by

$$\mathcal{G}_t u(\boldsymbol{x}) = G_t * u(\boldsymbol{x}) = \int_{\mathbb{R}^2} G_t(\boldsymbol{y}) u(\boldsymbol{x}-\boldsymbol{y})\,\mathrm{d}\boldsymbol{y},$$

where $G_t(\boldsymbol{x}) = (1/4\pi t)\mathrm{e}^{-|\boldsymbol{x}|^2/4t}$. If $t > 0$ is small, then the Gaussian G_t is well localized around zero and $\mathcal{G}_t u(\boldsymbol{x})$ is essentially an average of the values of $u(\boldsymbol{x})$ in a small neighborhood of $\boldsymbol{x}$. The importance of localization is related to the occlusion problem: Most optical images consist of a superposition of different objects that partially obscure one another. It is clear that we must avoid confusing them in the analysis, as would, for example, G_t if t is large. It is for reasons like this that we want the analysis to be as local as possible.

We will prove in Chapter 1 under rather general conditions that $u(t, \boldsymbol{x}) = G_t * u_0(\boldsymbol{x})$ is the unique solution of the heat equation

$$\frac{\partial u}{\partial t} = \Delta u$$

with initial value u_0. Thus, we can say that smoothing u_0 with the Gaussian G_t is equivalent to applying the heat equation to u_0. We will see that the heat equation is possibly the worst candidate in our search for the ideal smoothing operator, since, except for small t, it is poorly localized and produces a very blurred image.

Iteration

One might conjecture that a way around this problem with the heat equation would be to replace G_t with a more suitable positive kernel. This is not the case, but it does serve to introduce the second concept, which is *iteration.* We will show in Chapter 2 that under reasonable assumptions and appropriate rescalings, iterating a convolution with a positive kernel leads to the Gaussian, and thus directly back to the heat equation. There is, however, a different point of view that leads to useful smoothing operators: Instead of looking for a different kernel, look for other PDEs that provide smoothing. This program leads to a class of nonlinear PDEs, where the Laplacian in the heat equation is replaced by various nonlinear operators. We will see that for these operators it is generally better, from the localization point of view, to iterate a well localized operator than to apply it directly at a large scale. This, of course, is just not true for the heat equation; if you iterate n times the convolution $G_t * u$ you get exactly $G_{nt} * u$. This is a good place to point out that if we are dealing with smoothing, localization, and iteration, then we are talking about parabolic PDEs. This announcement is heuristic, and the object of the book is to formalize and to make precise the necessity and the role of several PDEs in image analysis.

Invariance

Our last concept is *invariance.* Invariance requirements play a central role in image analysis because the objects to be recognized must be recognized under varying conditions of illumination (contrast invariance) and from different points of view (projective invariance). Contrast invariance is one of the central requirements of the theory of image analysis called mathematical morphology (see, for example, Matheron [169] or Serra [228]). This theory involves a number of contrast-invariant image analysis operators, including dilations, erosions, median filters, openings, and closings. We are going to use this theory by attempting to localize as much as possible these *morphomath* operators to exploit their behavior at small scales. We will then iterate these operators. This will lead to the proof that several geometric PDEs, namely, the curvature motions, are asymptotically related to certain morphomath operators in much the same way that linear smoothing is related to the heat equation. Thus, through these PDEs, one is able to combine the scale space doctrine and mathematical morphology. In particular, affine-invariant morphomath operators, which seemed at first to be computationally impractical, turn out to yield in their local iterated

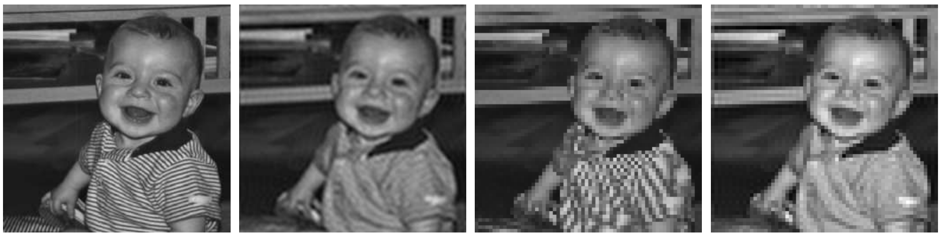

Figure I.5: Shannon theory and sampling. Left to right: original image; smoothed image; sampled version of the original image; sampled version of the smoothed image. This illustrates the famous Shannon-Nyquist law that an image must be smoothed before sampling in order to avoid the aliasing artifacts.

version a very affordable PDE, the so called *affine morphological scale space* (AMSS) equation.

Shannon's sampling theory

We mentioned in section I.1 that most of the digital images u_d that come to us in practice have been sampled from a smoothed version, call it u_f, of the "real image" u_S. This was basically a comment about the technology. Another comment (or assumption) was that the sampling rate was high enough to capture all of the information in u_f that is needed in practice. What we mean by this is that the representations of u_f that we reconstruct from u_d show no signs that u_f was undersampled. This is an empirical statement; we will comment on the theory in a moment, but first we wish to illustrate in Figure I.5 what can happen if an image is undersampled.

We call the original image on the left Victor. Notice that Victor's sweater contains a striped pattern, which has a spatial frequency that is high relative to other aspects of the picture. If we attempt to reduce the size of Victor by simply sampling, for example, by taking one pixel in sixteen in a square pattern, we obtain a new image (the third panel) in which the sampling has created new and unstable patterns. Notice how new stripes have been created with a frequency and direction that has nothing to do with the original. This is called *aliasing*, and it is caused by high spatial frequencies being projected onto lower frequencies, which creates new patterns. If this had been a video instead of being a still photo, these newly created patterns would move and flicker in a totally uncontrolled way. This kind of moving pattern often appears in recent commercial DVDs. They have simply not been sampled at a high enough rate. The second panel in Figure I.5 is a version of Victor that has been smoothed enough so that we no longer see the stripes in the sweater. This image is sampled the same way—every fourth pixel horizontally and vertically—and appears in panel four. It is not a good image, but there are no longer the kinds of artifacts that appear in the third image. To compare the images we have magnified the sampled versions by a factor of four. This example also shows that simply subsampling an image is a poor way to compress it.

This pragmatic discussion and the experiment have their theoretical counterpart, namely, Shannon's theory of sampling. Briefly, Shannon's theorem, in the two-dimensional case, states that for an image to be accurately reconstructed from samples, the image must be bandlimited, which means that it contains no spatial frequencies greater than some bound λ, and the sampling rate must be higher than a factor of λ. Some implications of these statements are that the image u must be infinitely differentiable, that its domain of definition is all of $\mathbb{R}^2$, and that there must be an infinite number of samples to accurately reconstruct u. Furthermore, in Shannon's theory, the image u is reconstructed as an infinite series of trigonometric functions. Note that this is very different from what was done in Figure I.5. So what does this have to do with the problems addressed in this book? What does this have to do with, say, a hypothesized u_S in $BV(\mathbb{R}^2)$ that is definitely not bandlimited? Our answer, which may smack of smoke and mirrors, is that we always are working in two parallel worlds, the theoretical one and the practical one based on numerical computations, and that these two worlds live together in harmony at a certain scale. Here is an example of what we mean: Suppose that u is not a bandlimited image. To sample it properly we would first have to smooth it with a bandlimited kernel. Suppose that instead we smooth it with the Gaussian G_t, which is not bandlimited. Theoretically this is wrong, but practically, the spectrum of G_t, which is G_t itself, decays exponentially. If $|\boldsymbol{x}|^2/4t$ is sufficiently large, then $G_t(\boldsymbol{x})$ appears as zero in computations, and thus it is "essentially" bandlimited. Arguments like this could be made for other situations, but the important point for the reader to keep in mind is that the parallel developments, theory and practice, make sense in the limit.

In the next section, we present a survey of most of the PDEs that have been proposed for image analysis. This provides an informal account of the mathematics that will be developed in detail in the following chapters.

We wish to end this section with a mild disclaimer, and for this we take a page from *Theory of Games and Economic Behavior* by John von Neumann and Oskar Morgenstern where they comment on their theory of a zero-sum two-person game [254] p. 147:

> We are trying to find a satisfactory theory,—at this stage for the zero-sum two-person game. Consequently we are not arguing deductively from the firm basis of an existing theory—which has already stood all reasonable tests—but we are searching for such a theory.. . . This consists in imagining that we have a satisfactory theory of a certain desired type, trying to picture the consequences of this imaginary intellectual situation, and then drawing conclusions from this as to what the hypothetical theory must be like in detail. If this process is applied successfully, it may narrow the possibilities for the hypothetical theory of the type in question to such an extent that only one possibility is left,—i.e. that the theory is determined, discovered by this device. Of course, it can happen that the application is even more "successful," and that it narrows the possibilities down to nothing—i.e. that it demonstrates that a consistent theory of the kind desired is inconceivable.

We take much the same philosophical position, and here is our variation on the von Neumann–Morgenstern statement: We do not suggest that what will be developed here is a necessary future for image analysis. However, if image analysis requires a smoothing theory, then here is how it should be done, and here is the proof that there is no other way to do it. This statement does not exclude the possibility of other theories, based on different principles, or even the impossibility of making any theory.

I.3 PDEs and image processing

We have argued that smoothing—suppressing high spatial frequencies—is a necessary part of image processing in at least two situations: An image needs to be smoothed before features can be extracted, and images must be smoothed before they are sampled. We have also mentioned that, while smoothing with the Gaussian is not a good candidate for the first situation (we will see that it is not contrast invariant, and it is not well localized except for small t), it is not unreasonable to use it numerically in the second situation, since it does a good job of suppressing high frequencies. These smoothing requirements and the fact that the Gaussian is the fundamental solution of the heat equation mean that the heat equation appears completely naturally in image processing, and indeed it is the first PDE to enter the picture in Chapters 1 and 2. Smoothing with the heat equation is illustrated in Figure I.6.

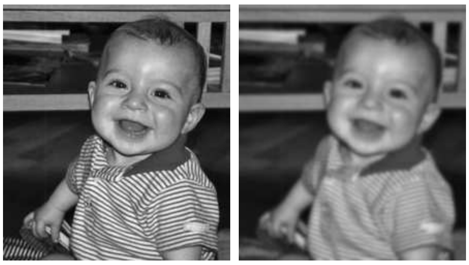

Figure I.6: Heat equation and smoothing. The original image is on the left; the heat equation has been applied at some scale, and the resulting blurred image is on the right.

There is another path hinted at in section I.1 that leads to the Gaussian and thus to the heat equation. Suppose that k is any positive kernel such that $k(\boldsymbol{x}) = k(|\boldsymbol{x}|)$ and such that k is localized in the sense that $k(\boldsymbol{x}) \to 0$ sufficiently rapidly as $|\boldsymbol{x}| \to \infty$. If k is normalized properly and if we write $k_h(\mathbf{x}) = (1/h)k(\mathbf{x}/h^{1/2})$, then

$$\frac{k_h * u_0(\mathbf{x}) - u_0(\mathbf{x})}{h} \to \Delta u_0(\mathbf{x})$$

as $h \to 0$ whenever the image u_0 is sufficiently smooth. We write this as

$$k_h * u_0(\mathbf{x}) - u_0(\mathbf{x}) = h\Delta u_0(\mathbf{x}) + o(h). \tag{I.1}$$

Now let $u(t, \boldsymbol{x})$ denote the solution of the heat equation

$$\frac{\partial u}{\partial t} = \Delta u, \quad u(0, \boldsymbol{x}) = u_0(\boldsymbol{x}).$$

If u_0 is sufficiently smooth, then we can write

$$u(t, \boldsymbol{x}) - u(0, \boldsymbol{x}) = t\Delta u_0(\mathbf{x}) + o(t). \tag{I.2}$$

The reverse heat equation

Equations (I.1) and (I.2) suggest that blurring u_0 with a kernel k_h for small h is equivalent to applying the heat equation to u_0 at some small scale t. This is true, and it will be made precise in Chapter 2. These equations also lead to another idea: We read in the paper [147] by Lindenbaum, Fischer, and Bruckstein that Kovasznay and Joseph [136] introduced in 1955 the notion that a slightly blurred image could be deblurred by subtracting a small amount of its Laplacian. Numerically, this amounts to subtracting a fraction λ of the Laplacian of the observed image from itself:

$$u_{\text{restored}} = u_{\text{observed}} - \lambda \Delta u_{\text{observed}}.$$

Dennis Gabor, who received the Nobel prize in 1971 for his invention of optical holography, studied this process and determined that the best value of λ was the one that doubled the steepest slope in the image [147]. Empirically, one can start with a small value of λ and repeat the process until a good image is obtained; with further repetitions the process blows up. Indeed, this process is just applying the reverse heat equation to the observed image, and the reverse heat equation is notoriously ill-posed. On the other hand, the Kovasznay–Joseph–Gabor method is efficient for sufficiently small λ and can be successfully applied to most images obtained from optical devices. This process is illustrated in Figure I.7. A few iterations can enhance the image (second panel), but the inverse heat equation finally blows up (third panel).

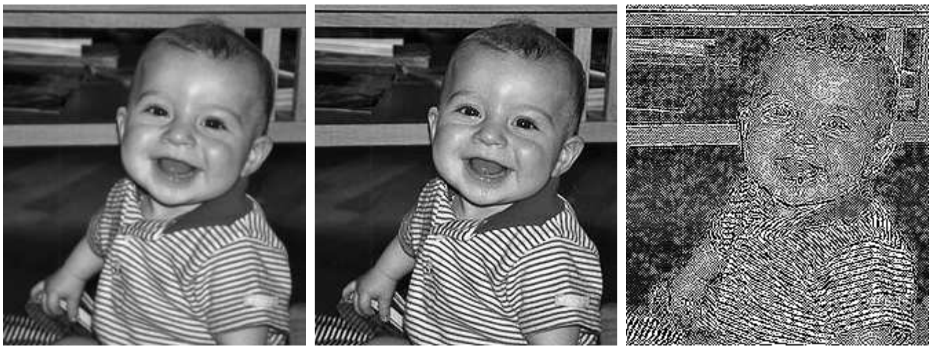

Figure I.7: Kovasznay–Joseph–Gabor deblurring. Left to right: original image; three iterations of the algorithm; ten iterations of the algorithm.

Figure I.8 shows that same experiment applied to an image of Victor that has been numerically blurred. Again, the process blows up, but it yields a significant improvement at some scales.

We have now seen the heat equation used in two senses, each with a different objective. In both cases, we have noted drawbacks. In the first instance, the heat equation (or Gaussian) was used to smooth an image, but as we have mentioned, this operator is not contrast invariant, and thus is not appropriate for any theory of image analysis that requires contrast-invariant operators. This does not mean that the Gaussian should be dismissed; it only means that it is not appropriate for our version of image analysis. To meet our objectives, we will replace the Laplacian, which is a linear isotropic operator, with nonlinear,

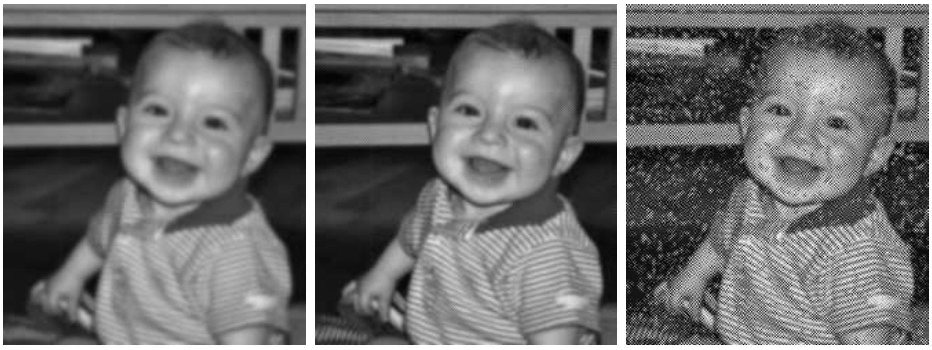

Figure I.8: Kovasznay–Joseph–Gabor deblurring. This is the same deblurring experiment as in Figure I.7, but it is applied to a much more blurred image.

nonisotropic smoothing operators. This will bring us to the central theme of the book: appropriate smoothing for a possible theory of image analysis.

In the second instance, the heat equation is run backward (the inverse heat equation) with the objective of restoring a blurred image. As we have seen, this is successful to some extent, but the drawback is that it is an unstable process. The practical problem is more complex than the fact that the inverse heat equation is not well posed. In the absence of noise, the best way to deblurr a slightly blurred image is to use the inverse heat equation. However, in the presence of noise, this isotropic operator acts equally in all direction, and while it enhances the definition of edges, the edges become jagged due to the noise. This observation led Gabor to try to improve matters by using more directional operators in place of the Laplacian. Gabor was concerned with image restoration, but his ideas will appear later in our story in connection with smoothing. (For an account of Gabor's work see [147].)

Shock filters

The objective for running the heat equation backward is image restoration, and although restoration is not the main subject of the book, we are going to pause here to describe two ways to improve the stability of the inverse heat equation. Image restoration is an extremely important area of image processing, and the techniques we describe illustrate another use of PDEs in image processing. There are indeed stable ways to "reverse" the heat equation. More precisely, there are "inverse diffusions" that deblurr an image and reach a steady state. The first example, due to Rudin in 1987 [214] and Osher and Rudin in 1990 [198] is a pseudoinverse for the heat equation, where the propagation term $|Du| = |(u_x, u_y)|$ is controlled by the sign of the Laplacian:

$$\frac{\partial u}{\partial t} = -\text{sign}(\Delta u)|Du|. \tag{I.3}$$

This equation is called a *shock filter*. We will see later that this operator propagates the level lines of an image with a constant speed and in the same direction as the reverse heat equation would propagate these lines; hence it acts as a pseudoinverse for the heat equation. This motion enhances the definition of the contours and thus sharpens the image. Equation (I.3) is similar to a classic nonlinear filter introduced by Kramer in the seventies [137]. Kramer's filter can be interpreted in terms of a PDE using the same kinds of heuristic arguments that have been used to derive the heat equation. This equation is

$$\frac{\partial u}{\partial t} = -\mathrm{sign}(D^2u(Du, Du))|Du|, \tag{I.4}$$

where the Laplacian has been replaced by

$$D^2u(Du, Du) = u_{xx}(u_x)^2 + 2u_{xy}u_xu_y + u_{yy}(u_y)^2. \tag{I.5}$$

We will see in Chapter 2 that $D^2u(Du, Du)/|Du|^2$ is the second derivative of u in the direction of its gradient Du, and we will interpret the differential operator (I.5) as Haralick's edge detector. Kramer's equation yields a slightly better version of a shock filter. The actions of these filters are illustrated in Figure I.9. The image on the left is a blurred image of Victor. The next image has been deblurred using the Rudin–Osher shock filter. This is a pseudoinverse of the heat equation that attains a steady state. The third image has been deblurred using Kramer's improved shock filter, which also attains steady state. The fourth image was deblurred using the Rudin–Osher–Fatemi restoration scheme, which is described below [215].

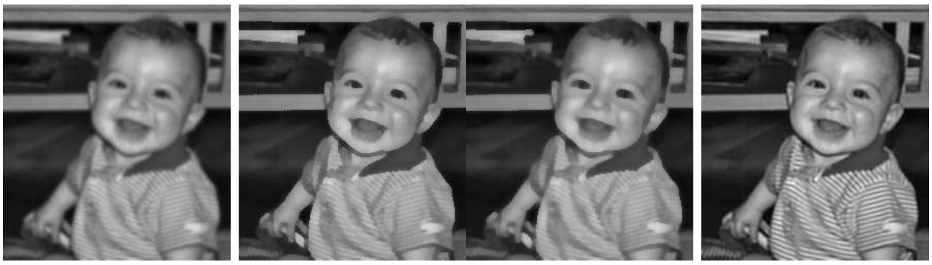

Figure I.9: Deblurring with shock filters and a variational method. Left to right: blurred image; Rudin–Osher shock filter; Kramer's improved shock filter; Rudin–Osher–Fatemi restoration method.

The deblurring algorithms (I.3) and (I.4) work to the extent that, experimentally, they attain steady states and do not blow up. However, a third deblurring method, the Rudin–Osher–Fatemi algorithm, is definitely better. It poses the deblurring problem as an inverse problem. It is very efficient when the observed image u_0 is of the form $k * u + n$, where k is known and where the statistics of the noise n are also known. Given the observed image u_0, one tries to find a restored version u such that $k * u$ is as close as possible to u_0 and such that the oscillation of u is nonetheless bounded. This is done by finding u that minimizes the functional

$$\int \left(|Du(\boldsymbol{x})| + \lambda(k * u(\boldsymbol{x}) - u_0(\boldsymbol{x}))^2\right) \mathrm{d}\mathbf{x}. \tag{I.6}$$

The parameter λ controls the oscillation in the restored version u. If λ is large, the restored version will closely satisfy the equation $k * u = u_0$, but it may be very oscillatory. If instead λ is small, the solution is smooth but inaccurate. This parameter can be computed in principle as a Lagrange multiplier. The obtained restoration can be remarkable. The best result we can obtain with the blurred Victor is shown in the fourth panel of Figure I.9. This scheme was selected by the French Space Agency (CNES) after a benchmark for satellite image deblurring, and it is currently being used by the CNES for satellite image restoration. This total variation restoration method also has fast wavelet packets versions.

From the heat equation to wavelets

The observation by Kovasznay, Joseph, and Gabor (and undoubtedly others) that the difference between a smoothed image and the original image is related to the Laplacian of the original image is also the departure of one of the paths that lead to wavelet theory. Here, very briefly, is the idea: If we convolve an image with an appropriate smoothing kernel and then take the difference, we obtain a new image related to the Laplacian of the original image (see equation (I.1)). This new "Laplacian image" turns out to be faded with respect to the original, and if one retains only the values greater than some threshold, the image is often sparse. This is illustrated in Figure I.10. The last panel on the right shows in black the values of this Laplacian image of Victor that differ significantly from zero. Here, and in most natural images, this representation is sparse and thus useful for compression. This experiment simulates the first step of a well-known algorithm due to Burt and Adelson.

In 1983, Burt and Adelson developed a compression algorithm called the Laplacian pyramid based on this idea [36]. Their algorithm consists of iterating two operations: a convolution followed by subsampling. After each convolution, one keeps only the difference $k_n * u_n - u_n$, where n is used here to indicate that each step takes place at a different scale due to the subsampling. The image is then coded by the (finite) sequence of these differences. These differences resemble the Laplacian of u_n, hence the name "Laplacian pyramid." An important aspect of this algorithm is that the discrete kernels k_n, which are low-pass filters, are all the same kernel k; the index n merely indicates that k is adjusted for the scale of the space where the subsampled image u_n lives. Ironically, the smoothing function cannot be the Gaussian, since the requirements for reconstructing the image from its coded version rule out the Gaussian. Burt and Adelson's algorithm turned out to be one of the key steps that led to multiresolution analyses and wavelets. Burt and Adelson were interested in compression, and, indeed, the differences $k_n * u_n - u_n$ tend to be sparse for natural images. On the other hand, we are interested in image analysis, and for us, the Burt and Adelson algorithm has the drawback that it is not translation invariant or isotropic because of the multiscale subsampling.

Back to edge detection

Early research in computer vision focused on *edge detection* as a main tool for image representation and analysis. It was assumed that the apparent contours of objects, and also the boundaries of the facets of objects, produce step discontinuities, while inside these boundaries, the image oscillates mildly. The apparent contour points, or *edges points*, were to be computed as points where the gradient is in some sense largest. Two ways were proposed to do this: Marr and Hildreth proposed computing the points where Δu crosses zero, the now-famous

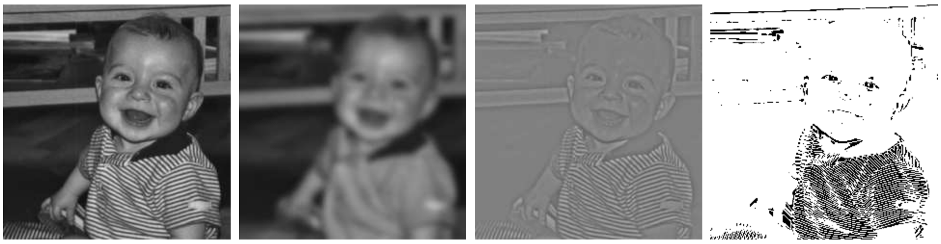

Figure I.10: The Laplacian pyramid of Burt and Adelson. Left to right: the original image; the image blurred by Gaussian convolution; the difference between the original image and the blurred version, which approximates the Laplacian of the original image; the points where this Laplacian image is large.

zero-crossings [165]. A significant improvement was made by Harakick who defined the boundaries, or edges, of an image as those points where $|Du|$ attains a local maximum along the gradient lines [101]. Two years later, Canny implemented Haralick's detector in an algorithm that consists of Gaussian smoothing followed by computing the (edge) points where $D^2u(Du, Du) = 0$ and $|Du|$ is above some threshold [38]. We refer to this algorithm as the Haralick–Canny edge detector. The fourth panel in Figure I.11 displays what happens when we smooth the image with the Gaussian (the heat equation) and then compute the points where $D^2u(Du, Du) = 0$ and $|Du|$ is above some threshold. If this computation is done on the raw image (first panel), then "edges" show up everywhere (second panel) because the raw image is a highly oscillatory function and contains a very dense set of inflexion points. After applying the heat equation and letting it evolve to some scale (third panel), we see that the Haralick–Canny edge detector is able to extract some meaningful structure.

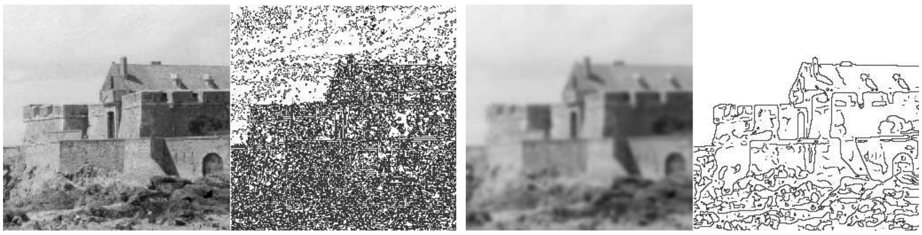

Figure I.11: Heat equation and Haralick's edge detector. Left to right: original image; edge points found in the original image using Haralick's detector; blurred image; edges found in the blurred image using the Haralick–Canny detector. The image "edges" are singled out after the image has been smoothed. This smoothing eliminates tiny oscillations and maintains the big ones.

The Perona-Malik equation

Given certain natural requirements such as isotropy, localization, and scale invariance, the heat equation is the only good linear smoothing operator. There are, however, many nonlinear ways to smooth an image. The first one was proposed by Perona and Malik in 1987 [203, 204]. Roughly, the idea is to smooth what needs to be smoothed, namely, the irrelevant homogeneous regions, and to enhance the boundaries. With this in mind, the diffusion should look like the heat equation when $|Du|$ is small, but it should act like the inverse heat equation when $|Du|$ is large. Here is an example of a Perona–Malik equation in divergence form:

$$\frac{\partial u}{\partial t} = \operatorname{div}(g(|Du|)Du), \tag{I.7}$$

where $g(s) = 1/(1+\lambda^2 s^2)$. It is easily checked that we have a diffusion equation when $\lambda|Du| \leq 1$ and an inverse diffusion equation when $\lambda|Du| > 1$. To see this, consider the second derivative of u in the direction of Du,

$$u_{\xi\xi} = D^2u\left(\frac{Du}{|Du|}, \frac{Du}{|Du|}\right),$$

and the second derivative of u in the orthogonal direction,

$$u_{\eta\eta} = D^2u\left(\frac{Du^\perp}{|Du|}, \frac{Du^\perp}{|Du|}\right),$$

where $Du = (u_x, u_y)$ and $Du^\perp = (-u_y, u_x)$. The Laplacian can be rewritten in the intrinsic coordinates (ξ, η) as $\Delta u = u_{\xi\xi} + u_{\eta\eta}$. The Perona–Malik equation then becomes

$$\frac{\partial u}{\partial t} = \frac{1}{1+\lambda^2|Du|^2} u_{\eta\eta} + \frac{1-\lambda^2|Du|^2}{(1+\lambda^2|Du|^2)^2} u_{\xi\xi}.$$

The first term in this representation always appears as a one-dimensional diffusion in the direction orthogonal to the gradient, tuned by the size of the gradient. The nature of the second term depends on the value of the gradient; it can be either diffusion in the direction Du or inverse diffusion in the same direction. This model indeed mixes the heat equation and the reverse heat equation. Figure I.12 is used to compare the Perona–Malik equation with the classical heat equation (illustrated in Figure I.11) in terms of accuracy of the boundaries obtained by the Haralick–Canny edge detector (see Chapter 6). At a comparable scale of smoothing, we clearly gain some accuracy in the boundaries and remove more "spurious" boundaries using this Perona–Malik equation. The representation is both more sparse and more accurate.

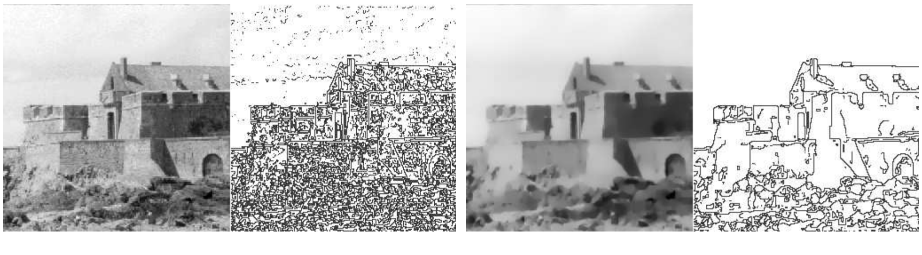

Figure I.12: A Perona–Malik equation and edge detection. This is the same experiment as in Figure I.11, but here the Perona–Malik equation is used in place of the heat equation. Notice that the edge map looks slightly better in this case.

The ambitious Perona–Malik model attempts to build into a single operator the ability to perform two very different tasks, namely, restoration and analysis. This has its cost: The model contains a "contrast threshold" λ^{-1} that must be set manually, and although experimental results have been impressive, the mathematical existence and uniqueness of solutions are not guaranteed, despite some partial results by Kichenassamy [127] and Weickert and Benhamouda [257]. There are three parameters involved in the overall smoothing and edge-detecting scheme: the gradient threshold λ^{-1} in the equation (6.2), the smoothing scale(s) t (or the time that equation (6.2) evolves), and the gradient threshold in the Haralick–Canny detector. We can use the same gradient threshold in both the Haralick–Canny detector and the Perona–Malik equation, but this still leaves us with a two-parameter algorithm. Can these parameters be dealt with automatically for an image analysis scheme? This question seems to have no general answer at present. An interesting attempt based on statistical arguments had been made, however, by Black et al. [29].

A proliferation of PDE's

If one believes that some nonlinear diffusion might be a good image analysis model, why not try them all? This is exactly what has happened during the last ten years. We can claim with some certainty that almost all possible nonlinear parabolic equations have been proposed. A few of the proposed models are even systems of PDEs. The common theme in this proliferation of models is this: Each attempt fixes one intrinsic diffusion direction and tunes the diffusion using the size of the gradient or the value of an estimate of the gradient. To keep the size of this introduction reasonable, we will focus on a few of the simplest models.

We begin with the Rudin–Osher–Fatemi model [215]. In this model the BV norm of u, $\int |Du(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{x}$, is one of the terms in the expression (I.6) that is minimized to obtain a restored image. It is this term that provides the smoothing. The gradient descent for $\int |Du(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{x}$ translates into the equation

$$\frac{\partial u}{\partial t} = \operatorname{div}\left(\frac{Du}{|Du|}\right) = \frac{1}{|Du|}u_{\eta\eta}.$$

Written this way, the method appears as a diffusion in the direction orthogonal to the gradient, tuned by the size of the gradient. Andreu et al. proved that this equation is well posed in the space BV of functions of bounded variation [12, 13]. A variant of this model was proposed independently by Alvarez, Lions, and Morel [8]. In this case, the relevant equation is

$$\frac{\partial u}{\partial t} = \frac{1}{|k * Du|}\,|Du|\operatorname{div}\left(\frac{Du}{|Du|}\right) = \frac{1}{|k * Du|}u_{\eta\eta},$$

and again the diffusion is in the direction $Du^{\perp}$ orthogonal to the gradient. Note that the rate of diffusion depends on the average value $k * Du$ of the gradient in a neighborhood of $\boldsymbol{x}$, whereas the direction of diffusion, $Du^{\perp}(\boldsymbol{x})/|Du(\boldsymbol{x})|$, depends on the value of $Du(\boldsymbol{x})$ at $\boldsymbol{x}$. The kernel k is usually the Gaussian. Kimia, Tannenbaum, and Zucker, working in a more general shape-analysis framework, proposed the simplest equation of our list [131]:

$$\frac{\partial u}{\partial t} = |Du|\operatorname{div}\left(\frac{Du}{|Du|}\right) = D^2u\left(\frac{Du^{\perp}}{|Du|}, \frac{Du^{\perp}}{|Du|}\right) = u_{\eta\eta}. \tag{I.8}$$

This equation had been proposed earlier in another context by Sethian as a tool for front-propagation algorithms [233]. This equation is a "pure" diffusion in the direction orthogonal to the gradient. We call this equation *the curvature equation*; this is to distinguish it from other equations that depend on the curvature of u in some other way. These latter will be called *curvature equations.* When we refer to the action of the equations, we often write *curvature motions* or *curvature-dependent motions.* (See Chapters 17 and 18.)

The Weickert equation can be viewed as a variant of the curvature equation [256]. It uses a nonlocal estimate of the direction orthogonal to the gradient for the diffusion direction. This direction is computed as the direction v of the eigenvector corresponding to the smallest eigenvalue of $k * (Du \otimes Du)$, where $(\boldsymbol{y} \otimes \boldsymbol{y})(\boldsymbol{x}) = (\boldsymbol{x} \cdot \boldsymbol{y})\boldsymbol{y}$. Note that if the convolution kernel is removed, then this eigenvector is simply $Du^{\perp}$. So the equation writes

$$\frac{\partial u}{\partial t} = u_{\eta\eta}, \tag{I.9}$$

where η denotes the coordinate in the direction v. The three models just described can be interpreted as diffusions in a direction orthogonal to the gradient (or an estimate of this direction), tuned by the size of the gradient. They are illustrated in Figure I.13. (The original image is in the first panel of Figure I.14.)

Carmona and Zhong proposed a diffusion in the direction of the eigenvector w corresponding to the smallest eigenvalue of D^2u [43]. So the equation is again 3.18, but this time η denotes the coordinate in the direction of w. This is illustrated in panel three of Figure I.14. Sochen, Kimmel, and Malladi propose instead a nondegenerate diffusion associated with a minimal surface variational formulation [236]. Their idea was to make a gradient descent for the area, $\int \sqrt{1 + |Du(\boldsymbol{x})|^2}\, \mathrm{d}\boldsymbol{x}$, of the graph of u. This leads to the diffusion equation

$$\frac{\partial u}{\partial t} = \operatorname{div}\left(\frac{Du}{\sqrt{1 + |Du|^2}}\right).$$

At points where Du is large this equation behaves like $\frac{\partial u}{\partial t} = \operatorname{div}\left(\frac{Du}{|Du|}\right)$, where we retrieve the Rudin-Osher-Fatemi model of Section I.3. At points where Du is small we have $\frac{\partial u}{\partial t} = \operatorname{div}(Du)$ which is the heat equation. This equation is illustrated in panel four of Figure I.14. Other diffusions have also been considered. For purposes of interpolation, Caselles, Morel, and Sbert proposed a diffusion that may be interpreted as the strongest possible image smoothing [49],

$$\frac{\partial u}{\partial t} = D^2u(Du, Du) = |Du|^2 u_{\xi\xi}.$$

This equation is not used for preprocessing the image as the others are; rather, it is a way to interpolate between the level lines of an image with sparse level lines (Figure I.15). Among the models mentioned, only the curvature motion proposed by Kimia, Tannenbaum, and Zucker was specifically introduced as a shape analysis tool. We are going to explain this, but to do so we must say more about image analysis.

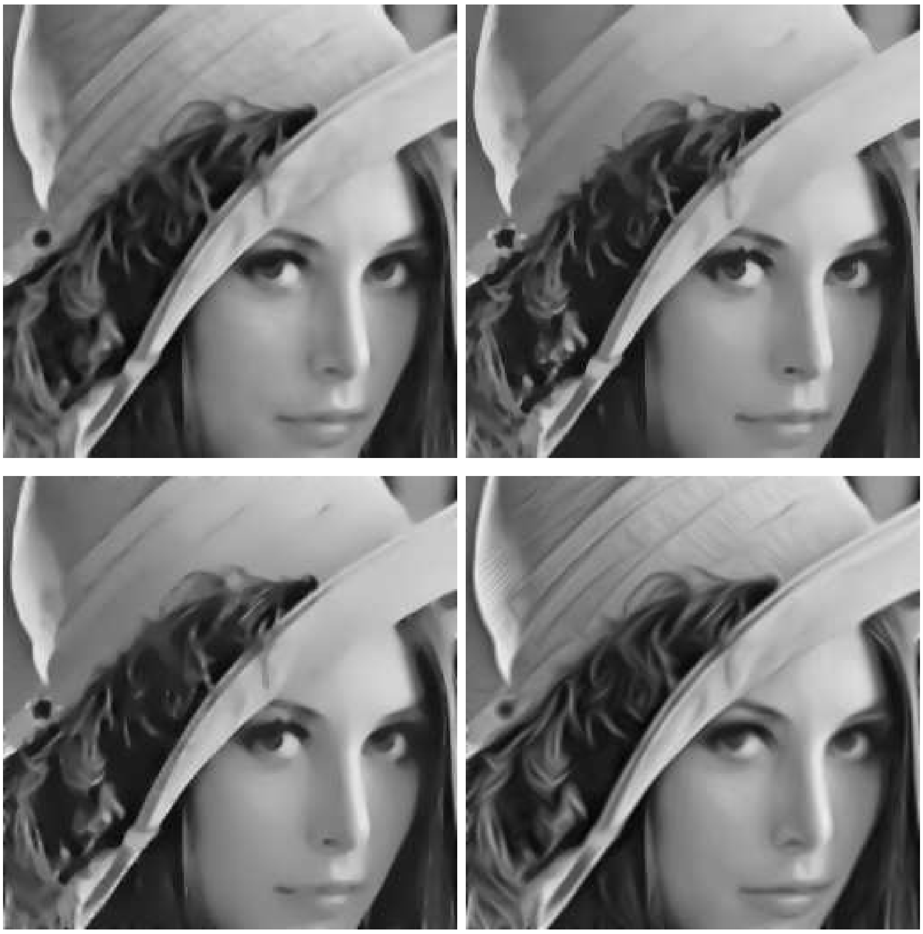

Figure I.13: Diffusion models I. Left to right: Osher, Sethian 1988: the curvature equation; Rudin, Osher, Fatemi 1992: minimization of the image's total variation; Alvarez, Lions, Morel 1992: nonlocal variant of the preceding; Weickert 1994: nonlocal variant of the curvature equation. All of these models diffuse only in the direction orthogonal to the gradient, using a more or less local estimate of this direction. This explains why the results of the filters are so similar. However, the Weickert model captures better the texture direction.

Principles of image analysis

There are probably as many ways to approach image analysis as there are uses of digital images, and today the range of applications covers much of human activity. Most scientific and technical activities, including particularly medicine, and even sound analysis (visual sonograms), involve the perceptual analysis of images. Our goal is to look for fundamental principles that underlie most of these applications and to develop algorithms that are widely applicable. From a less lofty point of view, we wish to examine the collection of existing and potential image operators to determine which among them fit our vision of

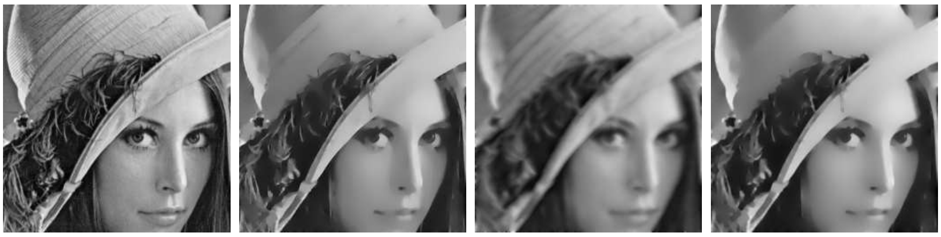

Figure I.14: Diffusion models II. Left to right: original image; Perona–Malik equation 1987, creating blurry parts separated by sharp edges; Carmona, Zhong 1998 which actually blurs the whole image: diffusion along the least eigenvector of D^2u; Sochen, Kimmel, Malladi 1998: minimization of the image graph area. This last equation has effects similar to the Perona-Malik model.

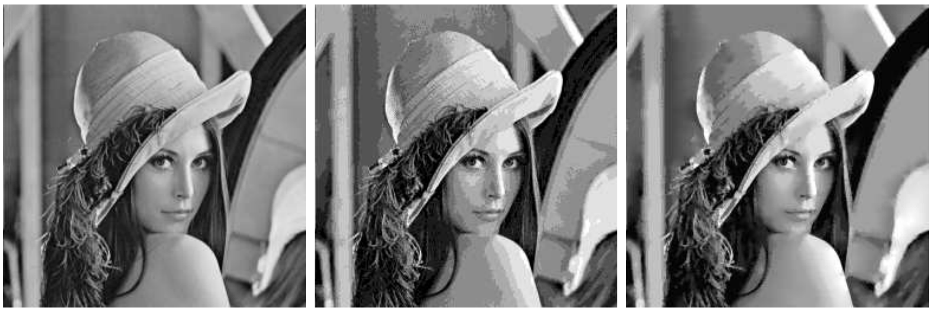

Figure I.15: Diffusion models III. Left to right: original image; quantized image (only 10 levels are kept - 3.32 bits/pixel); the quantized image reinterpolated using the Caselles–Sbert algorithm 1998. They apply a diffusion on the quantized image with values on the remaining level lines taken as boundary conditions.

image analysis. Instead of examining an endless list of partial and specific requirements, we rely on a mathematical shortcut, well known in mechanics, that consists of stating a short list of invariance requirements. These invariance requirements will lead to a classification of models and point out the ones that are the most suitable as image analysis tools. The first invariance requirement is the Wertheimer principle according to which visual perception (and therefore, we add, image analysis) should be independent of the image contrast [260]. We formalize this as follows:

Contrast-invariant classes. Two images u and v are said to be (perceptually) equivalent if there is a continuous increasing function g such that $v = g(u)$. In this case, u and v are said to belong to the same contrast-invariant class. ("Increasing" always means "strictly increasing.")

Contrast invariance requirement. An image analysis operator T must act directly on the equivalence class. As a consequence, we ask that $T(g(u)) = g(Tu)$, which means that the image analysis operator commutes with contrast changes.

The contrast invariance requirement rules out the heat equation and all of the models described above except the curvature motion (I.8). Contrast invariance led Matheron in 1975 to formulate image analysis as set analysis, namely, the analysis of the level sets of an image. The *upper level set* of an image u at level λ is the set

$$\mathcal{X}_\lambda u = \{\mathbf{x} \mid u(\mathbf{x}) \geq \lambda\}.$$

We define in exactly the same way the *lower level sets* by changing "$\geq$" into "$\leq$." The main point to retain here is the global invariance of level sets under contrast changes. if g is a continuous increasing contrast change, then

$$\mathcal{X}_{g(\lambda)} g(u) = \mathcal{X}_\lambda u.$$

According to mathematical morphology, the image analysis doctrine founded by Matheron and Serra, the essential image shape information is contained in its level sets. It can be proved (Chapter 11) that an image can be reconstructed, up to a contrast change, from its set of level sets [169]. Figure I.16 shows an image and one of its level sets.

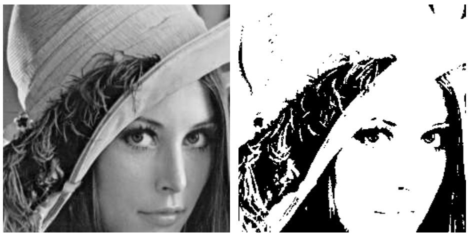

Figure I.16: An image and one of its level sets. On the right is level set 140 of the left image. This experiment illustrates Matheron's thesis that the main shape information is contained in the level sets of an image. Level sets are contrast invariant.

The contrast invariance requirement leads to powerful and simple denoising operators like the so-called *extrema killer, or area opening,* (Chapter 13) defined by Vincent in 1993 [253]. This image operator simply removes all connected components of upper and lower level sets with areas smaller than some fixed value. This operator is not a PDE; actually it's much simpler. Its effect is amazingly good for impulse noise, which includes the local destruction of the image and spots. The action of the extrema killer is illustrated in Figure I.17. The original image is in the first panel. In the third panel, the image has been degraded by adding "salt and pepper" noise to 75% of the pixels. The next panel shows its restoration using the extrema killer set to remove upper and lower level sets with areas smaller than 80 pixels. The second panel shows the result of the same operator applied to the original.

Level lines as a complete contrast invariant representation

In 1996, Caselles, Coll, and Morel further localized the contrast invariance requirement in image analysis. They proposed as the main objects of analysis the *level lines* of an image, that is, the boundaries of its level sets [46]. For this program—and the previous one involving level sets—to make sense, the levels sets and level lines must have certain topological and analytic properties. Level sets and isolevel sets $\{\mathbf{x} \mid u(\mathbf{x}) = \lambda\}$, which we would like to be the "level lines,"

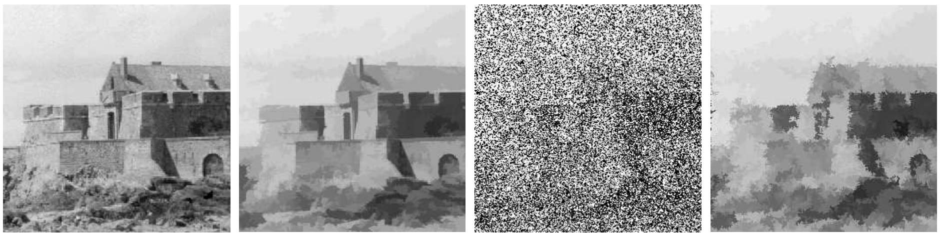

Figure I.17: The extrema killer filter. Left to right: original image; extrema killer applied with area threshold equal 80 pixels; 75% salt and pepper noise added to the original image; the same filter applied.

can be defined for any image (or function) u, but they will not necessarily be useful for image analysis. In particular, we cannot directly define useful level sets and level lines for a digital image u_d. What is needed is a representation of u_d for which these concepts make sense. But this is not a problem. By the assumptions of section I.1, a digital representation u_d of a natural image S has been obtained by suitably sampling a smooth version of S, call it u_f, and a smooth approximation of u_f is available to us by interpolation. There are, of course, different interpolation methods to produce smooth representations of u_d. One can also obtain a useful discontinuous representation by considering the extension of u_d that is constant on each pixel. For an interpolation method to be useful, the level lines should have certain minimal properties: They should be composed of a finite number of rectifiable Jordan curves, and they should be nested. This means that they do not cross, and thus that they form a tree by inclusion (Section 17.2.)

A study by Kronrod in 1950 shows that if the function u is continuous, then the isolevels sets $\{\mathbf{x} \mid u(\mathbf{x}) = \lambda\}$ are nested and thus form a tree when ordered by inclusion [139]. These isolevel sets are not necessarily curves; they are curves, however, if u has continuous first derivatives. Monasse proved Kronrod's result for lower semicontinuous and upper semicontinuous functions in 2000 [180] (see also [21]). His result implies that the extension of u_d that is constant on each pixel yields a nested set of Jordan curves bounding the pixels. Thus we have at least two ways to associate a set of nested Jordan curves with a digital image u_d, depending on how u_d is interpolated. Given an interpolation method, we call this set of nested curves a *topographic map* of the image.[1] By introducing the topographic map, the search for image smoothing, which had already been reduced to set smoothing, is further reduced to curve smoothing. Of course, we require that this smoothing preserves curve inclusion. Level lines of an image at a fixed level are shown in Figure I.18.

[1] The use of level lines is also consistent with the "BV assumption" mentioned in section I.1, according to which the correct function space for modeling images is the space BV of functions of bounded variation. In this case, the coarea formula can be used to associate a set of Jordan curves with an image (see [11]) It is, however, in general false for BV functions that the boundaries of lower and upper level sets form a nested set of curves; these curves may cross (see again [180].)

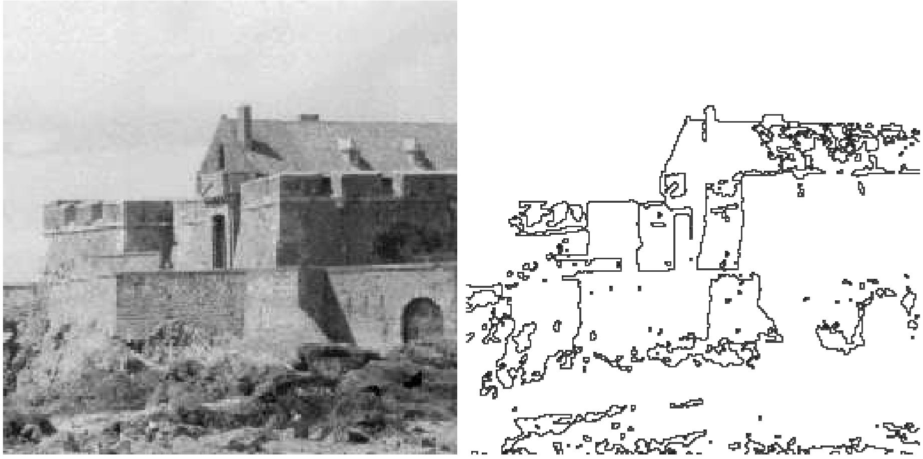

Figure I.18: Level lines of an image. Level lines, defined as the boundaries of level sets, can be defined to be a nested set of Jordan curves. They provide a contrast-invariant representation of the image. On the right are the level lines at level 183 of the left image.

Contrast invariant PDE's

Chen, Giga, and Goto [53, 54] and Alvarez et al. [7] proved that if one adds contrast invariance to the usual isotropic invariance requirement for image processing, then all multiscale image analyses should have a curvature-dependent motion of the form

$$\frac{\partial u}{\partial t} = F(\mathrm{curv}(u), t)|Du|, \tag{I.10}$$

where F is increasing with respect to its first argument (see chapters **??** and **??**). This equation can be interpreted as follows: Consider a point $\mathbf{x}$ on a given level curve C of u at time t. Let $n(\mathbf{x})$ denote the unit vector normal to C at $\mathbf{x}$ and let $\mathrm{curv}(\mathbf{x})$ denote its curvature. Then the preceding equation is associated with the curve motion equation

$$\frac{\partial \mathbf{x}}{\partial t} = F(|\boldsymbol{\kappa}|(\mathbf{x}), t)n(\mathbf{x})$$

that describes how the point $\mathbf{x}$ moves in the direction of the normal. The formula defining $curv(u)$ at a point $\mathbf{x}$ is (Chapter 17)

$$curv(u)(\mathbf{x}) = \frac{1}{|Du|^3} D^2u(Du^\perp, Du^\perp)(\mathbf{x}) = \frac{u_{xx}u_y^2 - 2u_{xy}u_xu_y + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{3/2}}(\mathbf{x}).$$

The curvature vector at a point of a C^2 curve is the second derivative for a curve $\mathbf{x}(s)$ parameterized by length : $\boldsymbol{\kappa} = \mathrm{d}^2\mathbf{x}/\mathrm{d}s^2$. We refer to Chapter 17 for the detailed definitions and the links between the curvature vector of a level line of u and $curv(u)$. Not much more can be said at this level of generality about F. Two specific cases play prominent roles in this subject. The first case is $F(\mathrm{curv}(u), t) = \mathrm{curv}(u)$, the curvature equation (I.8). The second case is $F(\mathrm{curv}(u), t) = (\mathrm{curv}(u))^{1/3}$.

This particular one-third power form for the curvature dependence provides an important additional invariance, namely, affine invariance. We would like to have complete projective invariance, but a theorem proved by Alvarez et al. shows that this is impossible [7] (Chapter **??**). The best we can have is invariance with respect to the so-called Chinese perspective, which preserves parallelism. Most of these equations, particularly when F is a power of the curvature, have a viscosity solution in the sense of Crandall and Lions [62]. This was shown in 1995 by Ishii and Souganidis [118]. We refer to Chapters **??** and **??** for all details.

As we have mentioned, contrast-invariant processing can be reduced to level set processing and, finally, to level curve processing. The equations mentioned above are indeed equivalent to curve evolution models if existence and regularity have been established. These results exist for the most important cases, namely, for $F(\text{curv}(u), t) = \text{curv}(u)$, called *curve shortening*, and for $F(\text{curv}(u), t) = (\text{curv}(u))^{1/3}$, known as *affine shortening*. Grayson proved existence, uniqueness, and analyticity for the curve shortening equation [98],

$$\frac{\partial \mathbf{x}}{\partial t} = \text{curv}(\mathbf{x}) n(\mathbf{x}), \tag{I.11}$$

NE PAS LAISSER COMMME C'EST : *curv* n'est pas la meme notation qu'apres et n'est pas meme defini!

and Angenent, Sapiro, and Tannenbaum proved the same results for the affine shortening equation [14],

$$\frac{\partial \mathbf{x}}{\partial t} = (\text{curv}(\mathbf{x}))^{\frac{1}{3}} n(\mathbf{x}). \tag{I.12}$$

These results are very important for image analysis because they ensure that the shortening processes do indeed reduce a curve to a more and more sketchy version of itself.

Affine invariance

An experimental verification of affine invariance for affine shortening is illustrated in Figure I.19. The numerical tests were made using a very fast numerical scheme for the affine shortening designed by Lionel Moisan [178]. The principle of this algorithm is explained in Chapter **??**. Unlike many numerical schemes, this one is itself affine invariant. Each of the three panels in Figure I.19 contains three shapes. The first panel shows the action of an affine transformation A: Call the first shape in the first panel X; then the second shape is $A(X)$ and the third shape is $A^{-1}A(X) = X$. The second panel shows that affine shortening, S, commutes with A: The shapes are, from left to right, $S(X)$, $SA(X)$, and $A^{-1}SA(X)$. Since this third shape is the same as the first, we see that $A^{-1}SA(X) = S(X)$, or that $SA(X) = AS(X)$. The third panel shows the same experiment with affine shortening replaced with curve shortening. Since the first and third shapes are different, this illustrates that A does not commute with curve shortening, and hence that curve shortening is not affine invariant.

Evans and Spruck [74] (also [75, 76, 77]) and Chen, Giga, and Goto [53, 54] proved in 1991 that a continuous function moves by the curvature motion

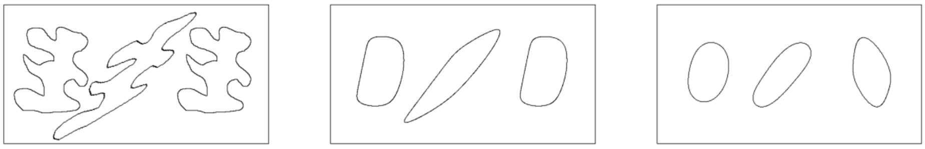

Figure I.19: Experimental verification of the affine invariance of the affine shortening (AMSS). The first panel contains three shapes, X, $A(X)$, and $A^{-1}A(X)$. The second panel contains $S(X)$, $SA(X)$, and $A^{-1}SA(X)$. The congruence of the first and third shapes implies that S and A commute. In the third panel, the same procedure has been applied using equation (I.11). Here the first and third shapes are not congruent, which shows that the curve shortening is not affine invariant, as expected.

(equation (I.10) with $F(\text{curv}(u), t) = \text{curv}(u)$) if and only if almost all of its level curves move by curve shortening (equation (I.11)). The same result is true for the affine invariant curve evolution (equation (I.10) with $F(\text{curv}(u), t) = (\text{curv}(u))^{1/3}$) and affine shortening (equation (I.12)).

In the case of the curvature motion, this result provides a mathematical justification for the now-classic Osher–Sethian numerical method for moving fronts [199]: They associate with some curve or surface C its signed distance function $u(\mathbf{x}) = \pm d(\mathbf{x}, C)$, and the curve or surface is handled indirectly as the zero isolevel set of u. Then u is evolved by, say, the curvature motion with a classic numerical difference scheme. Thus, the evolution of the curve C is dealt with efficiently and accurately as a by-product of the evolution of u. The point of view that we adopt is slightly different from that of Osher and Sethian. We view the image as a generalized distance function to each of its level sets, since we are interested in all of them.

We show in Figure I.20 how the level lines are simplified by evolving the image numerically using affine invariant curvature motion. For clarity, we display only sixteen levels of level curves. Notice that the aim here is not subsampling; we keep the same resolution. Nor is the aim restoration; the processed image is clearly worse than the original. The aim is invariant simplification leading to shape recognition.

Figures I.21 and I.22 illustrate the effect of affine curvature motion on the values of the curvature of an image. In Figure I.21 the sea bird image has been smoothed by affine curvature motion at calibrated scale 1. In Figure I.22 the smoothing is stronger at calibrated scale 4. (A calibrated scale t means that at this scale a disk with radius t disappears.) The absolute values of the curvature of the smoothed images are shown in the upper-right panels of both figures, with the convention that the darkest points have the largest curvature. For clarity, the curvature is shown only at points where the gradient of the image was larger than 6 in a scale ranging from 0 to 255. Note how the density of points having large curvature is reduced in the second figure where the smoothing is stronger. On the other hand, the regions with large curvature are more concentrated with stronger smoothing. Each degree of smoothing produces a different curvature

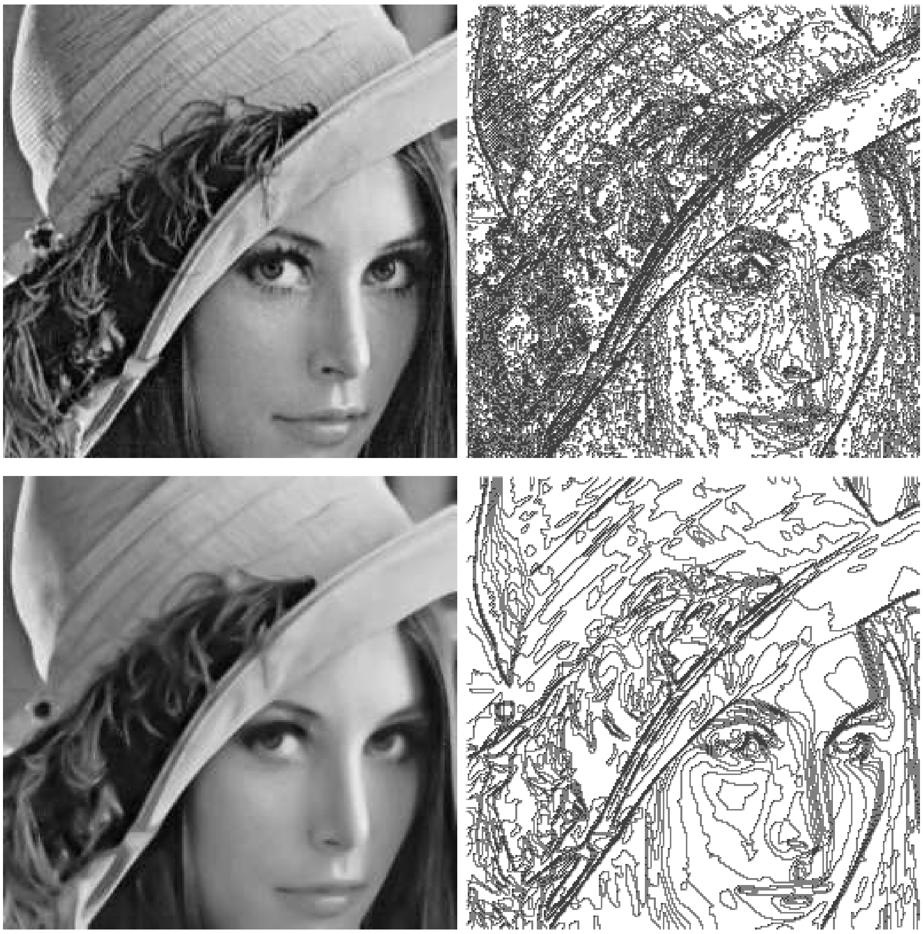

Figure I.20: The affine and morphological scale space (AMSS model). Left to right: original image; level lines of this image (16 levels only); original image smoothed using the AMSS equation; level lines of the third image.

map of the original image, and thus curvature motions can be used as a nonlinear means to compute a ”multiscale” curvature of the original image. The bottom two panels of the figures show, from left to right, the positive curvature and the negative curvature.

The snake method

Before proceeding to shape recognition, we mention that a variant of the curvature equation can be used for shape detection. This is a well-known method of contour detection, initially proposed by Kass, Witkin, and Terzopoulos [125]. Their method was very unstable. A better method is a variant of curvature motion proposed by Caselles, Catté, Coll, and Dibos [44] and improved simultaneously by Caselles, Kimmel, and Sapiro [47] and Malladi, Sethian, and Vemuri [156]. Here is how it works. The user draws roughly the desired contour in

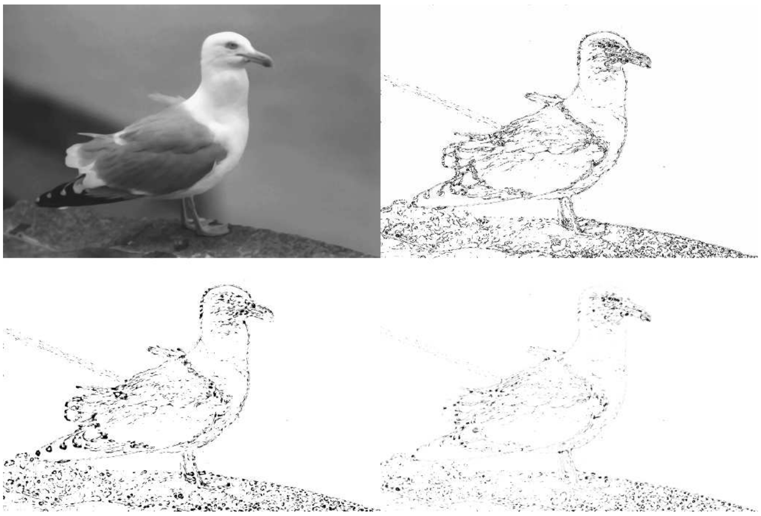

Figure I.21: Curvature scale space I. Top, left to right: original sea bird image smoothed by affine curvature motion at calibrated scale 1; the absolute value of the curvature. Bottom, left to right: the positive part of the curvature; the negative part. Compare with Figure I.22, where the calibrated smoothing scale is 4.

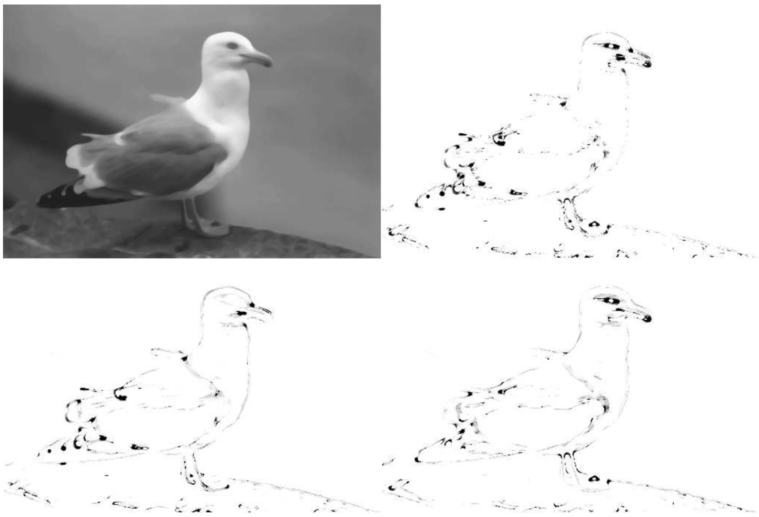

Figure I.22: Curvature scale space II. Top, left to right: original sea bird image smoothed by affine curvature motion at calibrated scale 4; the absolute value of the curvature. Bottom, left to right: the positive part of the curvature; the negative part. Compare with Figure I.21, where the calibrated smoothing scale is 1.

the image, and the algorithm then finds the best possible contour in terms of some variational criterion. This method is very useful in medical imaging. The motion of the contour is a tuned curvature motion that tends to minimize an energy function E. Given an original image u_0 containing some closed contour that we wish to approximate, we start with an edge map

$$g(\mathbf{x}) = \frac{1}{1 + |Du_0(\mathbf{x})|^2},$$

that is, a function that vanishes on the edges of the image. The user then designates the contour of interest by drawing a polygon γ_0 roughly following the desired contour. The *geodesic snake* algorithm then builds a distance function v_0 to this initial contour, so that γ_0 is the zero level set of v_0. The energy to be minimized is

$$E(\gamma) = \int_\gamma g(\mathbf{x}(s))\,\mathrm{d}s,$$

where g is the edge map associated with the original image u_0 and s denotes the parameter measuring the length along γ. The motion of the "analyzing image" v is governed by

$$\frac{\partial v}{\partial t}(\boldsymbol{x}, t) = g(\boldsymbol{x})|Dv(\boldsymbol{x})|\mathrm{curv}(v)(\boldsymbol{x}) - Dv(\boldsymbol{x}) \cdot Dg(\boldsymbol{x}).$$

This algorithm is illustrated with a medical example in Figure I.23.

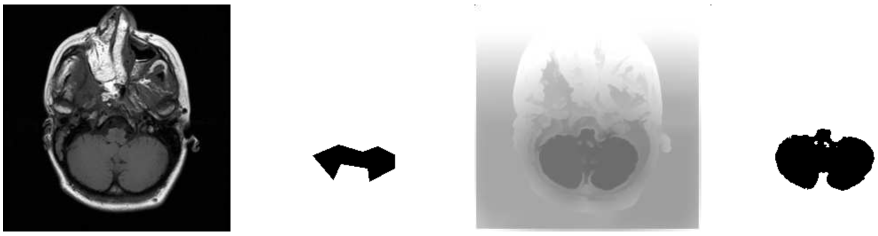

Figure I.23: Active contour, or "snake." Left to right: original image; initial contour; evolved distance function; final contour.

Shape retrieval

It seems to us that the most obvious application of invariant PDEs is shape retrieval in large databases. There are thousands of different definitions of shapes and a multitude of shape recognition algorithms. The real bottleneck has always been the ability to extract the relevant shapes. The discussion above points to a brute force strategy: All contrast-invariant local elements, or the level lines of the image, are candidates to be "shape elements." Of course, this notion of shape element suggests the contours of some object, but there is no way to give a simple geometric definition of objects. We must give up the hope of jumping from the geometry to the common sense world. We may instead simply ask the question, Given two images, can we retrieve all the level lines that are similar in both images? This would give a factual, a posteriori, definition of shapes. They would be defined as pieces of level lines common to two different images, irrespective of their relationships to real physical objects.

Of course, this brute force strategy would be impossible without the initial invariant filtering (AMSS). It is doable only if the level lines have been significantly simplified. This simplification entails the possibility of compressed invariant encoding. In Figure I.24, we present an experiment due to Lisani et al. [149]. Two images of a desk and the backs of chairs, viewed from different angles, are shown in the first two panels. All of the pieces of level lines in the two images that found a match in the other image are shown in the last two panels. Notice that several of these matches are doubled. Indeed, there are two similar chairs in each image. This brings to mind a Gestalt law that states that human perception tends to group similar shapes. We now see the numerical necessity of this perceptual grouping: A preliminary self-matching of each image, with grouping of similar shapes, must be performed before we can compare it with other images.

This concludes our overview of the use of PDEs in image analysis. The rest of the book is devoted to filling in the mathematical details that support most of the results mentioned in this introduction. We have tried to prove all of the mathematical statements, assuming only two or three years of mathematical training at the university level. Thus, for most of the PDEs addressed, and for all of the relevant ones, we prove the existence and uniqueness of solutions. We also develop invariant, monotone approximation schemes. This has been technically possible by combining tools from the recent, and remarkably simple, theory of viscosity solutions with the Matheron formalism for monotone set and function operators. Thus, the really necessary mathematical knowledge amounts to elementary differential calculus, linear algebra, and some results from the theory of Lebesgue integration, which are used in the chapters on the heat equation. Mathematical statements are not introduced as art for art's sake; all of the results are directed at proving the correctness of a model, of its properties, or of the associated numerical schemes. Numerical experiments, with detailed comments, are described throughout the text. They provide an independent development that is parallel to the central theoretical development. Most image processing algorithms mentioned in the text are accessible in the public software MegaWave. MegaWave was developed jointly by several university research groups in France, Spain and America, and it is available at `http://www.cmla.ens-cachan.fr`.

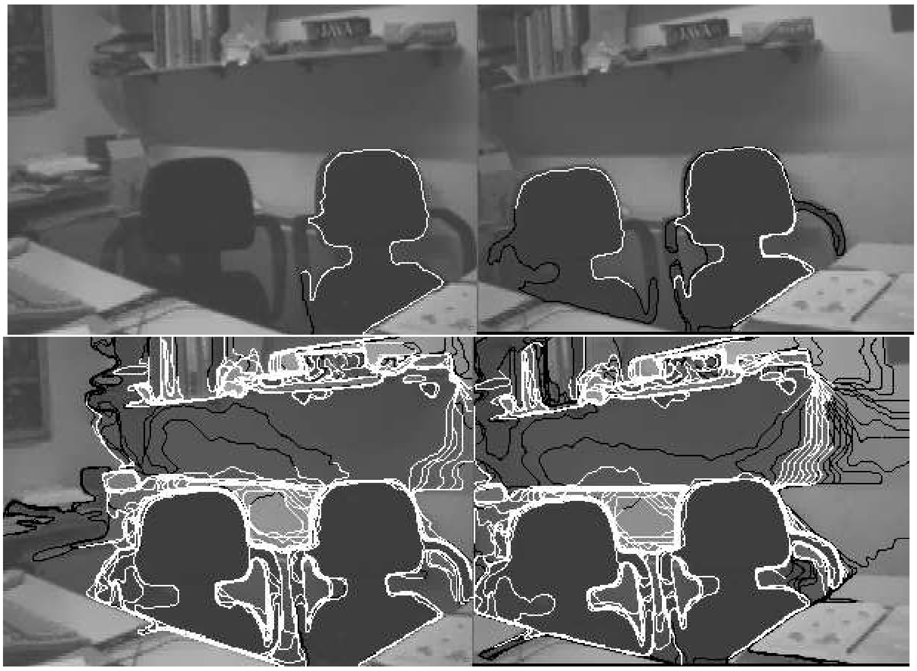

Figure I.24: A shape parser based on level lines. The two left images are of a desk and the backs of chairs viewed from different angles. In the far left panel, one level line has been selected (in white). In the second panel we show, also in white, all matching pieces of level lines. The match is ambiguous, as must be expected when the same object is repeated in the scene. In the two panels on the right, we display all the matching pairs of pieces of level lines (in white). The non matching parts of the same level lines are shown in black. Usually, recognized shape elements are pieces of level lines, seldom whole level lines. See []

Chapter 1

Notation and background material

$\mathbb{R}^N$ denotes the real N-dimensional Euclidian space. If $\mathbf{x} \in \mathbb{R}^N$ and $N > 2$, we write $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$; if $N = 2$, we usually write $\boldsymbol{x} = (x, y)$. For $\boldsymbol{x}$, $\boldsymbol{y} \in \mathbb{R}^N$, we denote their scalar product by $\boldsymbol{x} \cdot \boldsymbol{y} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N$ and write

$$|\boldsymbol{x}| = (\boldsymbol{x} \cdot \boldsymbol{x})^{1/2} = (x_1^2 + x_2^2 + \cdots + x_N^2)^{1/2}.$$

Let Ω be an open set in $\mathbb{R}^N$, and let $n \in \mathbb{N}$ be a fixed integer. $C^n(\Omega)$ denotes the set of real-valued functions $f : \Omega \to \mathbb{R}$ that have bounded continuous derivatives of all orders up to and including n. $f \in C^\infty(\Omega)$ means that f has continuous derivatives of all orders and that they are all bounded; $f \in C(\Omega) = C^0(\Omega)$ means that f is continuous and bounded on Ω. We will often write "f is C^n" as shorthand for $f \in C^n(\Omega)$, and we often omit the domain Ω if there is no chance of confusion.

We use multi-indices of the form $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N) \in \mathbb{N}^N$ as shorthand in several cases. For $\boldsymbol{x} \in \mathbb{R}^N$, we write $\boldsymbol{x}^\alpha$ and $|\boldsymbol{x}|^\alpha$ for $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_N^{\alpha_N}$ and $|x_1|^{\alpha_1} |x_2|^{\alpha_2} \cdots |x_N|^{\alpha_N}$, respectively. For $f \in C^n(\Omega)$, we abbreviate the partial derivatives of f by writing

$$\partial^\alpha f = \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_N^{\alpha_N}},$$

where $|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_N$ and $|\alpha| \leq n$.

We also write the partial derivatives of $f(\boldsymbol{x}) = f(x_1, x_2, \ldots, x_N)$ as $f_i = \partial f / \partial x_i$, $f_{ij} = \partial^2 f / \partial x_i \partial x_j$, and so on. In the two-dimensional case $f(\boldsymbol{x}) = f(x, y)$, we usually write $\partial f / \partial x = f_x$, $\partial f / \partial y = f_y$, $\partial^2 f / \partial x \partial y = f_{xy}$, and so on.

The gradient of f is denoted by Df. Thus, if $f(\boldsymbol{x}) = f(x_1, x_2, \ldots, x_N)$,

$$Df = (f_1, f_2, \ldots, f_N),$$

and

$$Df = (f_x, f_y)$$

in case $N = 2$. The Laplacian of f is denoted by Δf. Thus $\Delta f = f_{11} + f_{22} + \cdots + f_{NN}$ in general, and $\Delta f = f_{xx} + f_{yy}$ if $N = 2$.

We will often use the symbols O, o, and ε. They are defined as follows. We assume that h is a real variable that tends to a limit h_0 that can be finite or infinite. We assume that g is a positive function of h and that f is any other function of h. Then $f = O(g)$ means that there is a constant $C > 0$ such that $|f(h)| < Cg(h)$ for all values of h. The expression $f = o(g)$ means that $f(h)/g(h) \to 0$ as $h \to h_0$. We occasionally will use ε to denote a function of h that tends to zero as $h \to 0$. Thus, $f(h) = o(h)$ can be written equivalently as $f(h) = h\varepsilon(h)$.

Taylor's formula

An N-dimensional form of Taylor's formula is used several times in the book. We will first state it and then explain the notation. Assume that $f \in C^n(\Omega)$ for some open set $\Omega \in \mathbb{R}^N$, that $\boldsymbol{x}, \boldsymbol{y} \in \Omega$, and that the segment joining $\boldsymbol{x}$ and $\boldsymbol{x}+\boldsymbol{y}$ is also in Ω. Then

$$f(\boldsymbol{x}+\boldsymbol{y}) = f(\boldsymbol{x}) + \frac{1}{1!}Df(\boldsymbol{x})\boldsymbol{y}^{(1)} + \frac{1}{2!}D^2 f(\boldsymbol{x})\boldsymbol{y}^{(2)} + \cdots + \frac{1}{n!}D^n f(\boldsymbol{x})\boldsymbol{y}^{(n)} + o(|\boldsymbol{y}|^n).$$

This has been written compactly to resemble the one-dimensional case, but the price to be paid is to explain the meaning of $D^p f(\boldsymbol{x})\boldsymbol{y}^{(p)}$. We have already seen special cases of this expression in section I.3, for example, $D^2u(Du, Du)$ in equation (I.4). The expression $D^p f(\boldsymbol{x})\boldsymbol{y}^{(p)}$ is

$$D^p f(\boldsymbol{x})\boldsymbol{y}^{(p)} = D^p f(\boldsymbol{x})(\underbrace{\boldsymbol{y}, \boldsymbol{y}, \ldots, \boldsymbol{y}}_{p \text{ terms}}) = \sum_{(i_1, i_2, \ldots, i_p)} \frac{\partial^p f}{\partial x_{i_1} \partial x_{i_2} \cdots \partial x_{i_p}}(\mathbf{x}) y_{i_1} y_{i_2} \cdots y_{i_p},$$

where the sum is taken over all N^p different vectors $(i_1, i_2, \ldots, i_p)$, $i_j = 1, 2, \ldots, N$. Notice that $Df(\boldsymbol{x})\boldsymbol{y}^{(1)}$ is just $\sum_{j=1}^N f_j y_j = Df(\boldsymbol{x}) \cdot \boldsymbol{y}$, which is how we usually write it.

The so-called Lagrange variant of Taylor's formula will be equally useful,

$$f(\boldsymbol{x}+\boldsymbol{y}) = f(\boldsymbol{x}) + \cdots + \frac{1}{(n-1)!}D^{n-1} f(\boldsymbol{x})\boldsymbol{y}^{(n-1)} + \frac{1}{n!}D^n f(\boldsymbol{x}+\theta\mathbf{y})\boldsymbol{y}^{(n)}, \quad \text{(I.1)}$$

where $\theta = \theta(\mathbf{y})$ ranges in $[0, 1]$.

The implicit function theorem

Consider a real-valued C^1 function f defined on an open set Ω in $\mathbb{R}^N$. For ease of notation we write $\mathbf{z} = (\mathbf{x}, y)$, where $\mathbf{x} = (x_1, \ldots, x_{N-1})$ and $y = x_N$. Assume that $f(\mathbf{z}_0) = 0$ for a point $\mathbf{z}_0 \in \Omega$ and that $f_y(\mathbf{x}_0) \neq 0$. Then there is a neighborhood $M = M(\mathbf{x}_0)$ and a neighborhood $N = N(y_0)$ such that for every $\mathbf{x} \in M$ there is exactly one $y \in N$ such that $f(\mathbf{x}, y) = 0$. The function $y = \varphi(\mathbf{x})$ is C^1 on M and $y_0 = \varphi(\mathbf{x}_0)$. Furthermore, if $f \in C^n(\Omega)$, then $\varphi \in C^n(M)$.

Lebesgue integration

The Lebesgue integral, which first appeared in 1901 and is thus over a hundred years old, has become the workhorse of analysis. It plays a role in chapters 1 and 2 and appears briefly in other parts of the book. One does not need a profound understanding of abstract measure theory and integration to follow the arguments. One should, however, be familiar with a few key results and be comfortable with the basic manipulations of the integral. With this in mind, we restate some of these fundamentals.

The functions and sets in this book are always measurable. Thus we dispense in general with phrases like “let f be a measurable function.” We denote by $\mathcal{M}$ the set of Lebesgue measurable subsets of $\mathbb{R}^N$. Since we shall sometimes need to complete $\mathbb{R}^N$ by a point at infinity, ∞, we still denote by $\mathcal{M}$ the measurable sets of $S_N = \mathbb{R}^N \cup \{\infty\}$ and take measure$(\{\infty\}) = 0$. A function f defined on a subset A of $\mathbb{R}^N$ is integrable, if

$$\int_A |f(\boldsymbol{x})| \,\mathrm{d}\boldsymbol{x} < +\infty.$$

The Banach space of all integrable function defined on A is denoted as usual by $L^1(A)$; we write $\|f\|_{L^1(A)} = \int_A |f(\boldsymbol{x})| \,\mathrm{dx}$ to denote the norm of f in $L^1(A)$. The most important applications in the book are the two cases $A = \mathbb{R}^N$ and $A = [-1,1]^N$. Here are two results that we use in chapters 1 and 2. We state them not in the most general form, but rather in the simplest form suitable for our work.

A density theorem for $L^1(\mathbb{R}^N)$

If f is in $L^1(\mathbb{R}^N)$, then there exists a sequence of continuous functions $\{g_n\}$, each of which has compact support, such that $g_n \to f$ in $L^1(\mathbb{R}^N)$, that is, $\int |g_n(\mathbf{x}) - f(\mathbf{x})| d\mathbf{x} \to 0$ as $n \to +\infty$. This result is true for $L^1([-1,1]^N)$, in which case the g_n are continuous on $[-1,1]^N$.

Fubini’s theorem

Suppose that f is a measurable function defined on $A \times B \in \mathbb{R}^N \times \mathbb{R}^N$. Fubini’s theorem states that

$$\int_{A\times B} |f(\mathbf{z})| \,\mathrm{d}\boldsymbol{z} = \int_A \int_B |f(\mathbf{x},\mathbf{y})| \,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{y} = \int_B \int_A |f(\mathbf{x},\mathbf{y})| \,\mathrm{d}\mathbf{y}\,\mathrm{d}\mathbf{x},$$

where we have written $\mathbf{z} = (\mathbf{x},\mathbf{y})$. It further states, that if any one of the integrals is finite, then

$$\int_{A\times B} f(\mathbf{z}) \,\mathrm{d}\boldsymbol{z} = \int_B \int_A f(\mathbf{x},\mathbf{y}) \,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{y} = \int_A \int_B f(\mathbf{x},\mathbf{y}) \,\mathrm{d}\mathbf{y}\,\mathrm{d}\mathbf{x}.$$

Lebesgue’s dominated convergence theorem

If a sequence of functions $\{f_n\}$ is such that $f_n(\mathbf{x}) \to f(\mathbf{x})$ for almost every $\mathbf{x} \in \mathbb{R}^N$ as $n \to +\infty$, and if there is an integrable function g such that $|f_n(\mathbf{x})| \leq g(\mathbf{x})$ almost everywhere, then

$$\int_{\mathbb{R}^N} f_n(\mathbf{x}) \,\mathrm{d}\mathbf{x} \to \int_{\mathbb{R}^N} f(\mathbf{x}) \,\mathrm{d}\mathbf{x}.$$

We often use the following direct consequence: if A_n is a decreasing sequence of measurable sets with bounded measure then $\mathrm{measure(A_n)} \mapsto \mathrm{measure(A)}$. To prove this, apply Lebesgue's theorem to the characteristic functions of A_n and A, $\mathbf{1}_{A_n}$ and $\mathbf{1}_A$.

We also use the following result, which is a direct consequence of the dominated convergence theorem.

Interchanging differentiation and integration

Suppose that a function f defined on $(t_0, t_1) \times \mathbb{R}^N$, where (t_0, t_1) is any interval of $\mathbb{R}$, is such that $t \mapsto f(t, \mathbf{x})$ is continuously differentiable (for almost every $\mathbf{x} \in \mathbb{R}^N$) on some interval $[a, b] \subset (t_0, t_1)$. If there exists an integrable function g such that for all $t \in [a, b]$

$$\left| \frac{\partial f}{\partial t}(t, \mathbf{x}) \right| \leq g(\mathbf{x}) \quad \text{almost everywhere,}$$

then the integral $I(t) = \int_{\mathbb{R}^N} f(t, \mathbf{x})\, \mathrm{d}\mathbf{x}$ is differentiable for $t \in (a, b)$ and

$$\frac{dI}{dt}(t) = \int_{\mathbb{R}^N} \frac{\partial f}{\partial t}(t, \mathbf{x})\, \mathrm{d}\mathbf{x}.$$

A brief but comprehensive discussion of the Lebesgue integral can be found in the classic textbook by Walter Rudin [216].

I.0.1 A framework for sets and images

We start by fixing a simple and handy functional framework for images and sets, which will be maintained throughout the book. Until now, we have been vague about the domain of definition of an image. On one hand, a real digital image is defined on a finite grid. On the other hand, standard interpolation methods give a continuous representation defined on a finite domain of $\mathbb{R}^N$, usually a rectangle. Now, it is convenient to have images defined on all of $\mathbb{R}^N$, but it is not convenient to extend them by making them zero outside their original domains of definition because that would make them discontinuous. So an usual way is to extend them into a continuous function tending to a constant at infinity. One way to do that is illustrated in Figure I.1. First, an extension to a wider domain is performed by reflection across the domain's boundary and periodization. Then, it is easy to let the function fade at infinity or to make it compactly supported. This also means that we fix a value at infinity for u, which we denote by $u(\infty)$. We denote the topological completion of $\mathbb{R}^N$ by this infinity point by $S_N = \mathbb{R}^N \cup \{\infty\}$, which can also be denoted $\overline{\mathbb{R}^N}$. Let us justify the notation.

Proposition 1.1. *Consider the sphere $S_N = \{\mathbf{z} \in \mathbb{R}^{N+1},\ ||\mathbf{z}|| = 1\}$. Then the mapping $T:\ \mathbb{R}^N \cup \{\infty\} \to S_N$ defined by*

$$T(\mathbf{x}) = \left(\frac{2\mathbf{x}}{1 + \mathbf{x}^2},\ \frac{\mathbf{x}^2 - 1}{\mathbf{x}^2 + 1} \right)$$

is a homeomorphism (that is, a continuous bijection with continuous inverse.)

This is easily checked (Exercise 2.6).

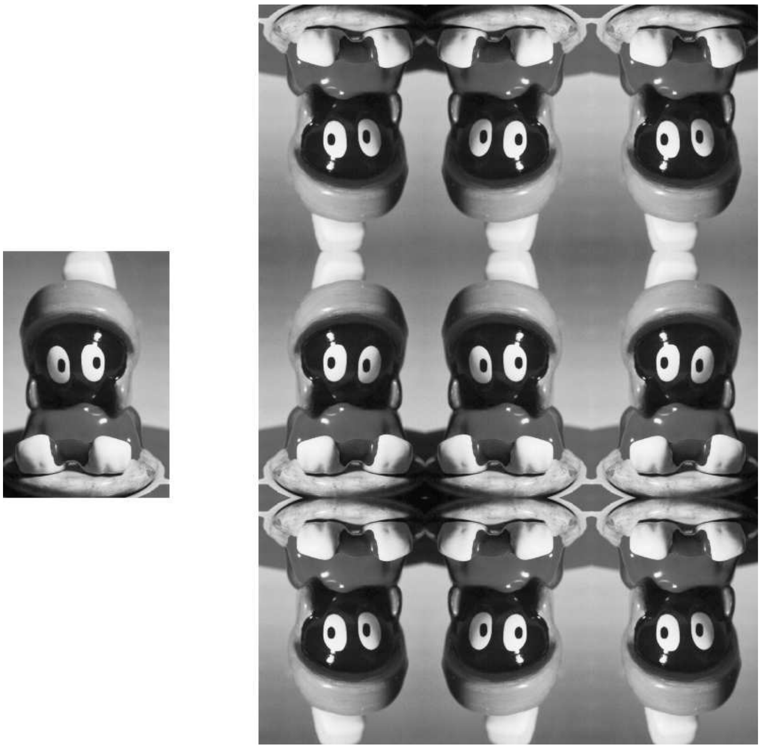

Figure I.1: Image extension by symmetry, followed by periodization. Then the image can be extended continuously to the rest of the plane into a function which is constant for $\mathbf{x}$ large. The purpose of these successive extensions of u to all of $\mathbb{R}^N$ is to facilitate the definition of certain operations on u, such as convolution with smoothing kernels, and, at the same time, to preserve the continuity of u. This method of extending a function is widely used in image processing; in particular, it is used in most compression and transmission standards. For instance, the discrete cosine transform (DCT) applied to the initial data u, restricted to $[0,1]^N$, is easily interpreted as an application of the FFT to the symmetric extension of u.

Definition 1.2. *We denote by $\mathcal{F}$ the set of continuous functions on S_N, which can be identified with the set of continuous functions on $\mathbb{R}^N$ tending to some constant at infinity. The natural norm of $\mathcal{F}$ is*

$$\|u\|_{\mathcal{F}} = \sup_{\mathbf{x}\in\mathbb{R}^N} |u(\mathbf{x})|. \tag{I.2}$$

We say that an image u in $\mathcal{F}$ is C^1, if the function u is C^1 at each point $\mathbf{x} \in \mathbb{R}^N$. We define in the same way the C^2,... C^∞ functions of $\mathcal{F}$.

Definition 1.3. *We say that a function u defined on $\mathbb{R}^N$ is uniformly continuous if for every $\mathbf{x}$, $\mathbf{y}$,*

$$|u(\mathbf{x}+\mathbf{y}) - u(\mathbf{x})| \leq \varepsilon(|\mathbf{y}|),$$

for some function ε *called modulus of continuity of* u*, satisfying* $\lim_{s\to 0} \varepsilon(s) = 0$.

Continuous functions on a compact set are uniformly continuous, so functions of $\mathcal{F}$ are uniformly continuous. We shall often consider the level sets of functions in $\mathcal{F}$, which simply are compact sets of S_N.

Definition 1.4. *We denote by* $\mathcal{L}$ *the set of all compact sets of* S_N*.*

These sets are easy to characterize:

Proposition 1.5. *The elements of* $\mathcal{L}$ *are of three kinds:*

- *compact subsets of* $\mathbb{R}^N$
- $F \cup \{\infty\}$*, where* F *is a compact set of* $\mathbb{R}^N$*.*
- $F \cup \{\infty\}$*, where* F *is an unbounded closed subset of* $\mathbb{R}^N$

Proof. Indeed, $B \cap \mathbb{R}^N$ is a closed set of $\mathbb{R}^N$ and is therefore either a bounded compact set or an unbounded closed set of $\mathbb{R}^N$. In the latter case, B must contain ∞. □

Part I

Linear Image Analysis

Chapter 2

The Heat Equation

The heat equation is the prototype of all the PDEs used in image analysis. There are strong reasons for that and it is the aim of this chapter to explain some of them. Some more will be given in Chapter **??**. Our first section is dedicated to a simple example of linear smoothing illustrating the relation between linear smoothing and the Laplacian. In the next section, we prove the existence and uniqueness of its solutions, which incidentally establishes the equivalence between the convolution with a Gaussian and the heat equation.

2.1 Linear smoothing and the Laplacian

Consider a continuous and bounded function u_0 defined on $\mathbb{R}^2$. If we wish to smooth u_0, then the simplest way to do so without favoring a particular direction is to replace $u_0(\mathbf{x})$ with the average of the values of u_0 in a disk $D(\mathbf{x}, h)$ of radius h centered at $\mathbf{x}$. This means that we replace $u_0(\mathbf{x})$ with

$$M_h u_0(\mathbf{x}) = \frac{1}{\pi h^2} \int_{D(\mathbf{x},h)} u_0(\mathbf{y})\,\mathrm{d}\mathbf{y} = \frac{1}{\pi h^2} \int_{D(0,h)} u_0(\mathbf{x}+\mathbf{y})\,\mathrm{d}\mathbf{y}. \tag{2.1}$$

Although the operator M_h is quite simple, it exhibits important characteristics of a general linear isotropic smoothing operator. For example, it is localizable: As h becomes small, M_h becomes more localized, that is, $M_h u_0(\mathbf{x})$ depends only on the values of $u_0(\mathbf{x})$ in a small neighborhood of $\mathbf{x}$. Smoothing an image by averaging over a small symmetric area is illustrated in Figure 2.1.

Our objective is to point out the relation between the action of M_h and the action of the Laplacian, or the heat equation. To do so, we assume enough regularity for u_0, namely that it is C^2. We shall actually prove in Theorem 3.2 that under that condition

$$M_h u_0(\mathbf{x}) = u_0(\mathbf{x}) + \frac{h^2}{8}\Delta u_0(\mathbf{x}) + h^2 \varepsilon(\mathbf{x}, h), \tag{2.2}$$

where $\varepsilon(\mathbf{x}, h)$ tends to 0 when $h \to 0$. As we have seen in the introduction, (2.2) provides the theoretical basis for deblurring an image by subtracting a small amount of its Laplacian. It also suggests that M_h acts as one step forward in the heat equation starting with initial condition u_0,

$$\frac{\partial u}{\partial t}(t, \mathbf{x}) = \frac{1}{8}\Delta u(t, \mathbf{x}), \quad u(0, \mathbf{x}) = u_0(\mathbf{x}). \tag{2.3}$$

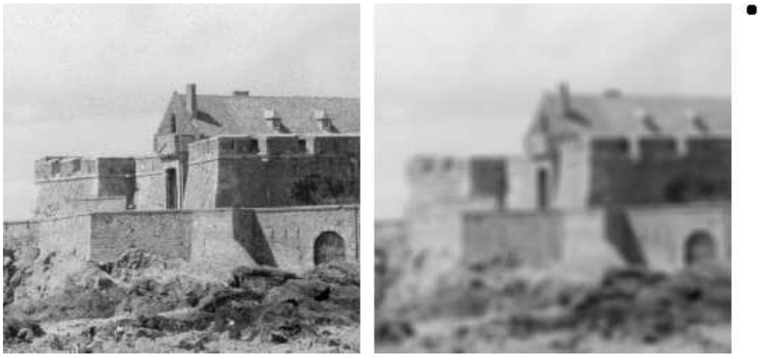

Figure 2.1: Local averaging algorithm. Left to right: original image; result of replacing the grey level at each pixel by the average of the grey levels over the neighboring pixels. The shape of the neighborhood is shown by the black spot displayed in the upper right-hand corner.

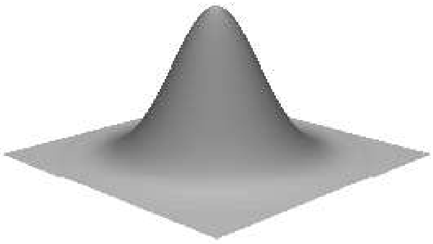

Figure 2.2: The Gaussian in two dimensions.

This statement is made more precise in Exercise 2.5. Equation (2.2) actually suggests that if we let $n \to +\infty$ and at the same time require that $nh^2 \to t$, then

$$(M_h^n u_0)(\mathbf{x}) \to u(t, \mathbf{x}) \tag{2.4}$$

where $u(t, x)$ is a solution of (2.3).

This heuristics justifies the need for a thorough analysis of the heat equation. The next chapter will prove that (2.4) is true under fairly general conditions. In the next section, we shall prove that the heat equation has a unique solution for a given continuous initial condition u_0, and that this solution at time t is equal to the convolution $G_t * u_0$, where G_t is the Gaussian (Figure 2.2). The effect on level lines of smoothing with the Gaussian is shown in Figure 2.4.

2.2 Existence and uniqueness of solutions of the heat equation

Definition 2.1. *We say that a function g defined on $\mathbb{R}^N$ is* rapidly decreasing*,* *or* has fast decay*, if for each multi-index β there is a constant C such that*

$$|\mathbf{x}|^\beta |g(\mathbf{x})| \leq C.$$

We say that g belongs to the Schwartz class $\mathcal{S}$ if $g \in C^\infty(\mathbb{R}^N)$ and if $\partial^\alpha g$ has fast decay for each multi-index α.

Proposition 2.2. *If $g \in \mathcal{S}$, then $g \in L^1(\mathbb{R}^N)$, that is, $\int_{\mathbb{R}^N} |g(\mathbf{x})| d\mathbf{x} < +\infty$. For each pair of multi-indices α, β, the function $\mathbf{x}^\beta \partial^\alpha g$ also belongs to $\mathcal{S}$, and $\partial^\alpha g$ is uniformly continuous on $\mathbb{R}^N$.*

Proof. The second statement follows from the Leibnitz rule for differentiating a product. By the definition of $\mathcal{S}$, there is a constant C such that $|\mathbf{x}|^{N+2}|g(\mathbf{x})| \leq C$. Thus there is another C such that $|g(\mathbf{x})| \leq C/(1+|\mathbf{x}|^{N+2})$; since $C/(1+|\mathbf{x}|^{N+2}) \in L^1(\mathbb{R}^N)$, $g \in L^1(\mathbb{R}^N)$. Finally, note that $|\partial^\alpha g(\mathbf{x})| \to 0$ as $|\mathbf{x}| \to \infty$. But any continuous function on $\mathbb{R}^N$ that tends to zero at infinity is uniformly continuous. □

Proposition 2.3 (The Gaussian and the heat equation). *For all $t > 0$, the function $\mathbf{x} \mapsto G_t(\mathbf{x}) = (1/(4\pi t)^{N/2}) \mathrm{e}^{-|\mathbf{x}|^2/4t}$ belongs to $\mathcal{S}$ and satisfies the heat equation*

$$\frac{\partial G_t}{\partial t} - \Delta G_t = 0.$$

Proof. It is sufficient to prove the first statement for the function $g(\mathbf{x}) = \mathrm{e}^{-|\mathbf{x}|^2}$. An induction argument shows that $\partial^\alpha g(\mathbf{x}) = P_\alpha(\mathbf{x})\mathrm{e}^{-|\mathbf{x}|^2}$, where $P_\alpha(\mathbf{x})$ is a polynomial of degree $|\alpha|$ in the variables $x_1, x_2, \dots, x_N$. The fact that, for every $k \in \mathbb{N}$, $x^k \mathrm{e}^{-x^2} \to 0$ as $|x| \to +\infty$ finishes the proof. Differentiation shows that G_t satisfies the heat equation. □

Exercise 2.1. Check that G_t is solution of the heat equation. ■

Linear image filtering is mainly done by convolving an image u with a positive integrable kernel g. This means that the smoothed image is given by the function $g * u$ defined as

$$g * u(\mathbf{x}) = \int_{\mathbb{R}^N} g(\mathbf{x}-\mathbf{y})u(\mathbf{y})\, d\mathbf{y} = \int_{\mathbb{R}^N} g(\mathbf{y})u(\mathbf{x}-\mathbf{y})\, d\mathbf{y}.$$

Exercise 2.2. Prove that the convolution, when it makes sense, is translation invariant. This means that $g * u(\mathbf{x}-\mathbf{z}) = g_{\mathbf{z}} * u(\mathbf{x})$, where $g_{\mathbf{z}}(\mathbf{x}) = g(\mathbf{x}-\mathbf{z})$. ■

Exercise 2.3. Check that $G_t * G_s = G_{t+s}$. ■

Linear filtering with the Gaussian at several scales is illustrated in Figure 2.3. The next result establishes properties of the convolution that we need for our treatment of the heat equation.

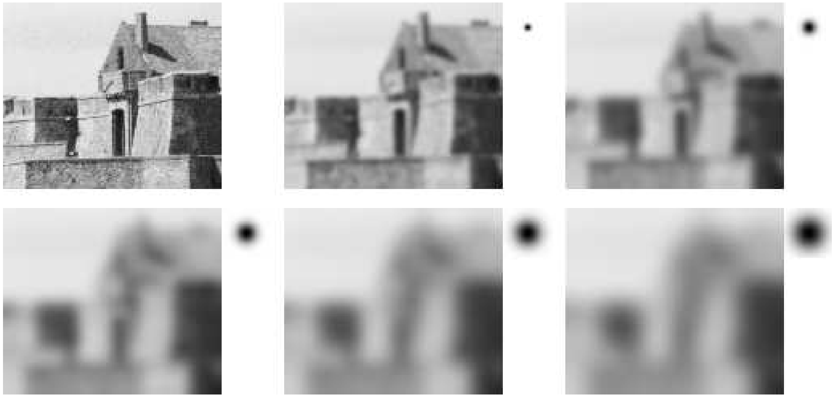

Figure 2.3: Convolution with Gaussian kernels (heat equation). Displayed from top-left to bottom-right are the original image and the results of convolutions with Gaussians of increasing variance. A grey level representation of the convolution kernel is put on the right of each convolved image to give an idea of the size of the involved neighborhood.

Proposition 2.4. *Assume that $u \in \mathcal{F}$ and that $g \in L^1(\mathbb{R}^N)$. Then the function $g * u$ belongs to $\mathcal{F}$ and satisfies the inequality*

$$\|g * u\|_{\mathcal{F}} \leq \|g\|_{L^1(\mathbb{R}^N)} \|u\|_{\mathcal{F}}. \tag{2.5}$$

Proof.

$$|g * u(\mathbf{x})| \leq \int_{\mathbb{R}^N} |g(\mathbf{x}-\mathbf{y})||u(\mathbf{y})| \, \mathrm{d}\mathbf{y} \leq \|u\|_{\mathcal{F}} \int_{\mathbb{R}^N} |g(\mathbf{x}-\mathbf{y})| \, \mathrm{d}\mathbf{y} = \|u\|_{\mathcal{F}} \|g\|_{L^1(\mathbb{R}^N)}.$$

□

Exercise 2.4. Verify that $g * u$ indeed is continuous and tends to $u(\infty)$ at infinity : this a direct application of Lebesgue Theorem. ■

We are now going to focus on kernels that, like the Gaussian, belong to $\mathcal{S}$.

Proposition 2.5. *If $u \in \mathcal{F}$ and $g \in \mathcal{S}$, then $g * u \in C^\infty(\mathbb{R}^N) \cap \mathcal{F}$ and*

$$\partial^\alpha (g * u) = (\partial^\alpha g) * u \tag{2.6}$$

for every multi-index α.

Proof. Since $g \in \mathcal{S}$, g is in $L^1(\mathbb{R}^N)$, and so is $\partial^\alpha g$ for any multi-index α (Proposition 2.2). Thus by Proposition 2.4, $(\partial^\alpha g) * u$ belongs to $\mathcal{F}$. To prove (2.6), it is sufficient to prove it for $\alpha = (1, 0, \dots, 0)$. Indeed, we know that $\partial^\alpha g$ is in $\mathcal{S}$ if g is in S, so the general case follows from the case $\alpha = (1, 0, \dots, 0)$ by induction. Letting $\boldsymbol{e}_1 = (1, 0, \dots, 0)$ and using Taylor's formula with Lagrange's form for the remainder, we can write

$$\begin{aligned} g * u(\mathbf{x} + h\boldsymbol{e}_1) - g * u(\mathbf{x}) &= \int_{\mathbb{R}^N} (g(\mathbf{x} + h\boldsymbol{e}_1 - \mathbf{y}) - g(\mathbf{x} - \mathbf{y}))u(\mathbf{y})\,\mathrm{d}\mathbf{y} \\ &= \int_{\mathbb{R}^N} (g(\mathbf{y} + h\boldsymbol{e}_1) - g(\mathbf{y}))u(\mathbf{x} - \mathbf{y})\,\mathrm{d}\mathbf{y} \\ &= h\int_{\mathbb{R}^N} \frac{\partial g}{\partial x_1}(\mathbf{y})u(\mathbf{x} - \mathbf{y})\,\mathrm{d}\mathbf{y} \\ &\quad + \frac{h^2}{2}\int_{\mathbb{R}^N} \frac{\partial^2 g}{\partial x_1^2}(\mathbf{y} + \theta(\mathbf{y})h\boldsymbol{e}_1)u(\mathbf{x} - \mathbf{y})\,\mathrm{d}\mathbf{y}, \end{aligned} \tag{2.7}$$

where $0 \leq \theta(\mathbf{y}) \leq 1$. To complete the proof, we wish to have a bound on the last integral that is independent of $\mathbf{x} \in C$. This last integral is of the form $f * u$, where f is defined by $f(\mathbf{y}) = (\partial^2 g/\partial x_1^2)(\mathbf{y}+\theta(\mathbf{y})h\boldsymbol{e}_1)$. Since $g \in \mathcal{S}$, $\partial^2 g/\partial x_1^2 \in \mathcal{S}$, and from this it is a simple computation to show that f is bounded and decays fast at infinity. Having done this, Proposition 2.4 applies, and we deduce that $g * u$ is differentiable in x_1 and that $\partial(g * u)/\partial x_1 = (\partial g/\partial x_1) * u$. □

Proposition 2.6. *Assume that g decreases rapidly at infinity, that $g(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^N$, and that $\int_{\mathbb{R}^N} g(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1$ and set, for $t > 0$, $g_t(\mathbf{x}) = (1/t^N)g(\mathbf{x}/t)$. Then: If $u_0 \in \mathcal{F}$, $g_t * u_0$ converges to u_0 uniformly as $t \to 0$. In addition, we have a maximum principle :*

$$\inf_{\mathbf{x} \in C} u_0(\mathbf{x}) \leq g_t * u_0(\mathbf{x}) \leq \sup_{\mathbf{x} \in C} u_0(\mathbf{x}). \tag{2.8}$$

Proof. Note first that g_t is normalized so that

$$\int_{\mathbb{R}^N} g_t(\mathbf{y})\,\mathrm{d}\mathbf{y} = 1. \tag{2.9}$$

A change of variable $\mathbf{x} \to t\mathbf{x}$ and an application of Lebesgue's theorem shows that, for any $\eta > 0$,

$$\int_{|\mathbf{y}| \geq \eta} g_t(\mathbf{y})\,\mathrm{d}\mathbf{y} \to 0 \text{ as } t \to 0. \tag{2.10}$$

Using (2.9), we have

$$g_t * u_0(\mathbf{x}) - u_0(\mathbf{x}) = \int_{\mathbb{R}^N} g_t(\mathbf{y})(u_0(\mathbf{x} - \mathbf{y}) - u_0(\mathbf{x}))\,\mathrm{d}\mathbf{y}. \tag{2.11}$$

As already mentioned, $u_0 \in \mathcal{F}$ is uniformly continuous. Thus, for any $\varepsilon > 0$, there is an $\eta = \eta(\varepsilon) > 0$ such that $|u_0(\mathbf{x} - \mathbf{y}) - u_0(\mathbf{x})| \leq \varepsilon$ when $|\mathbf{y}| \leq \eta$. Using this inequality, we have

$$\begin{aligned} |g_t * u_0(\mathbf{x}) - u_0(\mathbf{x})| \leq &\int_{|\mathbf{y}| < \eta} g_t(\mathbf{y})|u_0(\mathbf{x} - \mathbf{y}) - u_0(\mathbf{x})|\,\mathrm{d}\mathbf{y} \\ &+ \int_{|\mathbf{y}| \geq \eta} g_t(\mathbf{y})|u_0(\mathbf{x} - \mathbf{y}) - u_0(\mathbf{x})|\,\mathrm{d}\mathbf{y} \\ \leq &\varepsilon \int_{|\mathbf{y}| < \eta} g_t(\mathbf{y})\,\mathrm{d}\mathbf{y} + 2\|u\|_{L^\infty(C)} \int_{|\mathbf{y}| \geq \eta} g_t(\mathbf{y})\,\mathrm{d}\mathbf{y}. \end{aligned}$$

By (2.9) and (2.10), we conclude that $g_t * u$ tends to u_0 uniformly in $\mathbf{x}$ as $t \to 0$. Relation (2.8) is an immediate consequence of the assumption that $g_t(\mathbf{x}) \geq 0$ and equation (2.9). □

Lemma 2.7. *Let $u_0 \in \mathcal{F}$ and $u(t,\mathbf{x}) = (G_t * u_0)(\mathbf{x})$. Then for every $t_0 > 0$, $u(t,\mathbf{x}) \to u_0(\infty)$ uniformly for $t \leq t_0$ as $\mathbf{x} \to \infty$.*

Proof. By assumption,

$$\forall \varepsilon > 0,\ \exists R,\ |\mathbf{x}| \geq R \Rightarrow |u_0(\mathbf{x}) - u_0(\infty)| < \varepsilon. \tag{2.12}$$

Because of the fast decay of the gaussian at infinity (or using Lebesgue's theorem, as in the former proof), we have

$$\forall \varepsilon > 0,\ \exists r(\varepsilon),\ r \geq r(\varepsilon) \Rightarrow \int_{|\mathbf{y}| \geq r} G_{t_0}(\mathbf{y}) d\mathbf{y} < \varepsilon. \tag{2.13}$$

By using $\int G_t(\mathbf{y}) d\mathbf{y} = 1$, we deduce that

$$|u(t,\mathbf{x}) - u(\infty)| \leq \int_{|\mathbf{y}| \leq r} G_t(\mathbf{y}) |u_0(\mathbf{x}-\mathbf{y}) - u_0(\infty)| d\mathbf{y} + \int_{|\mathbf{y}| \geq r} G_t(\mathbf{y}) |u_0(\mathbf{x}-\mathbf{y}) - u_0(\infty)| d\mathbf{y}. \tag{2.14}$$

Using (2.13), the second term in (2.14) is bound from above for $r \geq r(\varepsilon)$ and $t \leq t_0$ by

$$(2 \sup |u_0|) \int_{|\mathbf{y}| \geq r} G_{t_0}(\mathbf{y}) \leq (2 \sup |u_0|) \varepsilon.$$

Fix therefore $r \geq r(\varepsilon)$. Then using $\int G_t = 1$, the first term in (2.14) is bound by ε by (2.12) for $|\mathbf{x}| \geq R + r$.

□

Lemma 2.8. *Let $u_0 \in \mathcal{F}$ and G_t the gaussian. Then*

$$(\partial G_t / \partial t) * u_0 = \partial (G_t * u_0) / \partial t.$$

Proof. Proposition 2.5 does not apply directly, since it applies to the spatial partial derivatives of G_t but not to the derivative with respect to t. Observe, however, that a slight modification of the proof of this proposition does the job: Replace g with G_t and $\mathbf{x}_1$ with t. Then the crux of the matter is to notice that, given an interval $0 < t_0 < t_1$, there is a rapidly decreasing function f such that $|(\partial^2 G_t / \partial t^2)(t + \theta(t)h, \mathbf{y})| \leq f(\mathbf{y})$ uniformly for $t \in [t_0, t_1]$, where f depends on t_0 and t_1 but not on h. Then Proposition 2.4 applies, and the last integral in equation (2.7) is uniformly bounded. □

All of the tools are in place to state and prove the main theorem of this chapter.

Theorem 2.9 (Existence and uniqueness of solutions of the heat equation). *Assume that $u_0 \in \mathcal{F}$ and define for $t > 0$ and $\mathbf{x} \in \mathbb{R}^N$, $u(t,\mathbf{x}) = (G_t * u_0)(\mathbf{x})$, $u(t,\infty) = u_0(\infty)$ and $u(0,\mathbf{x}) = u_0(\mathbf{x})$. Then*

(i) *u is C^∞ and bounded on $(0, +\infty) \times \mathbb{R}^N$;*

(ii) *$\mathbf{x} \to u(t, \mathbf{x})$ belongs to $\mathcal{F}$ for every $t \geq 0$;*

(iii) *for any $t_0 \geq 0$, $u(t, \mathbf{x})$ tends uniformly for $t \leq t_0$ to $u(\infty)$ as $\mathbf{x} \to \infty$;*

(iv) *$u(t, \mathbf{x})$ tends uniformly to $u_0(\mathbf{x})$ as $t \to 0$;*

(v) *$u(t, \mathbf{x})$ satisfies the heat equation with initial value u_0;*

$$\frac{\partial u}{\partial t} = \Delta u \quad \textit{and} \quad u(0, \mathbf{x}) = u_0(\mathbf{x}); \tag{2.15}$$

(vi) *More specifically,*

$$\sup_{\mathbf{x} \in \mathbb{R}^N,\, t \geq 0} |u(t, \mathbf{x})| \leq \|u_0\|_{\mathcal{F}}. \tag{2.16}$$

Conversely, given $u_0 \in \mathcal{F}$, $u(t, \mathbf{x}) = (G_t * u_0)(\mathbf{x})$ is the only C^2 bounded solution u of (2.15) that satisfies properties (ii)-(v).

Proof. Let us prove properties (i)-(vi). For each $t > 0$, $G_t \in \mathcal{S}$, so by Proposition 2.5 and Lemma 2.8,

$$\frac{\partial u}{\partial t} - \Delta u = u * \left(\frac{\partial G_t}{\partial t} - \Delta G_t \right). \tag{2.17}$$

Proposition 2.5 also tells us that $u(t, \cdot) \in C^\infty(\mathbb{R}^N) \cap \mathcal{F}$ for each $t > 0$. The right-hand side of (2.17) is zero by Proposition 2.3, and the fact that $|u(t, \mathbf{x}) - u_0(\mathbf{x})| \to 0$ uniformly as $t \to 0$ follows from Proposition 2.6. The inequality (2.16) is a direct application of Proposition 2.4. Relation (iii) comes from Lemma 2.7.

Uniqueness proof. If both v and w are solutions of the heat equation with the same initial condition $u_0 \in \mathcal{F}$, then $u = v - w$ is in $\mathcal{F}$ and satisfies (2.15) with the initial condition $u_0(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathbb{R}^N$. Also, by the assumptions of (ii), u is bounded on $[0, +\infty) \times \mathbb{R}^N$ and is C^2 on $(0, +\infty) \times \mathbb{R}^N$. We wish to show that $u(t, \mathbf{x}) = 0$ for all $t > 0$ and all $\mathbf{x} \in \mathbb{R}^N$. Assume that this is not the case. Then there is some point $(t, \mathbf{x})$ where $u(t, \mathbf{x}) \neq 0$. Assume that $u(t, \mathbf{x}) > 0$, by changing u to $-u$ if necessary.

We now consider the function u^ε defined by $u^\varepsilon(t, \mathbf{x}) = \mathrm{e}^{-\varepsilon t} u(t, \mathbf{x})$. This function tends to zero uniformly in $\mathbf{x}$ as $t \to 0$ and as $t \to +\infty$. It also tends uniformly to zero for each $t \leq t_0$ when $\mathbf{x} \to \infty$. These conditions imply that u^ε attains its supremum at some point $(t_0, \mathbf{x}_0) \in (0, +\infty) \times \mathbb{R}^N$, and this means that $\Delta u^\varepsilon(t_0, \mathbf{x}_0) = \mathrm{e}^{-\varepsilon t} \Delta u(t_0, \mathbf{x}_0) \leq 0$ and $(\partial u^\varepsilon / \partial t)(t_0, \mathbf{x}_0) = 0$. Here is the payoff: Using the fact that u is a solution of the heat equation, we have the following relations:

$$\begin{aligned} 0 = \frac{\partial u^\varepsilon}{\partial t}(t_0, \mathbf{x}_0) &= -\varepsilon u^\varepsilon(t_0, \mathbf{x}_0) + \mathrm{e}^{-\varepsilon t} \frac{\partial u}{\partial t}(t_0, \mathbf{x}_0) \\ &= -\varepsilon u^\varepsilon(t_0, \mathbf{x}_0) + \mathrm{e}^{-\varepsilon t} \Delta u(t_0, \mathbf{x}_0) \leq -\varepsilon u^\varepsilon(t_0, \mathbf{x}_0) < 0. \end{aligned}$$

This contradiction completes the uniqueness proof. □

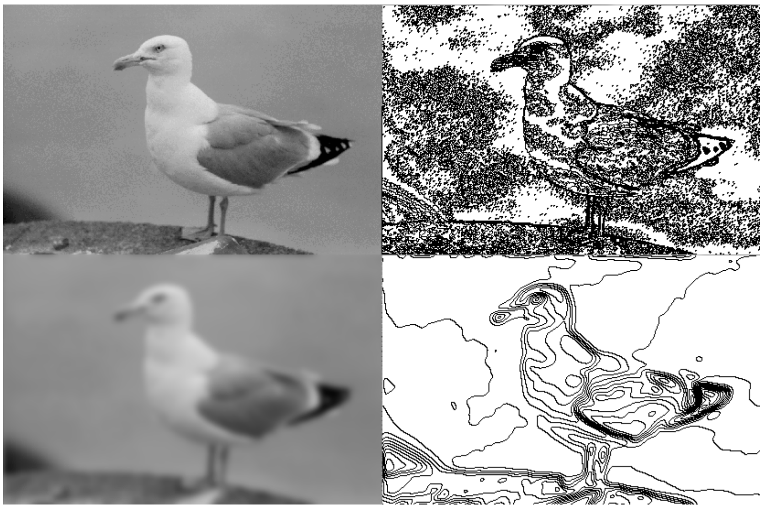

Figure 2.4: Level lines and the heat equation. Top, left to right: original 410×270 grey level image; level lines of original image for levels at multiples of 12. Bottom, left to right: original image smoothed by the heat equation (convolution with the Gaussian). The standard deviation of the Gaussian is 4, which means that its spatial range is comparable to a disk of radius 4. The image gets blurred by the convolution, which averages grey level values and removes all sharp edges. This can be appreciated on the right, where we have displayed all level lines for levels at multiples of 12. Note how some level lines on the boundaries of the image have split into parallel level lines that have drifted away from each other. The image has become smooth, but it is losing its structure.

2.3 Exercises

Exercise 2.5. The aim of this exercise is to prove relation (2.2) and its consequence: A local average is equivalent to one step forward of the heat equation. Theorem 3.2 yields actually a more general statement.

1) Expanding u_0 around the point $\mathbf{x}$ using Taylor's formula, write

$$u_0(\mathbf{x}+\mathbf{y}) = u_0(\mathbf{x}) + Du_0(\mathbf{x})\cdot\mathbf{y} + \frac{1}{2}D^2u_0(\mathbf{x})(\mathbf{y},\mathbf{y}) + o(|\mathbf{y}|^2). \tag{2.18}$$

Expand the various terms using the coordinates (x,y) of $\mathbf{x}$.

2) Apply M_h to both sides of this expansion and deduce relation (2.2).

3) Assume $u_0 \in \mathcal{F}$ and consider the solution $u(t,\mathbf{x})$ of the heat equation (2.3) Then, for fixed $t_0 > 0$ and $\mathbf{x}$, apply M_h to the function $u^{t_0} : \mathbf{x} \to u(t_0,\mathbf{x})$ and write equation (2.2) for u^{t_0}. Using that $u(t,\mathbf{x})$ is a solution of the heat equation and its Taylor expansion between t_0 and t_0+h, deduce that

$$M_h u(t_0,\mathbf{x}) = u(t_0+h^2,\mathbf{x}) + h^2\varepsilon(t_0,\mathbf{x},h). \tag{2.19}$$

■

Exercise 2.6. Consider the sphere $S_N = \{\mathbf{z} \in \mathbb{R}^{N+1},\ ||\mathbf{z}|| = 1\}$. Prove that the mapping $T:\ \mathbb{R}^N \cup \{\infty\} \to S_N$ defined in Proposition 1.1 by

$$T(\mathbf{x}) = \left(\frac{2\mathbf{x}}{1+\mathbf{x}^2}, \frac{\mathbf{x}^2-1}{\mathbf{x}^2+1}\right),\ T(\infty) = (0,1).$$

is a homeomorphism.

Exercise 2.7. Let u_0 be a continuous function defined on $\mathbb{R}^N$ having the property that there exist a constant $C > 0$ and an integer k such that

$$|u_0(\mathbf{x})| \leq C(1 + |\mathbf{x}|^k)$$

for all $\mathbf{x} \in \mathbb{R}^N$. Show that the function u defined by $u(t,\mathbf{x}) = G_t * u_0(\mathbf{x})$ is well defined and C^∞ on $(0,\infty) \times \mathbb{R}^N$ and that it is a classical solution of the heat equation. Hints: Everything follows from the fact that the Gaussian and all of its derivatives decay exponentially at infinity. ■

Exercise 2.8. We want to prove the general principle that any linear, translation invariant and continuous operator T is a convolution, that is $Tu = g*u$ for some kernel g. This is one of the fundamental principles of both mechanics and signal processing, and it has many generalizations that depend on the domain, range, and continuity properties of T. For instance, assume that T is translation invariant (commutes with translations) and is continuous from $L^2(\mathbb{R}^N)$ into $L^\infty(\mathbb{R}^N) \cap C^0(\mathbb{R}^N)$. Show that $Tu = g * u$, where the convolution kernel g is in $L^2(\mathbb{R}^N)$. This is a direct consequence of Riesz theorem, which states that every bounded linear functional on $L^2(\mathbb{R}^N)$ has the form $f \mapsto \int_{\mathbb{R}^N} f(\mathbf{x})g(\mathbf{x})\,\mathrm{d}\mathbf{x}$ for some $g \in L^2(\mathbb{R}^N)$. Show that if $u \geq 0$ ($u(\mathbf{x}) \geq 0$ for all $\mathbf{x}$) implies $Tu \geq 0$, then $g \geq 0$. ■

2.4 Comments and references

The heat equation. One should not conclude from Theorem 2.9 that the solutions of the heat equation are always unique. The assumption in (ii) that the solution was bounded is crucial. In fact, without this assumption, there are solutions u that grow so fast that gu is not in $L^1(\mathbb{R}^N)$ for $g \in \mathcal{S}$ (see, for example, [242, page 217]). The existence and uniqueness proof of Theorem 2.9 is classic and can be found in most textbooks on partial differential equations, such as Evans [73], Taylor [242], or Brezis [33].

Convolution. The heat equation—its solutions and their uniqueness—has been the main topic in this chapter, but to approach this, we have studied several aspects of the convolution, such as the continuity property (2.5). We also noted that the convolution commutes with translation. Conversely, as a general principle, any linear, translation invariant and continuous operator T is a convolution, that is, $Tu = g * u$ for some kernel g. This is a direct consequence of a result discovered independently by F. Riesz and M. Fréchet in 1907 (see [210, page 61] and exercise 2.8). Since we want smoothing to be translation invariant and continuous in some topology, this means that linear smoothing operators—which are called filters in the context of signal and image processing—are described by their convolution kernels. The Gaussian serves as a model for linear filters because it is the only one whose shape is stable under iteration. Other positive filters change their shape when iterated. This fact will be made precise in the next chapter where we show that a large class of iterated linear filters behaves asymptotically as a convolution with the Gaussian.

Smoothing and the Laplacian. One of the first tools proposed in the early days of image processing in the 1960s came, not surprisingly, directly from signal processing. The idea was to restore an image by averaging the gray levels locally (see, for example, [93] and [107]). The observation that the difference between an image and its local average is proportional to the Laplacian of the image has proved to be one of the most fruitful contributions to image processing. As noted in the Introduction, this method for deblurring an image was introduced by Kovasznay and Joseph in 1955 [136], and it was studied and optimized by Gabor in 1965 [87] (information taken from [147]). (See also [114] and [115].) Burt and Adelson based their Laplacian pyramid algorithm on this idea, and this was one of the results that led to multiresolution analysis and wavelets [36].

Chapter 3

Iterated Linear Filters and the Heat Equation

The title of this chapter is self-explanatory. The next section fixes fairly general conditions so that the difference of a smoothed image and the original be proportional to the Laplacian. The second section proves the main result, namely the convergence of iterated linear filters to the heat equation. So the choice of a smoothing convolution kernel is somewhat forced : Iterating the convolution with a smoothing kernel is asymptotically equivalent to the convolution with a Gauss function. This result is known in Probability as the central limit theorem, where it has a quite different interpretation. In image processing, it justifies the prominent role of Gaussian filtering. A last section is devoted to linear directional filters and their associated differential operators.

3.1 Smoothing and the Laplacian

There are minimal requirements on the smoothing kernels g which we state in the next definition.

Definition 3.1. *We say that a real-valued kernel $g \in L^1(\mathbb{R}^N)$ is Laplacian consistent if it satisfies the following moment conditions:*

(i) $\int_{\mathbb{R}^N} g(\mathbf{x})\, \mathrm{d}\mathbf{x} = 1$.

(ii) *For $i = 1, 2, \dots, N$, $\int_{\mathbb{R}^N} x_i g(\mathbf{x})\, \mathrm{d}\mathbf{x} = 0$.*

(iii) *For each pair $i, j = 1, 2, \dots, N$, $i \neq j$, $\int_{\mathbb{R}^N} x_i x_j g(\mathbf{x})\, \mathrm{d}\mathbf{x} = 0$.*

(iv) *For $i = 1, 2, \dots, N$, $\int_{\mathbb{R}^N} x_i^2 g(\mathbf{x})\, \mathrm{d}\mathbf{x} = \sigma$, where $\sigma > 0$.*

(v) $\int_{\mathbb{R}^N} |\mathbf{x}|^3 |g(\mathbf{x})|\, \mathrm{d}\mathbf{x} < +\infty$.

Note that we do not assume that $g \geq 0$; in fact, many important filters used in signal and image processing are not positive. However, condition (i) implies that g is “on average” positive. A discussion of the necessity of the requirements $(i) - (v)$ is performed in Exercise 3.4.

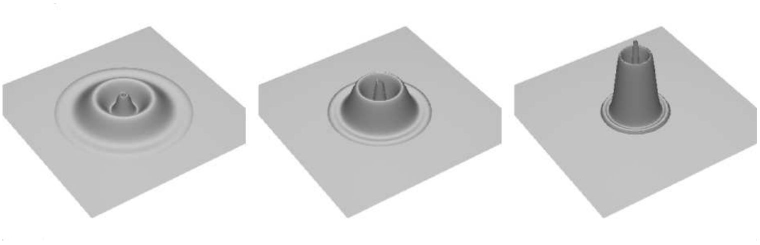

Figure 3.1: The rescalings $g_t(\mathbf{x}) = (1/t^2)g(\mathbf{x}/t)$ of a kernel for t=4, 3, and 2.

We say that a function g is *radial* if $g(\mathbf{x}) = g(|\mathbf{x}|)$, $\mathbf{x} \in \mathbb{R}^N$. This is equivalent to saying that g is invariant under all rotations around the origin in $\mathbb{R}^N$. As pointed out in Exercise 3.3, any radial function $g \in L^1(\mathbb{R}^N)$ can be rescaled to be Laplacian consistent if it decays fast enough at infinity and if $\int_{\mathbb{R}^N} \mathbf{x}_i^2 g(\mathbf{x})\,\mathrm{d}\mathbf{x}$ and $\int_{\mathbb{R}^N} g(\mathbf{x})\,\mathrm{d}\mathbf{x}$ have the same sign.

We consider rescalings of a kernel g defined by

$$g_h(\mathbf{x}) = \frac{1}{h^{N/2}}\, g\Big(\frac{\mathbf{x}}{h^{1/2}}\Big) \tag{3.1}$$

for $h > 0$ (see Figure 3.1). Notice that this rescaling differs slightly from the one used in Section 1.2. We have used the factor $h^{1/2}$ here because it agrees with the factor $t^{1/2}$ in the Gaussian. We denote the convolution of g with itself n times by g^{n*}. The main result of this section concerns the behavior of g_h^{n*} as $n \to +\infty$ and $h \to 0$.

Exercise 3.1. Prove the following two statements:

(i) g_h is Laplacian consistent if and only if g is Laplacian consistent.

(ii) If $g \in L^1(\mathbb{R}^N)$, then $(g_h)^{n*} = (g^{n*})_h$. ■

Theorem 3.2. *If g is Laplacian consistent, then for every $u \in \mathcal{F} \cap C^3(I\!R^N)$,*

$$g_h * u(\mathbf{x}) - u(\mathbf{x}) = h\frac{\sigma}{2}\Delta u(\mathbf{x}) + \varepsilon(h,\mathbf{x}) \tag{3.2}$$

where $|\varepsilon(h,\mathbf{x})| \le Ch^{3/2}$.

Proof. We use condition (i), the definition of g_h, and rescaling $\mathbf{y} = h^{1/2}\mathbf{z}$ inside the integral to see that

$$\begin{aligned} g_h * u(\mathbf{x}) - u(\mathbf{x}) &= \int_{\mathbb{R}^N} \frac{1}{h^{N/2}}\, g\Big(\frac{\mathbf{y}}{h^{1/2}}\Big)\big(u(\mathbf{x}-\mathbf{y}) - u(\mathbf{x})\big)\,\mathrm{d}\mathbf{y} \\ &= \int_{\mathbb{R}^N} g(\mathbf{z})\big(u(\mathbf{x} - h^{1/2}\mathbf{z}) - u(\mathbf{x})\big)\,\mathrm{d}\mathbf{z}. \end{aligned}$$

Using Taylor's formula with the Lagrange remainder (I.1), we have

$$\begin{aligned} u(\mathbf{x} - h^{1/2}\mathbf{z}) - u(\mathbf{x}) \quad &= \quad -h^{1/2}Du(\mathbf{x})\cdot\mathbf{z} + \frac{h}{2}D^2u(\mathbf{x})(\mathbf{z},\mathbf{z}) \\ &\qquad -\frac{1}{6}h^{3/2}D^3u(\mathbf{x} - h^{1/2}\theta\mathbf{z})(\mathbf{z},\mathbf{z},\mathbf{z}), \end{aligned}$$

where $\theta = \theta(\mathbf{x}, \mathbf{z}, h) \in [0,1]$. By condition (ii), $\int_{\mathbb{R}^N} g(\mathbf{z}) Du(\mathbf{x}) \cdot \mathbf{z} \, \mathrm{d}\mathbf{z} = 0$; by conditions (iii) and (iv), $\int_{\mathbb{R}^N} g(\mathbf{z}) D^2 u(\mathbf{x})(\mathbf{z}, \mathbf{z}) \, \mathrm{d}\mathbf{z} = \sigma \Delta u(\mathbf{x})$. Thus,

$$g_h * u(\mathbf{x}) - u(\mathbf{x}) = h\frac{\sigma}{2}\Delta u(\mathbf{x}) - \frac{1}{6}h^{3/2}\int_{\mathbb{R}^N} g(\mathbf{z}) D^3 u(\mathbf{x} - h^{1/2}\theta\mathbf{z})(\mathbf{z}, \mathbf{z}, \mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

We denote the error term by $\varepsilon(h, \mathbf{x})$. Then we have the following estimate:

$$\begin{aligned} |\varepsilon(h, \mathbf{x})| \leq & \frac{1}{6}h^{3/2}\int_{\mathbb{R}^N} |g(\mathbf{z}) D^3 u(\mathbf{x} - h^{1/2}\theta\mathbf{z})(\mathbf{z}, \mathbf{z}, \mathbf{z})| \, \mathrm{d}\mathbf{z} \\ \leq & \frac{1}{6}h^{3/2} N^3 \sup_{\alpha, \mathbf{x}} |\partial^\alpha u(\mathbf{x})| \int_{\mathbb{R}^N} |\mathbf{z}|^3 |g(\mathbf{z})| \, \mathrm{d}\mathbf{z}, \end{aligned}$$

where the supremum is taken over all vectors $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_N)$, $\alpha_j \in \{1, 2, 3\}$, such that $|\alpha| = 3$ and over all $\mathbf{x} \in \mathbb{R}^N$. □

The preceding theorem shows a direct relation between smoothing with a Laplacian-consistent kernel and the heat equation. It also shows why we require σ to be positive: If it is not positive, the kernel is associated with the inverse heat equation (see Exercise 3.4.)

3.2 The convergence theorem

The result of the next theorem is illustrated in Figure 3.2.

Theorem 3.3. *Let g be a nonnegative Laplacian-consistent kernel with $\sigma = 2$ and define g_h by* (3.1). *Write $T_h u_0 = g_h * u_0$ for $u_0 \in \mathcal{F}$, and let $u(t, \cdot) = G_t * u_0$ be the solution of the heat equation* (2.15). *Then, for each $t > 0$,*

$$(T_h^n u_0)(\mathbf{x}) \to u(t, \mathbf{x}) \text{ uniformly in } \mathbf{x} \text{ as } n \to +\infty \text{ and } nh \to t. \tag{3.3}$$

Proof. Let us start with some preliminaries. We have $(g_h * u_0)(\infty) = u_0(\infty)$ and therefore $T_h^n u_0(\infty) = u_0(\infty)$. The norm in $\mathcal{F}$ is $||u||_{\mathcal{F}} = \sup_{\mathbf{x} \in S_N} |u(\mathbf{x})| = \sup_{\mathbf{x} \in \mathbb{R}^N} |u(\mathbf{x})|$. The first order of business is to say precisely what is meant by the asymptotic limit (3.3): Given $t > 0$ and given $\varepsilon > 0$, there exists an $n_0 = n_0(t, \varepsilon)$ and a $\delta = \delta(t, \varepsilon)$ such that $\|T_h^n u_0 - u(t, \cdot)\|_{\mathcal{F}} \leq \varepsilon$ if $n > n_0$ and $|nh - t| \leq \delta$. This is what we must prove. We will first prove the result when $h = t/n$. We will then show that the result is true when h is suitably close to t/n.

We begin with comments about the notation. By Exercise 3.1, $(T_h)^n = (T^n)_h$, so there is no ambiguity in writing T_h^n. We will be applying T_h^n to the solution u of the heat equation, which is C^∞ on $(0, +\infty) \times \mathbb{R}^N$. In this situation, t is considered to be a parameter, and we write $T_h^n u(t, \mathbf{x})$ as shorthand for $T_h^n u(t, \cdot)(\mathbf{x})$. Throughout the proof, we will be dealing with error terms that we write as $O(h^r)$. These terms invariably depend on h, t, and $\mathbf{x}$. However, in all cases, given a closed interval $[t_1, t_2] \subset (0, +\infty)$, there will be a constant C such that $|O(h^r)| \leq Ch^r$ uniformly for $t \in [t_1, t_2]$ and $\mathbf{x} \in \mathbb{R}^N$. Finally, keep in mind that all functions of $\mathbf{x}$ tend to $u_0(\infty)$ as $\mathbf{x} \to \infty$.

We wish to fix an interval $[t_1, t_2]$, but since this depends on the point t in (3.3) and on ε, we must first choose these numbers. Thus, choose $\tau > 0$ and keep it fixed. This will be the "t" in (3.3). Next, choose $\varepsilon > 0$. Here are the conditions we wish t_1 and t_2 to satisfy:

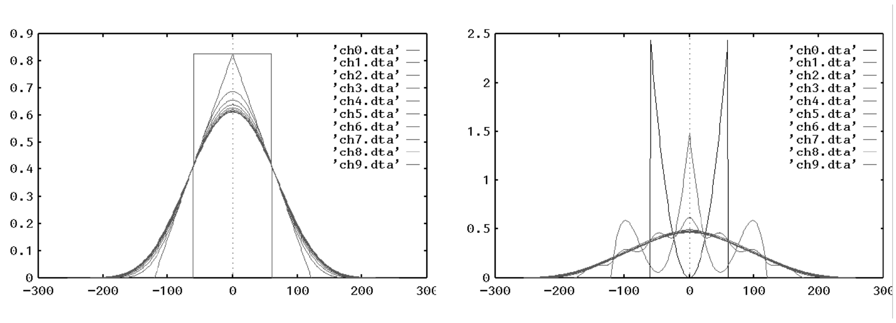

Figure 3.2: Iterated linear smoothing converges to the heat equation. In this experiment with one-dimensional functions, it can be appreciated how fast an iterated convolution of a positive kernel converges to a Gaussian. On the left are displayed nine iterations of the convolution of the characteristic function of an interval with itself, with appropriate rescalings. On the right, the same experiment is repeated with a much more irregular kernel. The convergence is almost as fast as the first case.

(1) t_1 is small enough so $\|u(t_1,\cdot) - u_0\|_{\mathcal{F}} < \varepsilon$. (This is possible by Theorem 2.9.)

(2) t_1 is small enough so $\|u(t_1+\tau,\cdot) - u(\tau,\cdot)\|_{\mathcal{F}} < \varepsilon$. (Again, by Theorem 2.9.)

(3) t_2 is large enough so $t_1 + \tau < t_2$.

There is no problem meeting these conditions, so we fix the interval $[t_1, t_2] \subset (0, +\infty)$.

Step 1, main argument : proof that

$$\lim_{\substack{n\to+\infty\\ nh=\tau}} T_h^n u(t_1, \mathbf{x}) = u(t_1+\tau, \mathbf{x}), \tag{3.4}$$

where the convergence is uniform for $\mathbf{x} \in \mathbb{R}^N$.
We can use Theorem 3.2 to write

$$T_h u(t,\mathbf{x}) - u(t,\mathbf{x}) = h\Delta u(t,\mathbf{x}) + O(h^{3/2}), \tag{3.5}$$

where $t \in [t_1, t_2]$. That the error function is bounded uniformly by $Ch^{3/2}$ on $[t_1, t_2] \times \mathbb{R}^N$ follows from the fact that $\sup_{\alpha,t,\mathbf{x}} |\partial^\alpha u(t,\mathbf{x})|$ is finite for $(t,\mathbf{x}) \in [t_1, t_2] \times \mathbb{R}^N$ (see the proof of Theorem 3.2). Since u is a solution of the heat equation, we also have by Taylor-Lagrange formula (I.1),

$$u(t+h,\mathbf{x}) - u(t,\mathbf{x}) = h\frac{\partial u}{\partial t}(t,\mathbf{x}) + \frac{h^2}{2}\frac{\partial^2}{\partial t^2}u(t+\theta h,\mathbf{x}) = h\Delta u(t,\mathbf{x}) + O(h^2). \tag{3.6}$$

This time the error term is bounded uniformly by Ch^2 on $[t_1, t_2] \times \mathbb{R}^N$ because $\frac{\partial^2 u}{\partial t^2}(t,\mathbf{x})$ is bounded on $[t_1, t_1] \times \mathbb{R}^N$. By subtracting (3.6) from (3.5) we see that

$$T_h u(t,\mathbf{x}) = u(t+h,\mathbf{x}) + O(h^{3/2}). \tag{3.7}$$

This shows that applying T_h to a solution of the heat equation at time t advances the solution to time $t+h$, plus an error term.

So far we have not used the assumption that g is nonnegative. Thus, (3.7) is true for any Laplacian-consistent kernel g with $\sigma = 2$. However, we now wish to apply the linear operator T_h to both sides of equation (3.7), and in doing so we do not want the error term to increase. Since $g \geq 0$, this is not a problem:

$$|T_h O(h^{3/2})| \leq \int_{\mathbb{R}^N} |O(h^{3/2})| g_h(\mathbf{x}-\mathbf{y})\,\mathrm{d}\mathbf{y} \leq \int_{\mathbb{R}^N} Ch^{3/2} g_h(\mathbf{x}-\mathbf{y})\,\mathrm{d}\mathbf{y} = Ch^{3/2}.$$

With this in hand, we can apply T_h to both sides of (3.7) and obtain

$$T_h^2 u(t,\mathbf{x}) = T_h u(t+h,\mathbf{x}) + O(h^{3/2}). \tag{3.8}$$

If we write equation (3.7) with $t+h$ in place of t and substitute the expression for $T_h u(t+h,\mathbf{x})$ in equation (3.8), we have

$$T_h^2 u(t,\mathbf{x}) = u(t+2h,\mathbf{x}) + 2O(h^{3/2}). \tag{3.9}$$

We can iterate this process and get

$$T_h^n u(t,\mathbf{x}) = u(t+nh,\mathbf{x}) + nO(h^{3/2}) \tag{3.10}$$

with the same constant C in the estimate $|O(h^{3/2})| \leq Ch^{3/2}$ as long as $t+nh \in [t_1,t_2]$. To ensure that this happens, we take $t = t_1$ and $h = \tau/n$. Then

$$T_h^n u(t_1,\mathbf{x}) = u(t_1+\tau,\mathbf{x}) + O\Big(\Big(\frac{\tau}{n}\Big)^{1/2}\Big) \tag{3.11}$$

and we obtain (3.4). If we could take $t_1 = 0$, this would end the proof. This is not possible because all of the O terms were based on a fixed interval $[t_1,t_2]$. However, we have taken t_1 small enough to finish the proof .

Step 2 : getting rid of t_1.
Since $\int_{\mathbb{R}^N} g(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1$, $\|g_h\|_{L^1(\mathbb{R}^N)} = 1$, and thus

$$\|g_h^{n*} * v\|_{\mathcal{F}} \leq \|v\|_{\mathcal{F}}.$$

If we take $v = u(t_1,\cdot) - u_0$, then this inequality and condition (1) imply that

$$\|T_h^n u(t_1,\cdot) - T_h^n u_0\|_{\mathcal{F}} < \varepsilon. \tag{3.12}$$

Relations (3.12) and (3.11) imply that

$$\|T_h^n u_0 - u(t_1+\tau,\cdot)\|_{\mathcal{F}} < 2\varepsilon. \tag{3.13}$$

This inequality and condition (2) show that

$$\|T_h^n u_0 - u(\tau,\cdot)\|_{\mathcal{F}} < 3\varepsilon \tag{3.14}$$

for $n > n_0$ and $h = \tau/n$. This proves the theorem in the case $h = \tau/n$.

Conclusion. It is a simple matter to obtain the more general result. Again, by Theorem 1.10, there is a $\delta = \delta(\tau,\varepsilon)$ such that $|nh-\tau| < \delta$ implies that $\|u(nh,\cdot) - u(\tau,\cdot)\|_{\mathcal{F}} < \varepsilon$ and that $nh \in [t_1,t_2]$ (by condition (3)). Combining this with (3.14) shows that

$$\|T_h^n u_0 - u(nh,\cdot)\|_{\mathcal{F}} < 4\varepsilon$$

if $n > n_0$ and $|nh-\tau| < \delta$, and this completes the proof. □

3.3 Directional averages and directional heat equations

In this section, we list easy extensions of Theorem 3.2. They analyze local averaging processes which take averages at each point in a singular neighborhood made of a segment. In that way, we will make appear several nonlinear generalizations of the Laplacian which will accompany us throughout the book. Consider a C^2 function from $\mathbb{R}^N$ into $\mathbb{R}$ and a vector $\mathbf{z} \in \mathbb{R}^N$ with $|\mathbf{z}| = 1$. We wish to compute the mean value of u along a segment of the line through $\mathbf{x}$ parallel to the vector $\mathbf{z}$. To do this, we define the operator $T_h^{\mathbf{z}}$, $h \in [-1, 1]$, by

$$T_h^{\mathbf{z}} u(\mathbf{x}) = \frac{1}{2h} \int_{-h}^{h} u(\mathbf{x} + s\mathbf{z})\, \mathrm{d}s.$$

This operator is the directional counterpart of the isotropic operator M_h defined by equation (1.1).

Proposition 3.4.

$$T_h^{\mathbf{z}} u(\mathbf{x}) = u(\mathbf{x}) + \frac{h^2}{6} D^2 u(\mathbf{x})(\mathbf{z}, \mathbf{z}) + o(h^2).$$

Proposition 3.4 is deduced from Theorem 3.2, and it suggests that iterations of the operator $T_h^{\mathbf{z}}$ are associated with the directional heat equation

$$\frac{\partial u}{\partial t}(t, \mathbf{x}) = \frac{1}{6} D^2 u(t, \mathbf{x})(\mathbf{z}, \mathbf{z}) \tag{3.15}$$

in the same way that the iterations of the operator T_h in Theorem 3.3 are associated with the ordinary heat equation. If $\mathbf{z}$ is fixed, then the operator $T_h^{\mathbf{z}}$ and equation (3.15) act on u along each line in $\mathbb{R}^N$ parallel to $\mathbf{z}$ separately; there is no “cross talk” between lines. Exercise 3.5 formalizes and clarifies these comments when $\mathbf{z}$ is fixed. However, Proposition 3.4 remains true when $\mathbf{z}$ is a function of $\mathbf{x}$. This means that we are able to approximate the directional second derivative by taking directional averages where $\mathbf{z}$ varies from point to point. The main choices considered in the book are $\mathbf{z} = Du/|Du|$ and $\mathbf{z} = Du^{\perp}/|Du|$, where $Du = (u_x, u_y)$ and $Du^{\perp} = (-u_y, u_x)$. Then by Proposition 3.4 we have the following limiting relations:

- Average in the direction of the gradient. By choosing $z = Du/|Du|$,

  $$\frac{1}{|Du|^2} D^2 u(Du, Du) = 6 \lim_{h \to 0} \frac{T_h^{Du/|Du|} u - u}{h^2}.$$

  We will interpret this differential operator as Haralick’s edge detector in section 6.1.

- Average in the direction orthogonal to the gradient. By choosing $z = Du/|Du^{\perp}|$,

  $$\frac{1}{|Du|^2} D^2 u(Du^{\perp}, Du^{\perp}) = 6 \lim_{h \to 0} \frac{T_h^{Du^{\perp}/|Du|} u - u}{h^2}.$$

  This differential operator appears as the second term of the curvature equation. (See Chapter 18.)

Although we have not written them as such, the limits are pointwise in both cases.

3.4 Exercises

Exercise 3.2. We will denote the characteristic function of a set $A \subset \mathbb{R}^N$ by $\mathbf{1}_A$. Thus, $\mathbf{1}_A(\mathbf{x}) = 1$ if $\mathbf{x} \in A$ and $\mathbf{1}_A(\mathbf{x}) = 0$ otherwise. Consider the kernel $g = (1/\pi)\mathbf{1}_{D(0,1)}$, where $D(0,1)$ is the disk of radius one centered at zero. In this case, g is a radial function and it is clearly Laplacian consistent. For $N = 2$, let $A = [-1/2, 1/2] \times [-1/2, 1/2]$. Then $g = \mathbf{1}_A$ is not radial. Show that it is, however, Laplacian consistent. If we take $B = [-1,1] \times [-1/2, 1/2]$, then $g = (1/2)\mathbf{1}_B$ is no longer Laplacian consistent because it does not satisfy condition (iv). Show that this kernel does, however, satisfy a relation similar to (3.2). ■

Exercise 3.3. The aim of the exercise is to prove roughly that radial functions with fast decay are Laplacian consistent. Assume $g \in L^1(\mathbb{R}^N)$ is radial with finite first second moments, $\int_{\mathbb{R}^N} |\mathbf{x}|^k |g(\mathbf{x})| \, \mathrm{d}\mathbf{x} < +\infty$, $k = 0, 1, 2, 3$ and such that $\int_{\mathbb{R}^N} \mathbf{x}_i^2 g(\mathbf{x}) \, \mathrm{d}\mathbf{x} > 0$. Show that g satisfies conditions (ii) and (iii) of Definition 3.1 and that, for suitably chosen $a, b \in \mathbb{R}$, the rescaled function $\mathbf{x} \mapsto ag(\mathbf{x}/b)$ satisfies conditions (i) and (iv), where σ can be taken to be an arbitrary positive number. ■

Exercise 3.4. The aim of the exercise is to illustrate by simple examples what happens to the iterated filter g^{n*}, $n \in \mathbb{N}$ when g does not satisfy some of the requirements of the Laplacian consistency (Definition 3.1). We recall the notation (3.1), $g_h(\mathbf{x}) = \frac{1}{h^{N/2}}\, g\Big(\frac{\mathbf{x}}{h^{1/2}}\Big)$.

1) Take on $\mathbb{R}$, $g(x) = 1$ on $[-1,1]$, $g(x) = 0$ otherwise. Which one of the assumptions $(i) - (v)$ is not satisfied in Definition 3.1 ? Compute $g^{n*}_{\frac{1}{n}} * u$, where $u = 1$ on $\mathbb{R}$. Conclude : the iterated filter blows up.

2) Take on $\mathbb{R}$, $g(x) = 1$ on $[0,1]$, $g(x) = 0$ otherwise. Which one of the assumptions $(i) - (v)$ is not satisfied in Definition 3.1 Compute $g^{n*} * u$, where $u(x) = x$ on $\mathbb{R}$. Conclude : the iterated filter "drifts".

3) Assume that the assumptions $(i) - (v)$ hold, except (iii). By a simple adaptation of its proof, draw a more general form of Theorem 3.2.

4) Perform the same analysis as in 3) when all assumptions hold but (iv).

5) Take the case of dimension $N = 1$ and assume that (i) hold but (ii) does not hold. Set $g_h(\mathbf{x}) = \frac{1}{h}\, g\Big(\frac{\mathbf{x}}{h}\Big)$ and give a version of Theorem 3.2 in that case (make an order 1 Taylor expansion of u). ■

Exercise 3.5. Let $\mathbf{z}$ be a fixed vector in $\mathbb{R}^N$ with $|\mathbf{z}| = 1$ and let u_0 be in $\mathcal{F}$. Define a one-dimensional kernel g by $g(s) = \frac{1}{2}\mathbf{1}_{[-1,1]}(s)$.

(i) Show that g is Laplacian consistent. Compute the variance σ of g.

(ii) Show that

$$u(t, \mathbf{x}) = \int_{\mathbb{R}} u_0(\mathbf{x} + s\mathbf{z}) G_t(s) \, \mathrm{d}s$$

is a solution of the directional heat equation

$$\frac{\partial u}{\partial t}(t, \mathbf{x}) = D^2 u(t, \mathbf{x})(\mathbf{z}, \mathbf{z}), \quad u(0, \mathbf{x}) = u_0(\mathbf{x}). \tag{3.16}$$

Give an example to show that $u(t, \cdot)$ is not necessarily C^2. This being the case, how does one interpret the right-hand side of (3.16)?

(iii) Let $g_h(s) = (6h)^{-1/2} g(s/(6h)^{-1/2})$ and $T_h u(\mathbf{x}) = \int_{\mathbb{R}} u(\mathbf{x} + s\mathbf{z}) g_h(s) \, \mathrm{d}s$. By applying Theorem 3.3 for $N = 1$, show that, for each $t > 0$,

$$T_h^n u_0 \to u(t, \cdot) \text{ in } \mathcal{F} \text{ as } n \to +\infty \text{ and } nh \to t. \ ■ \tag{3.17}$$

Exercise 3.6. The Weickert equation can be viewed as a variant of the curvature equation [256]. It uses a nonlocal estimate of the direction orthogonal to the gradient for the diffusion direction. This direction is computed as the direction v of the eigenvector corresponding to the smallest eigenvalue of $k*(Du\otimes Du)$, where $(\boldsymbol{y}\otimes\boldsymbol{y})(\boldsymbol{x}) = (\boldsymbol{x}\cdot\boldsymbol{y})\boldsymbol{y}$ and k is a smooth kernel, typically a gaussian. Prove that if the convolution kernel is removed, then this eigenvector is simply $Du^{\perp}$. So the equation writes

$$\frac{\partial u}{\partial t} = u_{\eta\eta}, \tag{3.18}$$

where η denotes the coordinate in the direction v. ■

Exercise 3.7. Suppose that $u \in C^2(\mathbb{R})$. Assuming that $u'(x) \neq 0$, show that

$$u''(x) = \lim_{h\to 0}\frac{1}{h^2}\Big(\max_{s\in[-h,h]} u(x+s) + \min_{s\in[-h,h]} u(x+s) - 2u(x)\Big). \tag{3.19}$$

What is the value of the right-hand side of (3.19) if $u'(x) = 0$?

Now consider $u \in C^2(\mathbb{R}^2)$. We wish to establish an algorithm similar to (3.19) to compute the second derivative of u in the direction of the gradient $Du = (u_x, u_y)$. For this to make sense, we must assume that $Du(\mathbf{x}) \neq 0$. With these assumptions, we know from (3.19) that

$$u_{\xi\xi}(\mathbf{x}) = \frac{\partial^2 v}{\partial \xi^2}(\mathbf{x},0) = \lim_{h\to 0}\frac{1}{h^2}\Big(\max_{s\in[-h,h]} u(\mathbf{x}+s\mathbf{z}) + \min_{s\in[-h,h]} u(\mathbf{x}+s\mathbf{z}) - 2u(\mathbf{x})\Big), \tag{3.20}$$

where $v(\mathbf{x},\xi) = u(\mathbf{x}+\xi\mathbf{z})$ and $\mathbf{z} = Du/|Du|$. The second part of the exercise is to prove that, in fact,

$$u_{\xi\xi}(\mathbf{x}) = \lim_{h\to 0}\frac{1}{h^2}\Big(\max_{\mathbf{y}\in D(0,h)} u(\mathbf{x}+\mathbf{y}) + \min_{\mathbf{y}\in D(0,h)} u(\mathbf{x}+\mathbf{y}) - 2u(\mathbf{x})\Big), \tag{3.21}$$

where $D(0,h)$ is the disk of radius h centered at the origin. Intuitively, (3.21) follows from (3.20) because the gradient indicates the direction of maximal change in $u(\mathbf{x})$, so in the limit as $h \to 0$, taking max and min in the direction of the gradient is equivalent to taking max and min in the disk. The point of the exercise is to formalize this. ■

3.5 Comments and references

Asymptotics. Our proof that iterated and rescaled convolutions of a Laplacian-consistent kernel tend asymptotically to the Gaussian is a version of the De Moivre–Laplace formula, or the central limit theorem, adapted to image processing [32]. This result is particularly relevant to image analysis, since it implies that iterated linear smoothing leads inevitably to convolution with the Gaussian, or equivalently, to the application of the heat equation. We do not wish to imply, however, that the Gaussian is the only important kernel for image processing. The Gaussian plays a significant role in our form of image analysis, but there are other kernels that, because of their spectral and algebraic properties, have equally important roles in other aspects of signal and image processing. This is particularly true for wavelet theory which combines recursive filtering and sub-sampling.

Directional diffusion. Directional diffusion has a long history that began when Hubel and Wiesel showed the existence of direction-sensitive cells in the visual areas of the neocortex [112]. There has been an explosion of publication on directional linear filters, beginning, for example, with influential papers such as that by Daugman [65]. We note again that Gabor's contribution to directional filtering is described in [147].

Chapter 4

From continuous to digital images, and back

Consider a continuous and bounded image $u(\mathbf{x})$ defined for every $\mathbf{x} = (x, y) \in \mathbb{R}^2$. All continuous *image* operators including the sampling will be written in capital letters A, B and their composition as a mere juxtaposition AB. For any affine map A of the plane consider the affine transform of a continuous image u defined by $Au(\mathbf{x}) =: u(A\mathbf{x})$. For instance $H_\lambda u(\mathbf{x}) =: u(\lambda\mathbf{x})$ denotes an expansion of u by a factor λ^{-1}. In the same way if R is a rotation, $Ru =: u \circ R$ is the image rotation by R^{-1}.

Sampling and interpolation

Digital images, only defined for $(n_1, n_2) \in \mathbb{Z}^2$, will be denoted by $\mathbf{u}(n_1, n_2)$. The δ-sampled image $\mathbf{u} = \mathbf{S}_\delta u$ is defined on $\mathbb{Z}^2$ by

$$\mathbf{u}(n_1, n_2) = (\mathbf{S}_\delta)u(n_1, n_2) =: u(n_1\delta, n_2\delta); \tag{4.1}$$

Conversely, the Shannon interpolate of a digital image $\mathbf{u}$ is defined as follows [91]. Let $\mathbf{u}$ be a digital image, defined on $\mathbb{Z}^2$ and such that $\sum_{n\in\mathbb{Z}^2} |\mathbf{u}(n)|^2 < \infty$ and $\sum_{n\in\mathbb{Z}^2} |\mathbf{u}(n)| < \infty$. (Of course, these conditions are automatically satisfied if the digital has a finite number of non-zero samples, which is the case here.) We call Shannon interpolate $I\mathbf{u}$ of $\mathbf{u}$ the only $L^2(\mathbb{R}^2)$ function u having $\mathbf{u}$ as samples and with spectrum support contained in $(-\pi, \pi)^2$. $u = I\mathbf{u}$ is defined by the Shannon-Whittaker formula

$$I\mathbf{u}(x, y) =: \sum_{(n_1,n_2)\in\mathbb{Z}^2} \mathbf{u}(n_1, n_2)\mathrm{sinc}(x - n_1)\mathrm{sinc}(y - n_2),$$

where sinc $x =: \frac{\sin \pi x}{\pi x}$. The Shannon interpolation has the fundamental property $\mathbf{S}_1 I\mathbf{u} = \mathbf{u}$. Conversely, if u is L^2 continuous image, band-limited in $(-\pi, \pi)^2$, then

$$I\mathbf{S}_1 u = u. \tag{4.2}$$

In that case we simply say that u is *band-limited.* We shall also say that a digital image $\mathbf{u} = \mathbf{S}_1 u$ is *well-sampled* if it was obtained from a band-limited image u.

4.0.1 The practical Shannon interpolation: zero-padding

Of course, the Shannon interpolate is unpractical in that it assumes the knowledge of infinitely many samples. In practice image samples and image interpolation will be performed on rectangle. For a sake of simplicity we describe here what happens on a square. Let $a > 0$ and consider a function u from $[0,a]^2$ to $I\!R$ such that $u(x+a, y+a) = u(x,y)$. Fix an integer N, and consider the N^2 samples of u, $\mathbf{u}_{k,l} = (\mathbf{S}u)\left(\frac{ka}{N}, \frac{la}{N}\right)$ on $[0,a]^2$.

Definition 4.1. *The discrete Fourier transform (DFT) of the N^2 samples $u = (u_{k,l})_{k,l=0,1,\dots,N-1}$ is the double sequence of discrete Fourier coefficients for $m,n \in \{-\frac{N}{2},\dots,\frac{N}{2}-1\}$ defined by*

$$DFT(u)_{m,n} = \tilde{u}_{m,n} = \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} u_{k,l}\omega_N^{-mk}\omega_N^{-nl}, \tag{4.3}$$

where $\omega_N =: e^{\frac{2i\pi}{N}}$ is the first N-root of 1.

Proposition 4.2. *Consider the trigonometric polynomial*

$$I\mathbf{u}(x,y) = \sum_{m,n=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_{m,n}\exp(\frac{2i\pi mx}{a})\exp(\frac{2i\pi ny}{a}). \tag{4.4}$$

Then its coefficients $\tilde{u}_{m,n}$ are the only complex numbers such that for every $k,l \in \{0,\dots,N-1\}$, $I\mathbf{u}\left(\frac{ka}{N},\frac{la}{N}\right) = \mathbf{u}\left(\frac{ka}{N},\frac{la}{N}\right)$. In consequence, the discrete inverse transform of the DFT $\mathbf{u} \to \tilde{u}$ is nothing but the calculation of the value of the polynomial at the samples $(\frac{ka}{N},\frac{la}{N})$, $0 \le k,l \le N-1$. In other terms, setting $\tilde{u} = DFT(\mathbf{u})$, the inverse transform DFT^{-1} is given by

$$\mathbf{u}(k,l) = \sum_{-\frac{N}{2}}^{\frac{N}{2}-1}\sum_{-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{\mathbf{u}}_{m,n}\omega_N^{km+ln},$$

for every $k,l = 0,1,\dots N-1$.

Exercise 4.1. Recall that $\omega_N = \exp\left(\frac{2i\pi}{N}\right)$, N-th root of 1. Show that $\sum_{k=0}^{N-1}\omega_N^k = 0$, and that $\sum_{k=0}^{N-1}\omega_N^{kl} = 0$ for $l \neq 0$ modulo N and finally that for every k_0, $\sum_{k=k_0}^{k_0+N-1}\omega_N^{kl} = 0$ for all $l \neq 0$ modulo N. Using these relation show the above proposition, namely that $DFT(DFT^{-1}) = Id$. ■

In conclusion, the interpolation and sampling operators we shall consider both in theory and practice are the usual sampling $\mathbf{S}$, implicitly restricted to a square. The inverse interpolation operator I is defined by (4.4), and Proposition 4.2 tells us that $\mathbf{S}I\mathbf{u} = \mathbf{u}$. The next statement gives the converse statement.

Proposition 4.3. *If $u(x,y)$ is a a-periodic band-limited function, then it is a trigonometric polynomial. If its highest degree is $\frac{N}{2}-1$, with N even, then its coefficients $\tilde{u}_{m,n}$ are obtained by DFT from the samples $\mathbf{u}(k,l) = u(\frac{ka}{N},\frac{la}{N})$. In consequence for such functions we have $I\mathbf{S}u = u$.*

Exercise 4.2. Give a detailed proof of Proposition 4.3. It is a direct consequence of Proposition 4.2. ■

In the rest of this chapter and of the book, we shall always take the functional setting of Proposition 4.3.

Zoom in by zero-padding

Let $(u_{k,l})$ be a digital image and define its zoomed version $(v_{i,j})_{i,j=0,\ ...,\ 2N-1}$ as the inverse discrete Fourier transform of $\tilde{v}_{i,j}$ defined for $i,j=-N,\ ...,\ N-1$ by

$$\tilde{v}_{m,n}=\tilde{u}_{m,n} \text{ if } -\frac{N}{2}\leq m,n\leq\frac{N}{2}-1,\ \ \tilde{v}_{m,n}=0 \text{ otherwise.} \qquad (4.5)$$

Proposition 4.4. *The image v whose Discrete Fourier Transform is given by (4.5) satisfies $v_{2k,2l}=u_{k,l}$, for $k,l=0,...,N-1$.*

Proof. Here is the proof in dimension 1:

$$v_{2k}=\sum_{-N}^{N-1}\tilde{v}_n\omega_{2N}^{2nk}=\sum_{-\frac{N}{2}}^{\frac{N}{2}-1}\tilde{u}_n\omega_N^{nk}=u_k.$$

Indeed, $\omega_{2N}^{2nk}=\omega_N^{nk}$. □

Exercise 4.3. Prove Proposition (4.4) in two dimensions. ■

4.1 The Gaussian semigroup

For a sake of simple notation, G_σ denotes the convolution operator on $\mathbb{R}^2$ with the gauss kernel $G_\sigma(x_1,x_2)=\frac{1}{2\pi(\sigma)^2}e^{-\frac{x_1^2+x_2^2}{2(\sigma)^2}}$, namely $G_\sigma\mathbf{u}(x,y)=:(G_\sigma*\mathbf{u})(x,y)$. Notice that the paramaterization of the gaussian is *not* the same as the parameterization used for the heat equation. To make a difference in notation, we use Greek letters for the new parameter. It is easily checked that G_σ satisfies the semigroup property

$$G_\sigma G_\beta=G_{\sqrt{\sigma^2+\beta^2}}. \qquad (4.6)$$

Exercise 4.4. Prove (9.1). ■

The proof of the next formula is a mere change of variables in the integral defining the convolution.

$$G_\sigma H_\gamma u=H_\gamma G_{\sigma\gamma}u. \qquad (4.7)$$

Exercise 4.5. Prove (9.2). ■

Discrete Gaussians

Many algorithms in computer vision and image processing assume that all blurs can be assumed gaussian. Thus, it will be crucial to prove that gaussian blur gives *in practice* well-sampled images. Thus, in all that follows, we are dealing with initial digital images obtained by sampling a continuous image with gaussian blur, $\mathbf{u}=\mathbf{S}_1G_\mathbf{c}u_0$;

Another question we need to deal with is how to perform a gaussian convolution on a discrete (digital) image. This is valid if and only if a discrete convolution can give an account of the underlying continuous one.

Definition 4.5. *The* **discrete gaussian convolution** *applied to a digital image* $\mathbf{u}$ *is defined as a digital operator by*

$$\mathbf{G}_\delta \mathbf{u} =: \mathbf{S}_1 G_\delta I \mathbf{u}. \tag{4.8}$$

Proposition 4.6. *This definition maintains the gaussian semi-group property,*

$$\mathbf{G}_\delta \mathbf{G}_\beta = \mathbf{G}_{\sqrt{\delta^2+\beta^2}}. \tag{4.9}$$

Proof Indeed, using twice (4.8) and once (9.1) and (4.2),

$$\mathbf{G}_\delta \mathbf{G}_\beta \mathbf{u} = \mathbf{S}_1 G_\delta I \mathbf{S}_1 G_\beta I u = \mathbf{S}_1 G_\delta G_\beta I \mathbf{u} = \mathbf{S}_1 G_{\sqrt{\delta^2+\beta^2}} I \mathbf{u} = \mathbf{G}_{\sqrt{\delta^2+\beta^2}} \mathbf{u}.$$

□

The SIFT method that we will study in detail uses repeatedly the semi-group formula and a 2-sub-sampling of images with a gaussian blur larger than 1.6. These SIFT sampling manoeuvres are valid if and only if the *empirical proposition* below is true.

Proposition 4.7. *For every σ larger than 0.8 and every continuous and bounded image u_0, the gaussian blurred image $G_\sigma u_0$ is well sampled, namely $I\mathbf{S}_1 G_\sigma u_0 = G_\sigma u_0$.*

This proposition is not a mathematical statement, but it will be checked experimentally in the next section, where we shall see that a 0.6 blur is enough to ensure good sampling in practice.

4.2 The right gaussian blur for well-sampling

Images need to be blurred before they are sampled. In principle gaussian blur cannot lead to a good sampling because it is not *stricto sensu* band limited. Therefore the Shannon-Whittaker formula does not apply. However, in practice it does. The aim in this section is to define a procedure that checks that a gaussian blur works and to fix the minimal variance of the blur ensuring well-sampling (up to a minor mean square and visual error).

One must distinguish two types of blur: The *absolute* blur with standard deviation $\mathbf{c}_a$ is the one that must be applied to an ideal infinite resolution (blur free) image to create an approximately band-limited image before 1-sampling. The *relative* blur $\sigma = \mathbf{c}_r(t)$ is the one that must be applied to a well-sampled image before a sub-sampling by a t factor. In the case of gaussian blur, because of the semi-group formula (9.1), the relation between the absolute and relative blur is

$$t^2\mathbf{c}_a^2 = \mathbf{c}_r^2(t) + \mathbf{c}_a^2,$$

which yields

$$\mathbf{c}_r(t) = \mathbf{c}_a\sqrt{t^2-1}. \tag{4.10}$$

In consequence, if $t \gg 1$, then $\mathbf{c}_r(t) \approx \mathbf{c}_a t$.

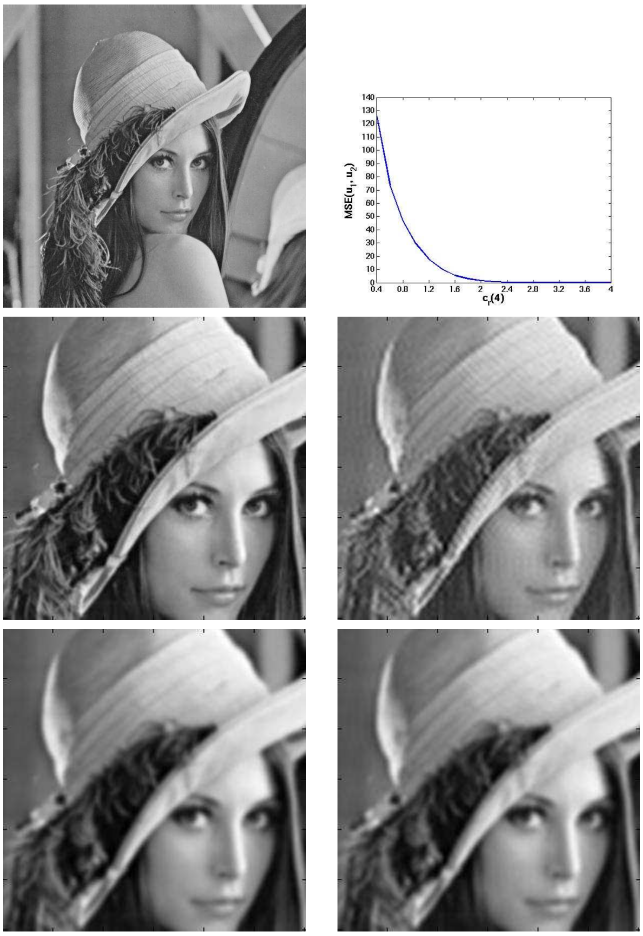

Figure 4.1: Top left: $\mathbf{u}$. Top right: MSE$(\mathbf{u}_1, \mathbf{u}_2)$ vs $\mathbf{c}_r(4)$. Middle (from left to right): $\mathbf{u}_1$ and $\mathbf{u}_2$ with $\mathbf{c}_r(4) = 1.2$. MSE$(\mathbf{u}_1, \mathbf{u}_2)$=17.5. Bottom (from left to right): $\mathbf{u}_1$ and $\mathbf{u}_2$ with $\mathbf{c}_r(4) = 2.4$. MSE$(\mathbf{u}_1, \mathbf{u}_2)$=0.33. Digital images are always displayed by coloring each square pixel with its central sample value.

Two experiments have been designed to calculate the anti-aliasing absolute gaussian blur $\mathbf{c}_a$ ensuring that an image is approximately well-sampled. The first experiment compares for several values of $\mathbf{c}_r(t)$ the digital images

$$\mathbf{u}_1 =: \mathbf{G}_{\mathbf{c}_r(t)} u = \mathbf{S}_1 G_{\mathbf{c}_r(t)} I\mathbf{u} \quad \text{and}$$

$$\mathbf{u}_2 =: (\mathbf{S}_{1/t} I)\mathbf{S}_t \mathbf{G}_{\mathbf{c}_r(t)} I\mathbf{u} = (\mathbf{S}_{1/t} I)\mathbf{S}_t G_{\mathbf{c}_r(t)} I\mathbf{u},$$

where $\mathbf{u}$ is an initial digital image that is (intuitively) well-sampled, $\mathbf{S}_t$ is a t sub-sampling operator, $\mathbf{S}_{\frac{1}{t}}$ a t over-sampling operator, and I a Shannon-Whitakker interpolation operator. The discrete convolution by a gaussian is defined in (4.8). Since t is an integer, the t sub-sampling is trivial.

If the anti-aliasing filter size $\mathbf{c}_r(t)$ is too small, $\mathbf{u}_1$ and $\mathbf{u}_2$ can be very different. The right value of $\mathbf{c}_r(t)$ should be the smallest value permitting $\mathbf{u}_1 \approx \mathbf{u}_2$. Fig. 4.1 shows $\mathbf{u}_1$ and $\mathbf{u}_2$ with $t = 4$ and plots their mean square error MSE$(\mathbf{u}_1, \mathbf{u}_2)$. An anti-aliasing filter with $\mathbf{c}_r(4) = 1.2$ is clearly not broad enough: $\mathbf{u}_2$ presents strong ringing artifacts. The ringing artifact is instead hardly noticeable with $\mathbf{c}_r(4) = 2.4$. The value $\mathbf{c}_r(4) \simeq 2.4$ is a good visual candidate, and this choice is confirmed by the curve showing that MSE$(\mathbf{u}_1, \mathbf{u}_2)$ decays rapidly until $\mathbf{c}_r(4)$ gets close to 2.4, and is stable and small thereafter. By (4.10), this value of $\mathbf{c}_r$ yields $\mathbf{c}_a = 0.62$. This value has been confirmed by experiments on ten digital images. A doubt can be cast on this experiment, however: Its result slightly depends on the assumption that the initial blur on $\mathbf{u}$ is equal to $\mathbf{c}_a$.

In a second experiment, $\mathbf{c}_a$ has been evaluated directly by using a binary image $\mathbf{u}_0$ that does not contain any blur. As illustrated in Fig. 4.2, $\mathbf{u}_0$ is obtained by binarizing the digital test image Lena (Fig. 4.1), the threshold being the median value of Lena. Since $\mathbf{u}_0$ is now blur-free, we can compare for several values of $\mathbf{c}_a$ and for $t = 4$, which is large enough, the digital images

$$\mathbf{u}_1 =: \mathbf{G}_{t\mathbf{c}_a} I\mathbf{u} = \mathbf{S}_1 G_{t\mathbf{c}_a} I\mathbf{u} \quad \text{and} \quad \mathbf{u}_2 =: (\mathbf{S}_{1/t} I)\mathbf{S}_t \mathbf{G}_{t\mathbf{c}_a} I\mathbf{u} = (\mathbf{S}_{1/t} I)\mathbf{S}_t G_{t\mathbf{c}_a} I\mathbf{u},$$

As shown in Fig. 4.2, $\mathbf{c}_a = 0.6$ is the smallest value ensuring no visual ringing in $\mathbf{u}_2$. Under this value, for example for $\mathbf{c}_a = 0.3$, clear ringing artifacts are present in $\mathbf{u}_2$. That $\mathbf{c}_a = 0.6$ is the correct value is confirmed by the MSE$(\mathbf{u}_1, \mathbf{u}_2)$ curve showing that the mean square error decays rapidly until $\mathbf{c}_a$ goes down to 0.6, and is stable and small thereafter. The result, confirmed in ten experiments with different initial images, is consistent with the value obtained in the first experimental setting.

4.2.1 Discrete sampling

If $\mathbf{u}$ is digital and therefore only defined on $\mathbb{Z}^2$ and if δ is an integer, then one can define any sub- or over-sampling operations on $\mathbf{u}$. But this requires interpolating $\mathbf{u}$ first.

Definition 4.8. *Thus we define a digital re-sampling operator by*

$$\mathcal{S}_\delta \mathbf{u} =: \mathbf{S}_\delta I\mathbf{u}. \tag{4.11}$$

$\mathcal{S}_\delta$ is a discrete filter. If $\delta < 1$ $\mathcal{S}_\delta$ is an over-sampling, and it is invertible. If $\delta > 1$ it is an sub-sampling, and may be not invertible.

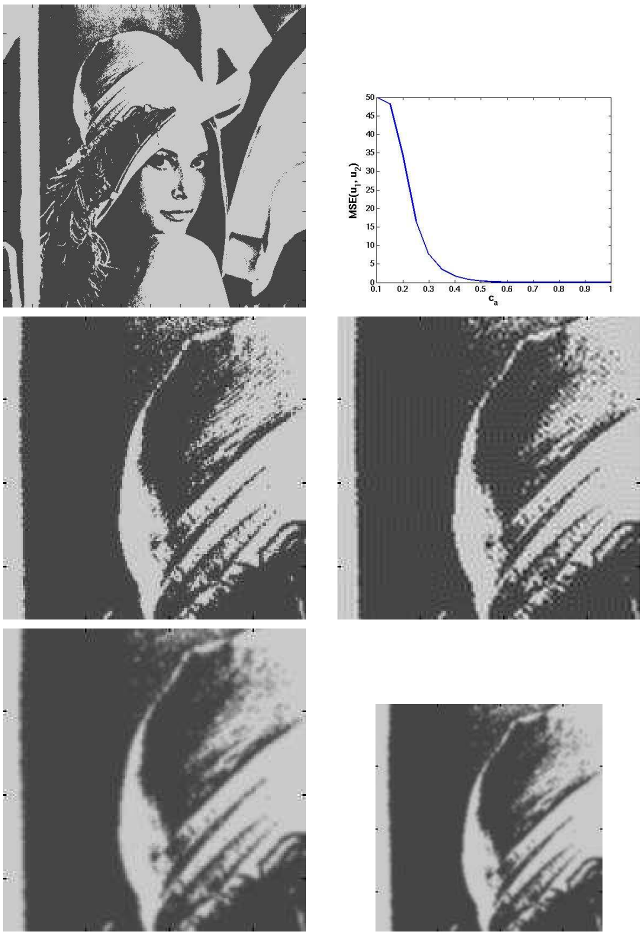

Figure 4.2: Top left: $\mathbf{u}$. Top right: MSE$(\mathbf{u}_1, \mathbf{u}_2)$ vs $\mathbf{c}_a$. Middle (from left to right): $\mathbf{u}_1$ and $\mathbf{u}_2$ with $\mathbf{c}_a = 0.3$. MSE$(\mathbf{u}_1, \mathbf{u}_2)$=7.46. Bottom (from left to right): $\mathbf{u}_1$ and $\mathbf{u}_2$ with $\mathbf{c}_a = 0.6$. MSE$(\mathbf{u}_1, \mathbf{u}_2)$=0.09.

Exercise 4.6. Show that if $\delta < 1$, then $\mathcal{S}_{\delta^{-1}}\mathcal{S}_\delta = Id$. What can happen if $\delta > 1$? ■

Over-sampling can be interpreted as *a zoom in.* A zoom in is not the same as a *blow up.* Blow up is a photographic term involving the use of a system of lenses increasing the image resolution : it permits to see more details of the observed object. Zoom in is instead is a digital term. Being just an interpolation, it adds no detail to the image. The next proposition confirms that it is just an image enlargement.

Proposition 4.9. *For every* $\gamma \leq 1$,

$$I\mathcal{S}_\gamma \mathbf{u} = H_\gamma I\mathbf{u}. \tag{4.12}$$

Proof. $I\mathbf{u}$ is well sampled, with spectrum in $[0, 2\pi]^2$. Thus since $\gamma < 1$, $I\mathcal{S}_\gamma\mathbf{u}$ is over-sampled: it has spectrum in $[0, 2\pi\gamma]^2$. Thus $I\mathcal{S}_\gamma\mathbf{u}$ is band limited, as $H_\gamma I\mathbf{u}$. Since

$$I\mathcal{S}_\gamma\mathbf{u}(n_1,n_2) = \mathcal{S}_\gamma\mathbf{u}(n_1,n_2) = I\mathbf{u}(n_1\gamma,n_2\gamma) \text{ and } H_\gamma I\mathbf{u}(n_1,n_2) = I\mathbf{u}(n_1\gamma,n_2\gamma),$$

both functions have the same $\mathbb{Z}^2$ samples and therefore coincide. □

Corollary 4.10. *If* $\gamma \leq 1$, *then*

$$\mathcal{S}_\beta\mathcal{S}_\gamma = \mathcal{S}_{\beta\gamma}. \tag{4.13}$$

Proof. using once (4.12) and twice (4.11),

$$\mathcal{S}_\beta\mathcal{S}_\gamma\mathbf{u} = \mathbf{S}_\beta I\mathcal{S}_\gamma\mathbf{u} = \mathbf{S}_\beta\mathbf{H}_\gamma I\mathbf{u} = \mathbf{S}_{\beta\gamma}I\mathbf{u} = \mathcal{S}_{\beta\gamma}\mathbf{u}.$$

□

Proposition 4.11. *A discrete commutation formula : Assume* $\mathbf{u}$ *is a digital image. Then for* $\gamma < 1$,

$$\mathbf{G}_\beta\mathcal{S}_\gamma\mathbf{u} = \mathcal{S}_\gamma\mathbf{G}_{\beta\gamma}\mathbf{u}. \tag{4.14}$$

Proof.

$$\mathbf{G}_\beta\mathcal{S}_\gamma\mathbf{u} \overset{(4.8)}{=} \mathbf{S}_1(G_\beta I\mathcal{S}_\gamma\mathbf{u}) \overset{(4.12)}{=} \mathbf{S}_1(G_\beta H_\gamma I\mathbf{u})$$

$$\overset{(9.2)}{=} \mathbf{S}_1(H_\gamma(G_{\beta\gamma}I\mathbf{u})) \overset{(4.1)}{=} \mathbf{S}_\gamma(G_{\beta\gamma}I\mathbf{u}) \overset{(4.2)}{=} \mathbf{S}_\gamma I\mathbf{S}_1(G_{\beta\gamma}I\mathbf{u}) \overset{(4.8,4.11)}{=} \mathcal{S}_\gamma\mathbf{G}_{\beta\gamma}\mathbf{u}.$$

Notice that we use $I\mathbf{S}_1 u = u$ with $u = G_{\beta\gamma}I\mathbf{u}$. Indeed, this last function is well sampled, because Iu is. □

Chapter 5

The SIFT Method

This chapter is devoted to Lowe's *Scale-Invariant Feature Transform* (SIFT [151]), a very efficient image comparison method. The initial goal of the SIFT method is to compare two images (or two image parts) that can be deduced from each other (or from a common one) by a rotation, a translation, and a zoom. The method turned out to be also robust to large enough changes in view point angle, which explains its success. This method uses as fundamental tool the heat equation or, in other terms, the linear scale space. The heat equation is used to simulate all zooms out of both images that have to be compared. Indeed, these images may contain similar objects taken at different distances. But at least two of the simulated zoomed in images should contain these objects at the same apparent distance. This is the principal ingredient of the SIFT method, but other invariance requirements must be addressed as well.

Sect. 5.2 gives a detailed description of the SIFT shape encoding method. Sect. 5.4 proves mathematically that the SIFT method indeed computes translation, rotation and scale invariants. This proof is correct under the main assumption that image blur can be assumed to be gaussian, and that images with a gaussian blur larger than 0.6 (SIFT takes 0.8) are approximately (but accurately) well-sampled and can therefore be interpolated. Chapter. 4.2 checked the validity of this crucial gaussian blur assumption.

5.1 Introduction

Image comparison is a fundamental step in many computer vision and image processing applications. A typical image matching method first detects points of interest, then selects a region around each point, and finally associates with each region a descriptor. Correspondences between two images may then be established by matching the descriptors of both images.

In the SIFT method, stable points of interest are supposed to lie at extrema of the Laplacian of the image in the image scale-space representation. The scale-space representation introduces a smoothing parameter σ. Images u_0 are smoothed at several scales to obtain $w(\sigma, x, y) =: (G_\sigma * u_0)(x, y)$, where we use the parameterization of the gaussian by its standard deviation σ,

$$G_\sigma(x, y) = G(\sigma, x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}.$$

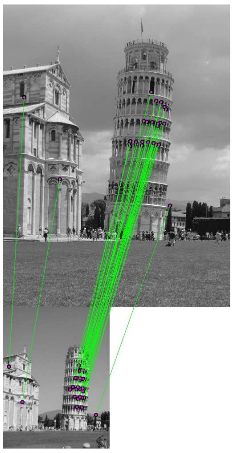

Figure 5.1: A result of the SIFT method, using an outliers elimination method [206]. Pairs of matching points are connected by segments.

Taking apart all sampling issues and several thresholds whose aim it is to eliminate unreliable features, the whole method can be summarized in one single sentence:

One sentence description *The SIFT method computes scale-space extrema* (σ_i, x_i, y_i) *of the space Laplacian of* $w(\sigma, x, y)$*, and then samples for each one of these extrema a square image patch whose origin is* (x_i, y_i)*, whose* x*-direction is one of the dominant gradients around* (x_i, y_i)*, and whose sampling rate is* $\sqrt{\sigma_i^2 + \mathbf{c}^2}$.

The constant $\mathbf{c} \simeq 0.8$ is the tentative standard deviation of the image blur. The resulting samples of the digital patch at scale σ_i are encoded by their gradient direction, which is invariant under nondecreasing contrast changes. This accounts for the robustness of the method to illumination changes. In addition, only local histograms of the direction of the gradient are kept, which accounts for the robustness of the final descriptor to changes of view angle (see Fig. 5.5).

Figs 5.1 and 5.6 show striking examples of the method scale invariance. Lowe claims that 1) his descriptors are invariant with respect to translation, scale and rotation, and that 2) they provide a robust matching across a substantial range of affine distortions, change in 3D viewpoint, addition of noise, and change in illumination. In addition, being local, they are robust to occlusion. Thus they match all requirements for shape recognition algorithms except one: they are not really affine invariant but only robust to moderate affine distortions.

5.2 A Short Guide to SIFT Encoding

The SIFT encoding algorithm consists of four steps: detection of scale-space extrema (Sect. 5.2.1), accurate localization of key points (Sect. 5.2.2), and descriptor construction (Sect. 5.2.3).

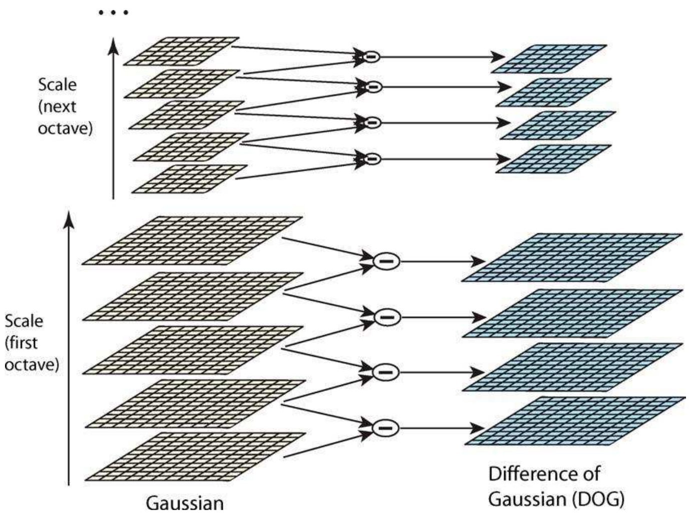

Figure 5.2: Gaussian pyramid for key points extraction (from [151])

5.2.1 Scale-Space Extrema

Following a classical paradigm, stable points of interest are supposed to lie at extrema of the Laplacian of the image in the image scale-space representation. We recall that the scale-space representation introduces a smoothing parameter σ, the scale, and convolves the image with Gaussian functions of increasing standard deviation σ. By a classical approximation inspired from psychophysics [164], the Laplacian of the Gaussian is replaced by a Difference of Gaussians at different scales (DOG). Extrema of the Laplacian are then replaced by extrema of DOG functions: $\mathbb{D}(\sigma, x, y) = w(k\sigma, x, y) - w(\sigma, x, y)$, where k is a constant multiplicative factor. Indeed, it is easy to show that $\mathbb{D}(\sigma, x, y)$ is an approximation of the Laplacian:

$$\mathbb{D}(\sigma, x, y) \approx (k-1)\sigma^2(\Delta G_\sigma * u_0)(x, y).$$

In the terms of David Lowe:

> *The factor $(k-1)$ in the equation is constant over all scales and therefore does not influence extrema location. The approximation error will go to zero as k goes to 1, but in practice we have found that the approximation has almost no impact on the stability of extrema detection or localization for even significant differences in scale, such as $k = \sqrt{2}$.*

To be more specific, quoting Lowe again:

$$\mathbb{D}(\sigma, x, y) =: (G(k\sigma, x, y) - G(\sigma, x, y)) * u_0(x, y) = w(k\sigma, x, y) - w(\sigma, x, y)$$

> *The relationship between D and $\sigma^2 \Delta G$ can be understood from the heat diffusion equation (parameterized in terms of σ rather than*

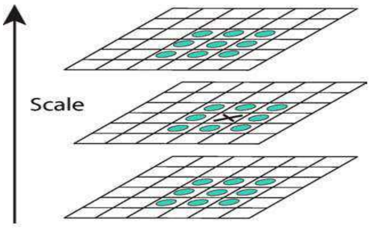

Figure 5.3: Neighborhood for the location of key points (from [151]). Local extrema are detected by comparing each sample point in $\mathbb{D}$ with its eight neighbors at scale σ and its nine neighbors in the scales above and below.

> *the more usual $t = \sigma^2$):*
>
> $$\frac{\partial G}{\partial \sigma} = \sigma \Delta G.$$
>
> *From this, we see that ΔG can be computed from the finite difference approximation to $\partial G/\partial \sigma$, using the difference of nearby scales at $k\sigma$ and σ:*
>
> $$\sigma \Delta G = \frac{\partial G}{\partial \sigma} \approx \frac{G(k\sigma, x, y) - G(\sigma, x, y)}{k\sigma - \sigma}$$
>
> *and therefore,*
>
> $$G(k\sigma, x, y) - G(\sigma, x, y) \approx (k-1)\sigma^2 \Delta G.$$
>
> *This shows that when the difference-of-Gaussian function has scales differing by a constant factor it already incorporates the σ^2 scale normalization required for the scale-invariant Laplacian.*

This leads to an efficient computation of local extrema of $\mathbb{D}$ by exploring neighborhoods through a Gaussian pyramid ; see Figs. 5.2 and 5.3.

Exercise 5.1. Show that the gaussian G_σ parameterized by its standard deviation σ satisfies as stated by Lowe the time-dependent heat equation $\frac{\partial G}{\partial \sigma} = \sigma \Delta G$. ■

5.2.2 Accurate Key Point Detection

In order to achieve sub-pixel accuracy, the interest point position is slightly corrected thanks to a quadratic interpolation. Let us call $\mathbf{x}_0 =: (\sigma_0, x_0, y_0)$ the current detected point in scale space, which is known up to the (rough) sampling accuracy in space and scale. Notice that all points $\mathbf{x} = (\sigma, x, y)$ here are scale-space coordinates. Let us call $\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{y}$ the real extremum of the DOG function. The Taylor expansion of $\mathbb{D}$ yields

$$\mathbb{D}(\mathbf{x}_0 + \mathbf{y}) = \mathbb{D}(\mathbf{x}_0) + (D\mathbb{D})(\mathbf{x}_0) \cdot \mathbf{y} + \frac{1}{2}\left(D^2\mathbb{D}\right)(\mathbf{x}_0)(\mathbf{y}, \mathbf{y}) + o(\|\mathbf{y}\|^2),$$

where $\mathbb{D}$ and its derivatives are evaluated at an interest point and $\mathbf{y}$ denotes an offset from this point. Since interest points are extrema of $\mathbb{D}$ in scale space, setting the derivative to zero gives:

$$\mathbf{y} = -\left(D^2\mathbb{D}(\mathbf{x}_0)\right)^{-1}(D\mathbb{D}(\mathbf{x}_0)), \tag{5.1}$$

which is the sub-pixel correction for a more accurate position of the key point of interest.

Exercise 5.2. Check that (5.1) is a point where the gradient of $\mathbb{D}$ vanishes. ■

Since points with low contrast are sensitive to noise, and since points that are poorly localized along an edge are not reliable, a filtering step is called for. Low contrast points are handled through a simple thresholding step. Edge points are swept out following the Harris and Stephen's interest points paradigm. Let H be the following Hessian matrix:

$$H = \begin{pmatrix} \mathbb{D}_{xx} & \mathbb{D}_{xy} \\ \mathbb{D}_{xy} & \mathbb{D}_{yy} \end{pmatrix}.$$

The reliability test is simply to assess whether the ratio between the larger eigenvalue and the smaller one is below a threshold r. This amounts to check:

$$\frac{\mathrm{Tr}(H)^2}{\mathrm{Det}(H)} < \frac{(r+1)^2}{r}. \tag{5.2}$$

This rules out standard edge points and puts points of interest at locations which are strong enough extrema, or saddle points.

Exercise 5.3. Explain why (5.2) is equivalent imposing that the ratio between the smaller eigenvalue and the larger eigenvalue of H is smaller than r. These eigenvalues are assumed to have the same sign. Why? ■

5.2.3 Construction of the SIFT descriptor

In order to extract rotation-invariant patches, an orientation must be assigned to each key point. Lowe proposes to estimate a semi-local average orientation for each key point. From each sample image L_σ, gradient magnitude and orientation is pre-computed using a 2×2 scheme. An orientation histogram is assigned to each key point by accumulating gradient orientations weighted by 1) the corresponding gradient magnitude and by 2) a Gaussian factor depending on the distance to the considered key point and on the scale. The precision of this histogram is 10 degrees. Peaks simply correspond to dominant directions of local gradients. Key points are created for each peak with similar magnitude, and the assigned orientation is refined by local quadratic interpolation of the histogram values.

Once a scale and an orientation are assigned to each key point, each keypoint is associated *a square image patch whose size is proportional to the scale and whose side direction is given by the assigned direction.* The next step is to extract from this patch *robust* information. Gradient samples are accumulated into orientation histograms summarizing the contents over 4×4 subregions surrounding the key point of interest. Each of the 16 subregions corresponds to a 8-orientations bins histogram, leading to a 128 element feature for each key point (see Fig. 5.5). Two modifications are made in order to reduce the effects of illumination changes: histogram values are thresholded to reduce importance of large gradients (in order to deal with a strong illumination change such as camera saturation), and feature vectors are normalized to unit length (making them invariant to affine changes in illumination).

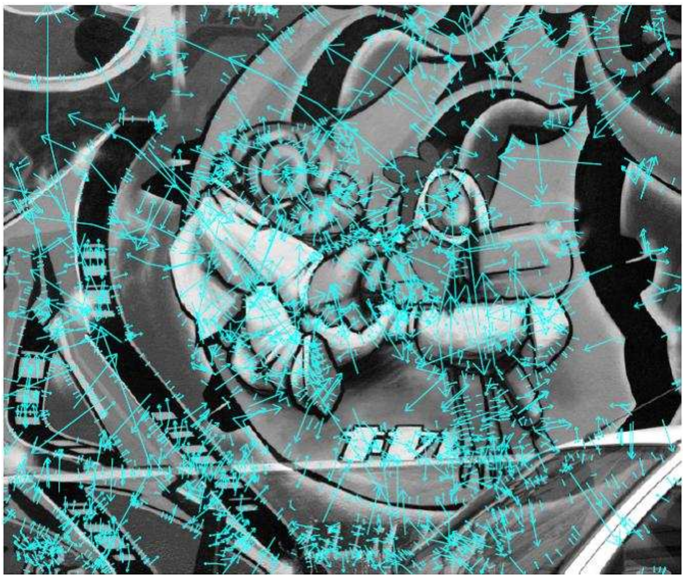

Figure 5.4: SIFT key points. The arrow starting point, length and the orientation signify respectively the key point position, scale, and dominant orientation. These features are covariant to any image similarity.

5.2.4 Final matching

The outcome is for each image, a few hundreds or thousands SIFT descriptors associated with as many key points. The descriptors of any image can be compared to the descriptors of any other image, or belonging to a database of descriptors built up from many images. The only remaining question is to decide when two descriptors match, or not. In the terms of Lowe again:

> The best candidate match for each keypoint is found by identifying its nearest neighbor in the database of keypoints from training images. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector. However, many features from an image will not have any correct match in the training database because they arise from background clutter or were not detected in the training images. Therefore, it would be useful to have a way to discard features that do not have any good match to the database. A global threshold on distance to the closest feature does not perform well, as some descriptors are much more discriminative than others. A more effective measure is obtained by comparing the distance of the closest neighbor to that of the second-closest neighbor. (...) This measure performs well because correct matches need to have the closest neighbor significantly closer than the closest incorrect match to achieve reliable matching. For false matches, there will likely be a number of other false matches within similar distances due to the high dimensionality of the feature space. We can think of the second-closest match as providing an estimate of the density of false matches within this portion of the feature space

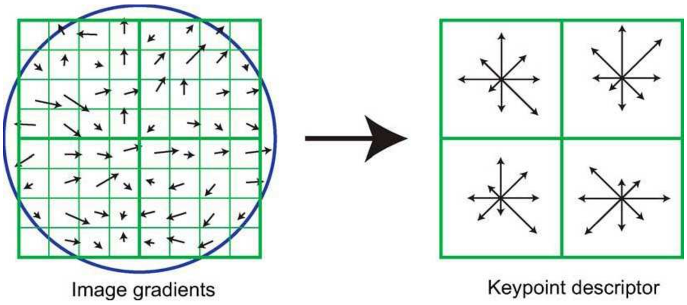

Figure 5.5: Example of a 2×2 descriptor array of orientation histograms (right) computed from an 8×8 set of samples (left). The orientation histograms are quantized into 8 directions and the length of each arrow corresponds to the magnitude of the histogram entry. (From [151])

> and at the same time identifying specific instances of feature ambiguity. (...) For our object recognition implementation, we reject all matches in which the distance ratio is greater than 0.8, which eliminates 90% of the false matches while discarding less than 5% of the correct matches.

5.3 Image acquisition model underlying SIFT

5.3.1 The camera model

We always work on the camera CCD plane, whose mesh unit is taken to be 1. We shall always assume that the camera pixels are indexed by $\mathbb{Z}^2$. The image sampling operator is therefore always $\mathbf{S}_1$. Our second assumption is that the digital initial image is well-sampled and obtained by a gaussian kernel. Thus, the digital image is $\mathbf{u} = \mathbf{S}_1 G_\delta A u_0$, where $\delta \geq \mathbf{c}$, $\mathbf{c} \simeq 0.6$ ensures well-sampling (see Chapter 4.2), and A is a similarity with positive determinant. (In fact Lowe's original paper assumes $\mathbf{c} \simeq 0.5$, which amounts to assume a slight under-sampling of the original image).

Definition 5.1. *We model all digital frontal images obtained from a given ideal planar object whose frontal infinite resolution image is u_0 as*

$$\mathbf{u}_0 =: \mathbf{S}_1 G_\delta A u_0 \tag{5.3}$$

where $\delta \geq \mathbf{c}$ and A is a $A = RH_\lambda \mathcal{T}$ is the composition of a translation and of a similarity.

So the possibility of aliasing (under-sampling, $\delta < \mathbf{c}$ is discarded). Taking into account the way the digital image is blurred and sampled in the SIFT method, we can now list the SIFT assumptions and formalize the method itself. The description is by far simpler if we do it without mixing in sampling issues. We need not mix them in, since the fact that images are well-sampled at all stages permits equivalently to describe all operations with the continuous images directly, and to deduce afterwards the discrete operators on samples. We refer to section 4.2.1 for this passage from continuous to discrete operations in the well-sampled world.

5.3.2 Condensed description of the SIFT method

1. There is an underlying infinite resolution bounded planar image u_0;
2. The initial digital image is $\mathbf{S}_1 G_\delta A u_0$ where $\delta \geq \mathbf{c}$, and $A = RH_\lambda \mathcal{T}$ is the composition of a rotation, a zoom, and a translation;
3. the SIFT method computes a sufficient scale-space sampling of $u(\sigma, \mathbf{x}) = (G_\sigma G_\delta A u_0)(\mathbf{x})$, and deduces by the Newton method the accurate location or key points defined as extrema in scale-space of the spatial image Laplacian, $\Delta u(\sigma; \mathbf{x})$;
4. The blurred $u(\sigma, \cdot)$ image is then re-sampled around each characteristic point with sampling mesh $\sqrt{\sigma^2 + \mathbf{c}^2}$;
5. the directions of the sampling axes are fixed by a dominant direction of the gradient of $u(\sigma, \cdot)$ in a neighborhood, with size proportional to $\sqrt{\sigma^2 + \mathbf{c}^2}$ around the characteristic point;
6. the rest of the operations in the SIFT method is a contrast invariant encoding of the samples around each characteristic point. It is not needed for the discussion to follow.

5.4 Scale and SIFT: consistency of the method

In this section, in conformity with the SIFT model of Sect. 5.3.2, the digital image is a frontal view of an infinite resolution ideal image u_0. In that case, $A = H\mathcal{T}R$ is the composition of a homothety H, a translation $\mathcal{T}$ and a rotation R. Thus the digital image is $\mathbf{u} = \mathbf{S}_1 G_\delta H \mathcal{T} R u_0$, for some H, $\mathcal{T}$, R as above. Assuming that the image is not aliased boils down, by the experimental results of Sect. 4.2, to assuming $\delta \geq 0.8$.

Consider $\mathcal{T}$ an arbitrary image translation, R an arbitrary image rotation, H_λ an arbitrary image homothety, G an arbitrary gaussian convolution, D the gradient and Δ the Laplacian, all applied to continuous images. We say that there is strong commutation of two of these operators if we can exchange the order of their application to any image. We say that there is weak commutation between two of these operators if we can exchange their order by changing one of the parameters of one of the operators. For example we have $R\mathcal{T} = \mathcal{T}'R$, meaning that given R and $\mathcal{T}$ there is $\mathcal{T}'$ such that the former relation occurs. The next lemma is straightforward.

Lemma 5.2. *All of the aforementioned operators weakly commute, with the following exceptions: R and G commute strongly, $DH_\lambda = \lambda H_\lambda D$, $\Delta H_\lambda = \lambda^2 H_\lambda \Delta$, and D and Δ do not commute.*

Exercise 5.4. Check all of the mentioned commutations and give their exact formula. There are six kinds of operators: translations, rotations, homotheties, gaussian convolutions, gradients, and Laplacians. Thus, there are 15 verifications to make. ■

Lemma 5.3. *For any rotation R and any translation $\mathcal{T}$, the SIFT descriptors of $\mathbf{S}_1 G_\delta H \mathcal{T} R u_0$ are identical to those of $\mathbf{S}_1 G_\delta H u_0$.*

Proof. By the weak commutation of translations and rotations with all other operators (Lemma 5.2), the SIFT descriptors of a rotated or translated image are identical to those of the original. Indeed, the set of scale space Laplacian extrema is covariant to space translations and rotations. The normalization process for each SIFT descriptor situates the origin at each extremum in turn, thus canceling the translation. The local sampling grid defining the SIFT patch has axes given by peaks in its gradient direction histogram. Such peaks are translation invariant and rotation covariant. Thus, the normalization of the direction also cancels the rotation. □

Lemma 5.4. *Let $\mathbf{u}$ and $\mathbf{v}$ be two digital images that are frontal snapshots of the same continuous flat image u_0, $\mathbf{u} = \mathbf{S}_1 G_\beta H_\lambda u_0$ and $\mathbf{v} =: \mathbf{S}_1 G_\delta H_\mu u_0$, taken at different distances, with different gaussian blurs and possibly different sampling rates. Let $w(\sigma, \mathbf{x}) = (G_\sigma u_0)(\mathbf{x})$ denote the scale space of u_0. Then the scale spaces of $u = G_\beta H_\lambda u_0$ and $v = G_\delta H_\mu u_0$ are*

$$u(\sigma, \mathbf{x}) = w(\lambda\sqrt{\sigma^2+\beta^2}, \lambda\mathbf{x}) \quad \textit{and} \quad v(\sigma, \mathbf{x}) = w(\mu\sqrt{\sigma^2+\delta^2}, \mu\mathbf{x}).$$

If $(s_0, \mathbf{x}_0)$ is a key point of w satisfying $s_0 \geq \max(\lambda\beta, \mu\delta)$, then it corresponds to a key point of u at the scale σ_1 such that $\lambda\sqrt{\sigma_1^2+\beta^2} = s_0$, whose SIFT descriptor is sampled with mesh $\sqrt{\sigma_1 + \mathbf{c}^2}$. In the same way $(s_0, \mathbf{x}_0)$ corresponds to a key point of v at scale σ_2 such that $s_0 = \mu\sqrt{\sigma_2^2+\delta^2}$, whose SIFT descriptor is sampled with mesh $\sqrt{\sigma_2^2 + \mathbf{c}^2}$.

Proof. The interpolated initial images are by (4.2)

$$u =: \boldsymbol{I}\mathbf{S}_1 G_\beta H_\lambda u_0 = G_\beta H_\lambda u_0 \text{ and } v =: \boldsymbol{I}\mathbf{S}_1 G_\delta H_\mu u_0 = G_\delta H_\mu u_0.$$

Computing the scale-space of these images amounts to convolve these images for every $\sigma > 0$ with G_σ, which yields, using the commutation relation (9.2) and the semigroup property (9.1):

$$u(\sigma, \cdot) = G_\sigma G_\beta H_\lambda u_0 = G_{\sqrt{\sigma^2+\beta^2}} H_\lambda u_0 = H_\lambda G_{\lambda\sqrt{\sigma^2+\beta^2}} u_0.$$

By the same calculation, this function is compared by SIFT with

$$v(\sigma, \cdot) = H_\mu G_{\mu\sqrt{\sigma^2+\delta^2}} u_0.$$

Set $w(s, \mathbf{x}) =: G_s u_0$. Then the scale spaces compared by SIFT are

$$u(\sigma, \mathbf{x}) = w(\lambda\sqrt{\sigma^2+\beta^2}, \lambda\mathbf{x}) \quad \text{and} \quad v(\sigma, \mathbf{x}) = w(\mu\sqrt{\sigma^2+\delta^2}, \mu\mathbf{x}).$$

Let us consider an extremal point $(s_0, \mathbf{x}_0)$ of the Laplacian of the scale space function w. If $s_0 \ \geq \max(\lambda\beta, \mu\delta)$, an extremal point occurs at scales σ_1 for the Laplacian of $u(\sigma, \mathbf{x})$ and at scale σ_2 for the Laplacian of $v(\sigma, \mathbf{x})$ satisfying

$$s_0 = \lambda\sqrt{\sigma_1^2+\beta^2} = \mu\sqrt{\sigma_2^2+\delta^2}. \tag{5.4}$$

□

Theorem 5.5. *Let* $\mathbf{u}$ *and* $\mathbf{v}$ *be two digital images that are frontal snapshots of the same continuous flat image* u_0*,* $\mathbf{u} = \mathbf{S}_1 G_\beta H_\lambda \mathcal{T} R u_0$ *and* $\mathbf{v} =: \mathbf{S}_1 G_\delta H_\mu u_0$*, taken from arbitrary distances, with possibly different camera gaussian blurs, with an arbitrary camera translation parallel to its focal plane, and an arbitrary rotation around its optical axe. Without loss of generality, assume* $\lambda \leq \mu$*. Then if the camera blurs are standard (*$\beta = \delta = \mathbf{c}$*), each SIFT descriptor of* $u = I\mathbf{u}$ *is identical to some SIFT descriptor of* $v = I\mathbf{v}$*. If* $\beta \neq \delta$ *(or* $\beta = \delta \neq \mathbf{c}$*), the SIFT descriptors of* u *become (quickly) similar to SIFT descriptors of* v *when their scales grow, namely as soon as* $\frac{\sigma_1}{\max(\mathbf{c},\beta)} \gg 1$ *and* $\frac{\sigma_2}{\max(\mathbf{c},\delta)} \gg 1$.

Proof. By the result of Lemma 5.3, we can neglect the effect of translations and rotations. Therefore assume w.l.o.g. that the images under comparison are as in Lemma 5.4. Assume a key point $(s_0, \mathbf{x}_0)$ of w has scale $s_0 \geq \max(\lambda\beta, \mu\delta)$. This key point has a sampling rate proportional to s_0. There is a corresponding key point $(\sigma_1, \frac{\mathbf{x}_0}{\lambda})$ for u with sampling rate $\sqrt{\sigma_2^2 + \mathbf{c}^2}$ and a corresponding key point $(\sigma_2, \frac{\mathbf{x}_0}{\mu})$ with sampling rate $\sqrt{\sigma_2^2 + \mathbf{c}^2}$ for $\mathbf{v}$. To have a common reference for these sampling rates, it is convenient to refer to the corresponding sampling rates for $w(s_0, \mathbf{x})$, which are $\lambda\sqrt{\sigma_1^2 + \mathbf{c}^2}$ for the SIFT descriptors of u at scale σ_1, and $\mu\sqrt{\sigma_2^2 + \mathbf{c}^2}$ for the descriptors of $\mathbf{v}$ at scale σ_2. Thus the SIFT descriptors of u and v for $\mathbf{x}_0$ will be identical if and only if $\lambda\sqrt{\sigma_1^2 + \mathbf{c}^2} = \mu\sqrt{\sigma_2^2 + \mathbf{c}^2}$. Now, we have $\lambda\sqrt{\sigma_1^2 + \beta^2} = \mu\sqrt{\sigma_2^2 + \delta^2}$, which implies $\lambda\sqrt{\sigma_1^2 + \mathbf{c}^2} = \mu\sqrt{\sigma_2^2 + \mathbf{c}^2}$ if and only if

$$\lambda^2\beta^2 - \mu^2\delta^2 = (\lambda^2 - \mu^2)\mathbf{c}^2. \tag{5.5}$$

Since λ and μ are proportional to camera distances to the observed object u_0, they are arbitrary and generally different. Thus, the only way to ensure (5.5) is to have $\beta = \delta = \mathbf{c}$, which means that the blurs of both images (or of both cameras) are ideal and gaussian. In any case, $\beta = \delta = \mathbf{c}$ does imply that the SIFT descriptors of both images are identical.

The second statement is straightforward: If σ_1 and σ_2 are large enough with respect to β, δ and $\mathbf{c}$, the relation $\lambda\sqrt{\sigma_1^2 + \beta^2} = \mu\sqrt{\sigma_2^2 + \delta^2}$, implies $\lambda\sqrt{\sigma_1^2 + \mathbf{c}^2} \simeq \mu\sqrt{\sigma_2^2 + \mathbf{c}^2}$. □

The almost perfect scale invariance of SIFT stated in Theorem 5.5 is illustrated by the striking example of Fig. 5.6. The 28 SIFT key points of a very small digital image $\mathbf{u}$ are compared to the 86 key points obtained by zooming in $\mathbf{u}$ by a 32 factor: The resulting digital image is the digital image $\mathbf{v} = \mathbf{S}_{\frac{1}{32}} I\mathbf{u}$, again obtained by zero-padding. For better observability, both images are displayed with the same size by enlarging the pixels of $\mathbf{u}$. Almost each key point (22 out of 28) of $\mathbf{u}$ finds its counterpart in $\mathbf{v}$. 22 matches are detected between the descriptors as shown on the right. If we trust Theorem 5.5, all descriptors of $\mathbf{u}$ should have been retrieved in $\mathbf{v}$. This does not fully happen for two reasons. First, the SIFT method thresholds (not taken into account in the theorem) eliminate many potential key points. Second, the zero-padding interpolation giving $\mathbf{v}$ is imperfect near the image boundaries.

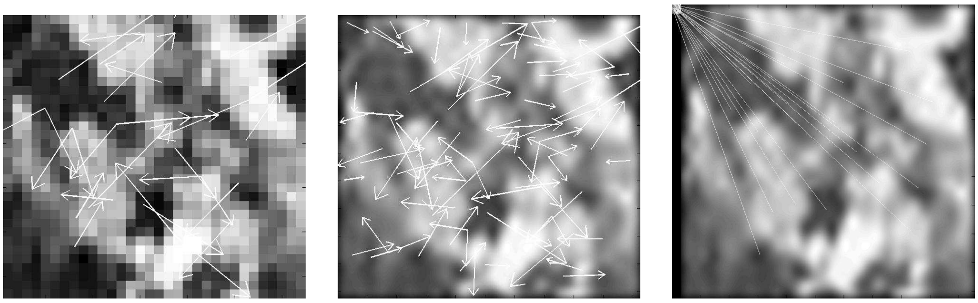

Figure 5.6: Scale invariance of SIFT, an illustration of Theorem 5.5. Left: a very small digital image $\mathbf{u}$ with its 28 key points. For the conventions to represent key points and matches, see the comments in Fig. 5.4. Middle: this image is over sampled by a 32 factor to $\mathbf{v} = \mathbf{S}_{\frac{1}{32}} I\mathbf{u}$. It has 86 key points. Right: 22 matches found between $\mathbf{u}$ and $\mathbf{S}_{\frac{1}{32}} I\mathbf{u}$.

5.5 Exercises

Exercise 5.5. The aim of the exercise is to explain why the experiment of Fig. 5.6 works, and to illustrate Theorem 5.5. The digital zoom in by a factor λ is nothing but the discrete over-sampling operator $\mathcal{S}_{\frac{1}{\lambda}}$ with sampling step $\frac{1}{\lambda}$, defined in (4.11). Here, $\lambda = 32$. In the experiment an original digital image $\mathbf{u} = \mathbf{S}_1 G_\delta u$ is zoomed into $\mathbf{v} = \mathcal{S}_{\frac{1}{\lambda}}\mathbf{u}$.

1) Using the definition of the discrete zoom and the right commutation relations given in this chapter and in the former one (give their numbers), show that

$$\mathbf{v} = \mathbf{S}_{\frac{1}{\lambda}} G_\delta I\mathbf{u} = \mathbf{S}_1 G_{\lambda\delta} H_{\frac{1}{\lambda}} I\mathbf{u}.$$

2) Is $\mathbf{v}$ well-sampled if $\mathbf{u}$ was?

3) By applying carefully Theorem 5.5, assuming that $\delta \simeq \mathbf{c}$, discuss why SIFT manages to match SIFT descriptors of $\mathbf{u}$ and $\mathbf{v}$. ■

5.6 Comments and references

Many variations exist on the computation of interest points, following the pioneering work of Harris and Stephens [105]. The Harris-Laplace and Hessian-Laplace region detectors [172, 175] are invariant to rotation and scale changes. Some moment-based region detectors [146, 25] including Harris-Affine and Hessian-Affine region detectors [173, 175], an edge-based region detector [245], an intensity-based region detector [245], an entropy-based region detector [124], and two independently developed level line-based region detectors MSER ("maximally stable extremal region") [168] and LLD ("level line descriptor") [187, 189, 191] are designed to be invariant to affine transformations. These two methods stem from the Monasse image registration method [179] that used well contrasted extremal regions to register images. MSER is the most efficient one and has shown better performance than other affine invariant detectors [177]. However, as pointed out in [151], no known detector is actually fully affine invariant: All of them start with initial feature scales and locations selected in a non-affine invariant manner. The difficulty comes from the scale change from an image to another: This change of scale is actually an under-sampling, which means that the images differ by a blur.

In his milestone paper [151], Lowe has addressed this central problem and has proposed the so called scale-invariant feature transform (SIFT) descriptor, that is invariant to image translations and rotations, to scale changes (blur), and robust to illumination changes. It is also surprisingly robust to large enough orientation changes of the viewpoint (up to 60 degrees). Based on the scale-space theory [145], the SIFT procedure simulates all gaussian blurs and normalizes local patches around scale covariant image key points that are Laplacian extrema. A number of SIFT variants and extensions, including PCA-SIFT [126] and gradient location-orientation histogram (GLOH) [176], that claim to have better robustness and distinctiveness with scaled-down complexity have been developed ever since [85, 143]. Demonstrated to be superior to other descriptors [174, 176], SIFT has been popularly applied for scene recognition [79, 182, 217, 251, 96, 226] and detection [86, 194], robot localization [28, 196, 120], image registration [264], image retrieval [104], motion tracking [247, 128], 3D modeling and reconstruction [211, 252], building panoramas [3, 34], or photo management [263, 141, 51].

As pointed out by several benchmarks, the robustness and repeatability of the SIFT descriptors outperforms other methods. However, such benchmarks mix three very different criteria that, in our opinion, should have been discussed separately. The first one is the formal real invariance of each method when all thresholds have been eliminated. This real invariance has been proved here for SIFT. The second criterion is the practical validity of the sampling method used in SIFT, that has been again checked in Chapter 4.2. The last criterion is the clever fixing of several thresholds in the SIFT method ensuring robustness, repeatability, and a low false alarm rate. This one has been extensively tested and confirmed in previous benchmark papers (see also the very recent and complete report [80]). We think, however, that the success of SIFT in these benchmarks is primarily due to its full scale invariance.

Chapter 6

Linear Scale Space and Edge Detection

The general analysis framework in which an image is associated with smoothed versions of itself at several scales is called *scale space.* Following the results of Chapter 3, a linear scale space must be performed by applying the heat equation to the image. The main aim of this smoothing is to find out *edges* in the image. We shall first explain this doctrine. In the second section, we discuss experiments and several serious objections to such an image representation.

6.1 The edge detection doctrine

One of the uses of linear theory in two dimensions is *edge detection.* The assumption of the edge detection doctrine is that relevant information is contained in the traces produced in an image by the apparent contours of physical objects. If a black object is photographed against a white background, then one expects the silhouette of the object in the image to be bounded by a closed curve across which the light intensity u_0 varies strongly. We call this curve an *edge.* At first glance, it would seem that this edge could be detected by computing the gradient Du_0, since at a point $\mathbf{x}$ on the edge, $|Du_0(\mathbf{x})|$ should be large and $Du(\mathbf{x})$ should point in a direction normal to the boundary of the silhouette. It would therefore appear that finding edges amounts to computing the gradient of u_0 and determining the points where the gradient is large. This conclusion is unrealistic for two reasons:

(a) There may be many points where the gradient is large due to small oscillations in the image that are not related to real objects. Recall that digital images are always noisy, and thus there is no reason to assume the existence or computability of a gradient.

(b) The points where the gradient exceeds a given threshold are likely to form regions and not curves.

As we emphasized in the Introduction, objection (a) is dealt with by smoothing the image. We associate with the image u_0 smoothed versions $u(t, \cdot)$, where the scale parameter t indicates the amount of smoothing. In the classical linear theory, this smoothing is done by convolving u_0 with the Gaussian G_t.

One way that objection (b) has been approached is by redefining edge points. Instead of just saying an edge point is a point $\mathbf{x}$ where $|Du_0(\mathbf{x})|$ exceeds a threshold, one requires the gradient to satisfy a maximal property. We illustrate this in one dimension. Suppose that $u \in C^2(\mathbb{R})$ and consider the points where $|u'(x)|$ attains a local maximum. At some of these points, the second derivative u'' changes sign, that is, $\text{sign}(u''(x-h)) \neq \text{sign}(u''(x+h))$ for sufficiently small h. These are the points where u'' *crosses zero*, and they are taken to be the edge points. Note that this criterion avoids classifying a point x as an edge point if the gradient is constant in an interval around x. Marr and Hildreth generalized this idea to two dimensions by replacing u'' with the Laplacian Δu, which is the only isotropic linear differential operator of order two that generalizes u'' [165]. Haralick's edge detector is different but in the same spirit [101]. Haralick gives up linearity and defines edge points as those points where the gradient has a local maximum in the direction of the gradient. In other words, an edge point $\mathbf{x}$ satisfies $g'(0) = 0$, where $g(t) = |Du(\mathbf{x} + tDu(\mathbf{x})|/|Du(\mathbf{x})|$. This implies that $D^2u(\mathbf{x})(Du(\mathbf{x}), Du(\mathbf{x})) = 0$ (see Exercise 6.2). We are now going to state these two algorithms formally. They are illustrated in Figures 6.2 and 6.3, respectively.

Algorithm 6.1 (Edge detection: Marr–Hildreth zero-crossings).

(1) Create the multiscale images $u(t, \cdot) = G_t * u_0$ for increasing values of t.

(2) At each scale t, compute all the points where $Du \neq 0$ and Δu changes sign. These points are called zero-crossings of the Laplacian, or simply zero-crossings.

(3) (Optional) Eliminate the zero-crossings where the gradient is below some prefixed threshold.

(4) track back from large scales to fine scales the "main edges" detected at large scales.

Algorithm 6.2 (Edge detection: The Haralick–Canny edge detector).

(1) As before, create the multiscale images $u(t, \cdot) = G_t * u_0$ for increasing values of t.

(2) At each scale t, find all points $\mathbf{x}$ where $Du(\mathbf{x}) \neq 0$ and $D^2u(\mathbf{x})(\mathbf{z}, \mathbf{z})$ crosses zero, $\mathbf{z} = Du/|Du|$. At such points, the function $s \mapsto u(\mathbf{x} + s\mathbf{z})$ changes from concave to convex, or conversely, as s passes through zero.

(3) At each scale t, fix a threshold $\theta(t)$ and retain as *edge points at scale t* only those points found above that satisfy $|Du(\mathbf{x})| > \theta(t)$. The backtracking step across scales is the same as for Marr–Hildreth.

In practice, edges are computed for a finite number of dyadic scales, $t = 2^n$, $n \in \mathbb{Z}$.

6.1.1 Discussion and critique

The Haralick–Canny edge detector is generally preferred for its accuracy to the Marr–Hildreth algorithm. Their use and characteristics are, however, essentially

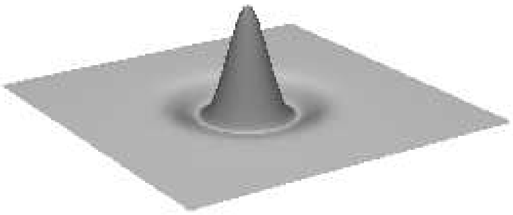

Figure 6.1: A three-dimensional representation of the Laplacian of the Gaussian. This convolution kernel, which is a wavelet, is used to estimate the Laplacian of an image at different scales of linear smoothing.

the same. There are also many variations—attempted improvements—of the algorithms we have described, and the following discussion adapts easily to these related edge detection schemes. The first thing to notice is that, by Proposition 2.5, $u(t,\cdot) = G_t * u_0$ is a C^∞ function for each $t > 0$ if $u_0 \in \mathcal{F}$. Thus we can indeed compute second order differential operators applied to $u(t,\cdot) = G_t * u_0$, $t > 0$. In the case of linear operators like the Laplacian or the gradient, the task is facilitated by the formula proved in the mentioned proposition. For example, we have $\Delta u(t,\mathbf{x}) = \Delta(G_t * u_0)(\mathbf{x}) = (\Delta G_t) * u_0(\mathbf{x})$, where in dimension two (Figure 6.1),

$$\Delta G_t(\mathbf{x}) = \frac{|\mathbf{x}|^2 - 4t}{16\pi t^3} \mathrm{e}^{-|\mathbf{x}|^2/4t}.$$

In the same way, Haralick's edge detector makes sense, because u is C^∞, at all points where $Du(\mathbf{x}) \neq 0$. If $Du(\mathbf{x}) = 0$, then $\mathbf{x}$ cannot be an edge point, since u is "flat" there. Thus, thanks to the filtering, there is no theoretical problem with computing edge points. There are, however, practical objections to these methods, which we will now discuss.

Linear scale space

The first serious problems are associated with the addition of an extra dimension: Having many images $u(t,\cdot)$ at different scales t confounds our understanding of the image and adds to the cost of computation. We no longer have an absolute definition of an edge. We can only speak of edges at a certain scale. Conceivably, a way around this problem would be to track edges across scales. In fact, it has been observed in experiments that the "main edges" persist under convolution as t increases, but they lose much of their spatial accuracy. On the other hand, filtering with a sharp low-pass filter, that is, with t small, keeps these edges in their proper positions, but eventually, as t becomes very small, even these main edges can be lost in the crowd of spurious edge signals due to noise and texture. The scale space theory of Witkin proposes to identify the main edges at some scale t and then to track them backward as t decreases [262]. In theory, it would seem that this method could give an accurate location of the main edges. In practice, any implementation of these ideas is computationally costly due to the problems involved with multiple thresholdings and following edges across scales. In fact, tracking edges across scales is incompatible with having thresholds for the gradients, since such thresholds may remove edges at

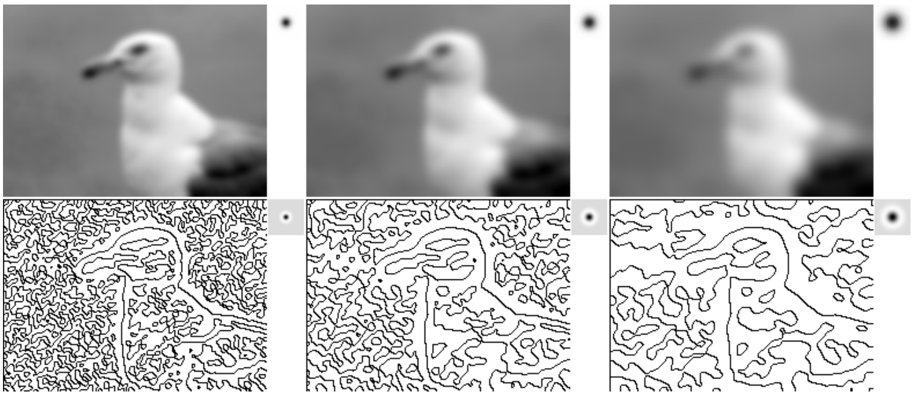

Figure 6.2: Zero-crossings of the Laplacian at different scales. This figure illustrates the original scale space theory as developed by David Marr [163]. To extract more global structure, the image is convolved with Gaussians whose variances are powers of two. One computes the Laplacian of the smoothed image and displays the lines along which this Laplacian changes sign: the zero-crossings of the Laplacian. According to Marr, these zero-crossings represent the "raw primal sketch" of the image, or the essential information on which further vision algorithms should be based. Above, left to right: the results of smoothing and the associated Gaussian kernels at scales 1, 2, and 4. Below, left to right: the zero-crossings of the Laplacian and the corresponding kernels, which are the Laplacians of the Gaussians used above.

certain scales and not at others. The conclusion is that one should trace all zero-crossings across scales without considering whether they are true edges or not. This makes matching edges across scales very difficult. For example, experiments show that zero-crossings of sharp edges that are sparse at small scales are no longer sparse at large scales. (Figure 6.4 shows how zero-crossings can be created by linear smoothing.) The Haralick–Canny detector suffers from the same problems, as is well demonstrated by experiments.

Other problems with linear scale space are illustrated in Figures 6.5 and 6.6. Figure 6.5 illustrates how linear smoothing can create new gray levels and new extrema. Figure 6.6 shows that linear scale space does not maintain the inclusion between objects. The shape inclusion principal will be discussed in Chapter 21.

We must conclude that the work on linear edge detection has been an attempt to build a theory that has not succeeded. After more than thirty years of activity, it has become clear that no robust technology can be based on these ideas. Since edge detection algorithms depend on multiple thresholds on the gradient, followed by "filling-the-holes" algorithms, there can be no scientific agreement on the identification of edge points in a given image. In short, the problems associated with linear smoothing followed by edge detection have not been resolved by the idea of chasing edges across scales.

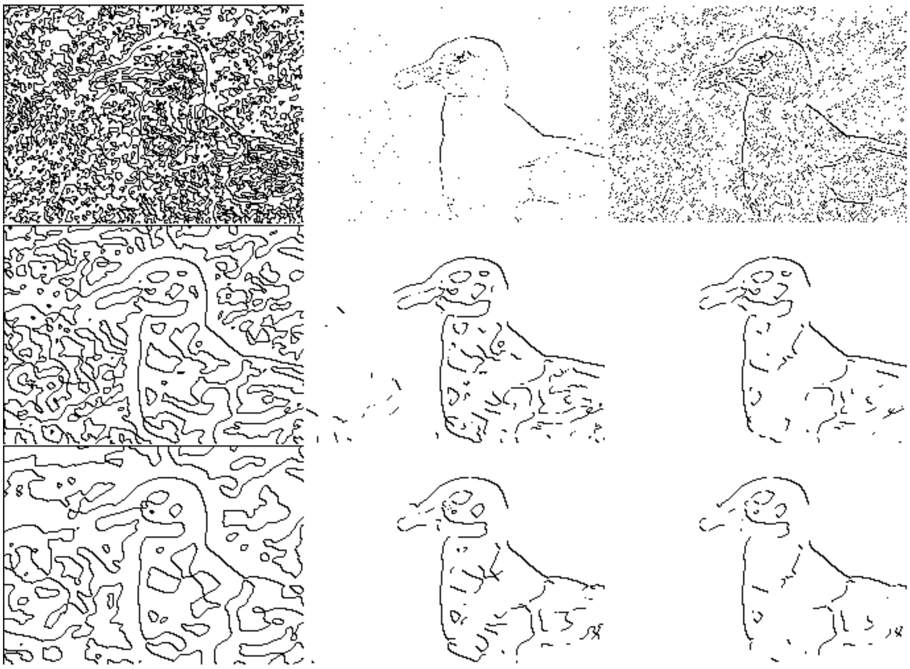

Figure 6.3: Canny's edge detector. These images illustrate the Canny edge detector. Left column: result of the Canny filter without the threshold on the gradient. Middle column: result with a visually "optimal" scale and an image-dependent threshold (from top to bottom: 15, 0.5, 0.6). Right column: result with a fixed gradient threshold equal to 0.7. Note that such an edge detection theory depends on no fewer than two parameters that must be fixed by the user: smoothing scale and gradient threshold .

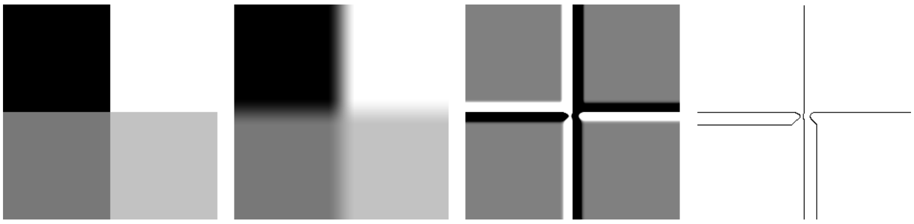

Figure 6.4: Zero-crossings of the Laplacian of a synthetic image. Left to right: the original image; the image linearly smoothed by convolution with a Gaussian; the sign of the Laplacian of the filtered image (the gray color corresponds to values close to 0, black to clear-cut negative values, white to clear-cut positive values); the zero-crossings of the Laplacian. This experiment clearly shows a drawback of the Laplacian as edge detector.

Contrast invariance

The use of contrast-invariant operators can solve some of the technical problems associated with linear smoothing and other linear image operators. An (image) operator $u \mapsto Tu$ is contrast invariant if T commutes with all nondecreasing

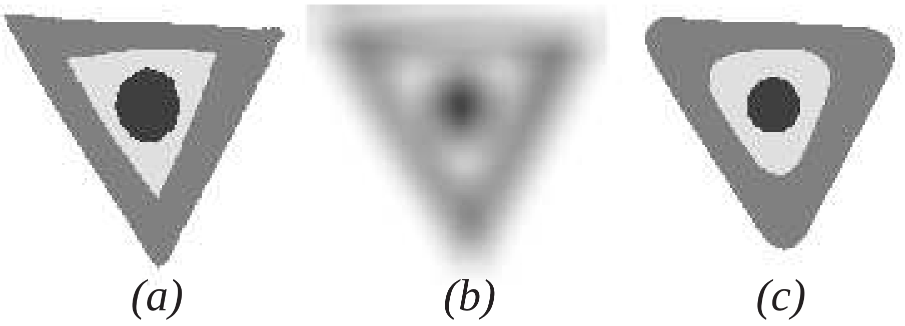

Figure 6.5: The heat equation creates structure. This experiment shows that linear scale space can create new structures and thus increase the complexity of an image. Left to right: The original synthetic image (a) contains three gray levels. The black disk is a regional and absolute minimum. The "white" ring around the black disk is a regional and absolute maximum. The outer gray ring has a gray value between the other two and is a regional minimum. The second image (b) shows what happens when (a) is smoothed with the heat equation: New local extrema have appeared. Image (c) illustrates the action on (a) of a contrast-invariant local filter, the iterated median filter, which is introduced in Chapter 16.

functions g, that is, if

$$g(Tu) = T(g(u)). \tag{6.1}$$

If image analysis is to be robust, it must be invariant under changes in lighting that produce contrast changes. It must also be invariant under the nonlinear response of the sensors used to capture an image. These, and perhaps other, contrast changes are modeled by g. If g is strictly increasing, then relation (6.1) ensures that the filtered image $Tu = g^{-1}(T(g(u)))$ does not depend on g. A problem with linear theory is that linear smoothing, that is, convolution, is not generally contrast invariant:

$$g(k * u) \neq k * (g(u)).$$

In the same way, the operator T_t that maps u_0 into the solution of the heat equation, $u(t, \cdot)$ is not generally contrast invariant. In fact, if g is C^2, then

$$\frac{\partial(g(u))}{\partial t} = g'(u)\frac{\partial u}{\partial t}$$

and

$$\Delta(g(u)) = g'(u)\Delta u + g''(u)|Du|^2.$$

Exercise 6.1. Prove this last relation. Prove that if $g(s) = as + b$ then $g(u)$ satisfies the heat equation if u does. ■

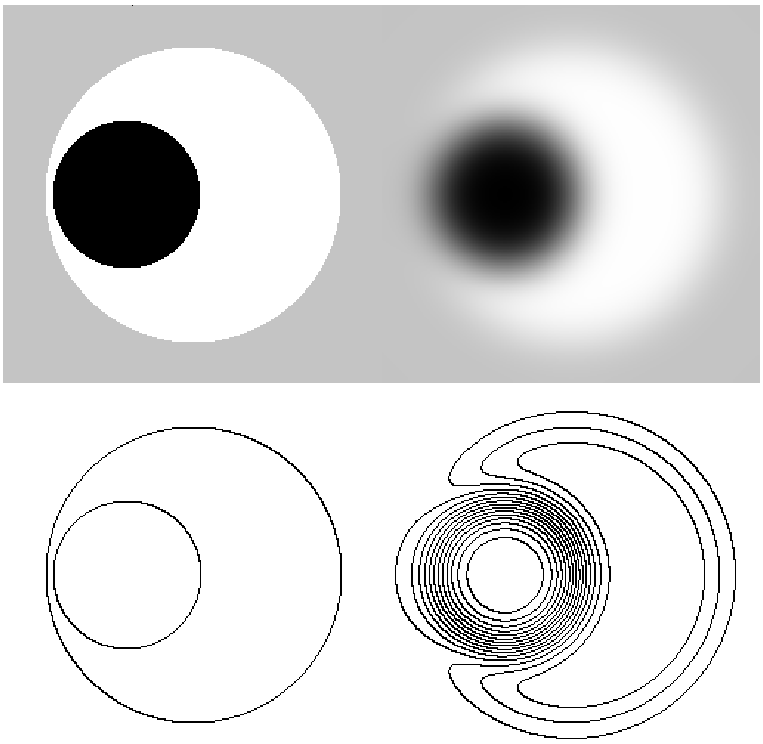

Figure 6.6: Violation of the inclusion by the linear scale space. Top, left: an image that contains a black disk enclosed by a white disk. Top, right: At a certain scale, the black and white circles mix together. Bottom, left: The boundaries of the two circles. Bottom, right: After smoothing with a certain value of t, the inclusion that existed for very small t in no longer preserved. We display the level lines of the image at levels multiples of 16.

6.2 Exercises

Exercise 6.2. Define an edge point $\mathbf{x}$ in a smooth image u as a point $\mathbf{x}$ at which $g(t)$ attains a maximum, where

$$g(t) = |Du\left(\mathbf{x} + t\frac{Du(\mathbf{x})}{|Du(\mathbf{x})|}\right)|.$$

Prove by differentiating $g(t)$ that edge points satisfy $D^2u(\mathbf{x})(Du(\mathbf{x}), Du(\mathbf{x})) = 0$ ■

Exercise 6.3. Construct simple functions u, g, and k such that $g(k * u) \neq k * (g(u))$. ■

Exercise 6.4. Consider the Perona–Malik equation in divergence form:

$$\frac{\partial u}{\partial t} = \operatorname{div}(g(|Du|)Du), \tag{6.2}$$

where $g(s) = 1/(1 + \lambda^2 s^2)$. It is easily checked that we have a diffusion equation when $\lambda|Du| \leq 1$ and an inverse diffusion equation when $\lambda|Du| > 1$. To see this, consider the second derivative of u in the direction of Du,

$$u_{\xi\xi} = D^2u\bigg(\frac{Du}{|Du|}, \frac{Du}{|Du|}\bigg),$$

and the second derivative of u in the orthogonal direction,

$$u_{\eta\eta} = D^2 u\left(\frac{Du^\perp}{|Du|}, \frac{Du^\perp}{|Du|}\right),$$

where $Du = (u_x, u_y)$ and $Du^\perp = (-u_y, u_x)$. The Laplacian can be rewritten in the intrinsic coordinates (ξ, η) as $\Delta u = u_{\xi\xi} + u_{\eta\eta}$. Prove that the Perona–Malik equation then becomes

$$\frac{\partial u}{\partial t} = \frac{1}{1 + \lambda^2 |Du|^2} u_{\eta\eta} + \frac{1 - \lambda^2 |Du|^2}{(1 + \lambda^2 |Du|^2)^2} u_{\xi\xi}.$$

Interpret the local behavior of the equation as a heat equation or a reverse heat equation according to the size of $|Du|$ compared to λ^{-1}. ■

6.3 Comments and references

Scale space. The term "scale space" was introduced by Witkin in 1983. He suggested tracking the zero-crossings of the Laplacian of the smoothed image across scales [262]. Yuille and Poggio proved that these zero-crossings can be tracked for one-dimensional signals [268]. Hummel and Moniot [113, 116] and Yuille and Poggio [269] analyzed the conjectures of Marr and Witkin according to which an image is completely recoverable from its zero-crossings at different scales. Mallat formulated Marr's conjecture as an algorithm in the context of wavelet analysis. He replaced the Gaussian with a two-dimensional cubic spline, and he used both the zero-crossings of the smoothed images and the nonzero values of the gradients at these points to reconstruct the image. This algorithm works well in practice, and the conjecture was that these zero-crossings and the values of the gradients determined the image. A counterexample given by Meyer shows that this is not the case. Perfect reconstruction is possible in the one-dimensional case for signals with compact support if the smoothing kernel is the Tukey window, $k(x) = 1 + \cos x$ for $|x| \le \pi$ and zero elsewhere. An account of the Mallat conjecture and these examples can be found in [119]. Koenderink presents a general and insightful theory of image scale space in [134].

Gaussian smoothing and edge detection. The use of Gaussian filtering in image analysis is so pervasive that it is impossible to point to a "first paper." It is, however, safe to say that David Marr's famous book, *Vision* [163], and the original paper by Hildreth and Marr [165] have had an immeasurable impact on edge detection and image processing in general. The term "edge detection" appeared as early as 1959 in connection with television transmission [121]. The idea that the computation of derivatives of an image necessitates a previous smoothing has been extensively developed by the Dutch school of image analysis [30, 84]. See also the books by Florack [83], Lindeberg [144], and Romeny [243], and the paper [78]. Haralick's edge detector [101], as implemented by Canny [38], is probably the best known image analysis operator. A year after Canny's 1986 paper, Deriche published a recursive implementation of Canny's criteria for edge detection [69].

Chapter 7

Four Algorithms to Smooth a Shape

In this short but important chapter, we discuss algorithms whose aim it is to smooth shapes. Shape must be understood as a rough data which can be extracted from an image, either a subset of the plane, or the curve surrounding it. Shape smoothing is directed at the elimination of spurious, often noisy, details. The smoothed shape can then be reduced to a compact and robust code for recognition. The choice of the right smoothing will make us busy throughout the book. A good part of the solution stems from the four algorithms we describe and their progress towards more robustness, more invariance and more locality. What we mean by such qualities will be progressively formalized. We will discuss two algorithms which directly smooth *sets*, and two which smooth Jordan curves. One of the aims of the book is actually to prove that both approaches, different though they are, eventually yield the *very same process*, namely a curvature motion.

7.1 Dynamic shape

In 1986, Koenderink and van Doorn defined a *shape* in $\mathbb{R}^N$ to be a closed subset X of $\mathbb{R}^N$ [135]. They then proposed to smooth the shape by applying the heat equation $\partial u/\partial t - \Delta u = 0$ directly to $\mathbf{1}_X$, the characteristic function of X. Of course, the solution $G_t * \mathbf{1}_X$ is not a characteristic function. The authors defined the evolved shape at scale t to be

$$X_t = \{\mathbf{x} \mid u(t, \mathbf{x}) \geq 1/2\}.$$

The value $1/2$ is chosen so the following simple requirement is satisfied: Suppose that X is the half-plane $X = \{(x, y) \mid (x, y) \in \mathbb{R}^2, x \geq 0\}$. The requirement is that this half plane doesn't move,

$$X = X_t = \{(x, y) \mid G_t * \mathbf{1}_X(x, y) \geq \lambda\},$$

and this is true only if $\lambda = 1/2$. There are at least two problems with dynamic shape evolution for image analysis. The first concerns nonlocal interactions, as illustrated in Figure 7.1. Here we have two disks that are near one another.

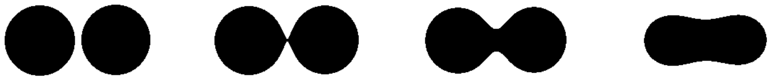

Figure 7.1: Nonlocal interactions in the dynamic shape method. Left to right: Two close disks interact as the scale increases. This creates a new, qualitatively different, shape. The change of topology, at the scale where the two disks merge into one shape, also entails the appearance of a singularity (a cusp) on the shape(s) boundaries.

The evolution of the union of both disks, considered as a single shape, is quite different from the evolution of the disks separately. A related problem, also illustrated in Figure 7.1, is the creation of singularities. Note how a singularity in orientation and the curvature of the boundary of the shape develops at the point where the two disks touch. Figure 7.2 further illustrates the problems associated with the dynamic shape method.

7.2 Curve evolution using the heat equation

We consider shapes in $\mathbb{R}^2$ whose boundaries can be represented by a finite number of simple closed rectifiable Jordan curves. Thus, each curve we consider can be represented by a continuous mapping $f : [0,1] \to \mathbb{R}^2$ such that f is one-to-one on $(0,1)$ and $f(0) = f(1)$, and each curve has a finite length. We also assume that these curves do not intersect each other. We will focus on smoothing one of these Jordan curves, which we call C_0. We assume that C_0 is parameterized by $s \in [0,L]$, where L is the length of the curve. Thus, C_0 is represented as $\mathbf{x}_0(s) = (x(s), y(s))$, where s is the length of the curve between $\mathbf{x}_0(0)$ and $\mathbf{x}_0(s)$.

At first glance, it might seem reasonable to smooth C_0 by smoothing the coordinate functions x and y separately. If this is done linearly, we have seen from Theorem 2.3 that the process is asymptotic to smoothing with the heat equation. Thus, one is led naturally to consider the vector heat equation

$$\frac{\partial \mathbf{x}}{\partial t}(t,s) = \frac{\partial^2 \mathbf{x}}{\partial s^2}(t,s) \tag{7.1}$$

with initial condition $\mathbf{x}(0,s) = \mathbf{x}_0(s)$. If $\mathbf{x}(t,s) = (x(t,s), y(t,s))$ is the solution of (7.1), then we know from Proposition 1.9 that

$$\inf_{s\in[0,L]} x_0(s) \le x(t,s) \le \sup_{s\in[0,L]} x_0(s),$$
$$\inf_{s\in[0,L]} y_0(s) \le y(t,s) \le \sup_{s\in[0,L]} y_0(s),$$

for $s \in [0,L]$ and $t \in [0,+\infty)$. Thus, the evolved curves C_t remain in the rectangle that held C_0. Also, we know from Proposition 2.5 that the coordinate functions $x(t,\cdot)$ and $y(t,\cdot)$ are C^∞ for $t > 0$. There are, however, at least two reasons that argue against smoothing curves this way:

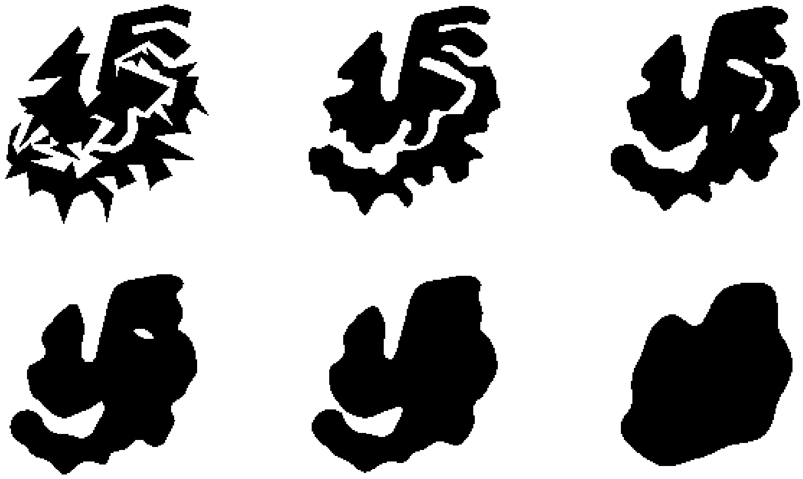

Figure 7.2: Nonlocal behavior of shapes with the dynamic shape method. This image displays the smoothing of two irregular shapes by the dynamic shape method (Koenderink–van Doorn). Top left: initial image, made of two irregular shapes. From left to right, top to bottom: dynamic shape smoothing with increasing Gaussian variance. Notice how the shapes merge more and more. We do not have a separate analysis of each shape but rather a "joint analysis" of the two shapes. The way the shapes merge is of course sensitive to the initial distance between the shapes. Compare with Figure 7.4.

(1) When $t > 0$, s is no longer a length parameter for the evolved curve C_t.

(2) Although $x(t, \cdot)$ and $y(t, \cdot)$ are C^∞ for $t > 0$, this does not imply that the curves C_t have similar smoothness properties. In fact, it can be seen from Figure 7.3 that it is possible for an evolved curve to cross itself and it is possible for it to develop singularities.

How is this last mentioned phenomenon possible ? It turns out that one can parameterize a curve with corners or cusps with a very smooth parameterization: see Exercise 7.1.

In image processing, we say that a process that introduces new features, such as described in item (2) above, is not *causal.* [1]

7.3 Restoring locality and causality

Our main objective is to redefine the smoothing processes so they are local and do not create new singularities. This can be done by alternating a small-scale linear convolution with a natural renormalization process.

[1] This informal definition should not be confused with the use of "causality," as it is used, for example, when speaking about filters: A filter F is said to be causal, or realizable, if the equality of two signals s_0 and s_1 up to time t_0 implies that $Fs_0(t) = Fs_1(t)$ for the same period.

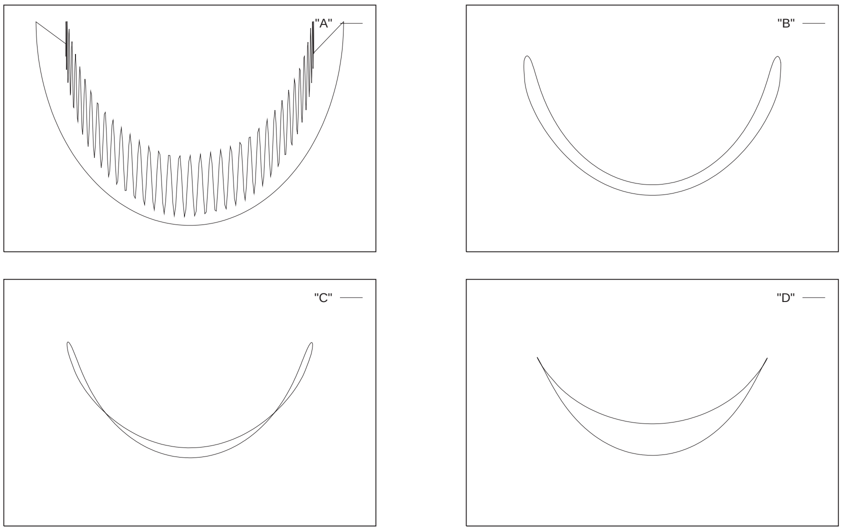

Figure 7.3: Curve evolution by the heat equation. The coordinates of the curves are parameterized by the arc length and then smoothed as real functions of the length using the heat equation. From A to D: the coordinates are smoothed with an increasing scale. Each coordinate function therefore is C^∞; the evolving curve can, however, develop self-crossings (as in C) or singularities (as in D).

7.3.1 Localizing the dynamic shape method

In the case of dynamic shape analysis, we define an alternate dynamic shape algorithm as follows:

Algorithm 7.1 (The Merriman–Bence–Osher algorithm).

(1) Convolve the characteristic function of the initial shape X_0 with G_h, where h is small.

(2) Define $X_1 = \{\mathbf{x} \mid G_h * \mathbf{1}_{X_0} \geq 1/2\}$.

(3) Set $X_0 = X_1$ and go back to (1).

This is an iterated dynamic shape algorithm. The dynamic shape method itself is an example of a *median filter*, which will be defined in Chapter 16. The Merriman–Bence–Osher algorithm is thus an *iterated median filter* (see Figure 7.4). We will see in Chapters **??** and **??** that median filters have asymptotic properties that are similar to those expressed in Theorem 3.3. In the case of median filters, the associated partial differential equation will be a curvature motion equation (defined in Chapter 18).

7.3.2 Renormalized heat equation for curves

In 1992, Mackworth and Mokhtarian noticed the loss of causality when the heat equation was applied to curves [154]. Their method to restore causality looks, at least formally, like the remedy given for the nonlocalization of the dynamic

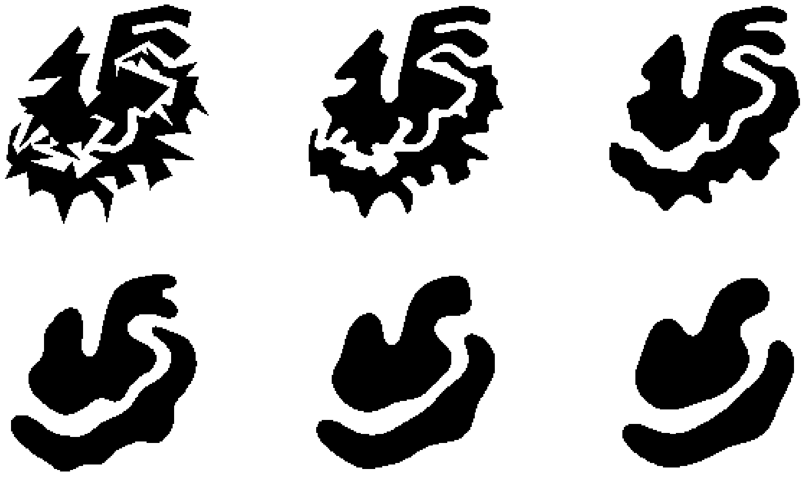

Figure 7.4: The Merriman–Bence–Osher shape smoothing method is a localized and iterated version of the dynamic shape method. A convolution of the binary image with small-sized Gaussians is alternated with mid-level thresholding. It uses the same initial data (top, left) as in Figure 7.2. From left to right, top to bottom: smoothing with increasing scales. Notice that the shapes remain separate. In fact, their is no interaction between the evolving shapes. Each one evolves as if the other did not exist.

shape method. Instead of applying the heat equation for relatively long times (or, equivalently, convolving the curve $\mathbf{x}$ with the Gaussian G_t for large t), they use the following algorithm:

Algorithm 7.2 (Renormalized heat equation for curves).

(1) Convolve the initial curve $\mathbf{x}_0$, parameterized by its length parameter $s_0 \in [0, L_0]$, with the Gaussian G_h, where h is small.

(2) Let L_n denote the length of the curve $\mathbf{x}_n$ obtained after n iterations and let s_n denote its length parameter. For $n \geq 1$, write $\tilde{\mathbf{x}}_{n+1}(s_n) = G_h * \mathbf{x}_n(s_n)$. Then reparameterize $\tilde{\mathbf{x}}_{n+1}$ by its length parameter $s_{n+1} \in [0, L_{n+1}]$, and denote it by $\mathbf{x}_{n+1}$.

(3) Iterate.

This algorithm is illustrated in Figure 7.5. It should be compared with Figure 7.3.

Theorem 7.1. *Let $\mathbf{x}$ be a C^2 curve parameterized by its length parameter $s \in [0, L]$. Then for small h,*

$$G_h * \mathbf{x}(s) - \mathbf{x}(s) = h \frac{\partial^2 \mathbf{x}}{\partial s^2} + o(h). \tag{7.2}$$

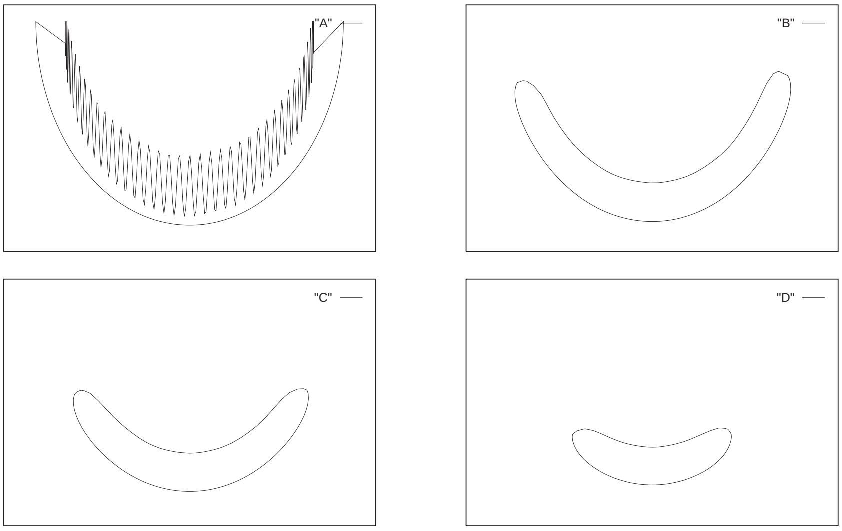

Figure 7.5: Curve evolution by the renormalized heat equation (Mackworth–Mokhtarian). After each smoothing step, the coordinates of the curve are reparameterized by the arc length of the smoothed curve. From A to D: the curve is smoothed with an increasing scale. Note that, in contrast with the linear heat equation (Figure 7.3), the evolving curve shows no singularities and does not cross itself.

This theorem is easily checked, see Exercise 7.2

In view of (7.2) and what we have seen regarding asymptotic limits in Theorem 3.3 and Exercise 3.5, it is reasonable to conjecture that, in the asymptotic limit, Algorithm 7.2 will yield the solution of following evolution equation:

$$\frac{\partial \mathbf{x}}{\partial t} = \frac{\partial^2 \mathbf{x}}{\partial s^2}, \tag{7.3}$$

where $\mathbf{x}_0 = \mathbf{x}(0, \cdot)$. It is important to note that (7.3) is *not* the heat equation (7.1). Indeed, from Algorithm 7.2 we see that s must denote the length parameter of the evolved curve $\mathbf{x}(t, \cdot)$ at time t. In fact $\partial^2\mathbf{x}/\partial s^2$ has a geometric interpretation as a curvature vector. We will study this nonlinear curve evolution equation in Chapter 18.

7.4 Exercises

Exercise 7.1. Construct a C^∞ mapping $f : [0,1] \to \mathbb{R}^2$ such that the image of $[0,1]$ is a square. This shows that a curve can have a C^∞ parameterization without being smooth. ■

Exercise 7.2. Prove Theorem 7.1. If $\mathbf{x}$ is a C^3 function of s, then the result follows directly from Theorem 3.2. The result holds, however, for a C^2 curve. ■

7.5 Comments and references

Dynamic shape, curve evolution, and restoring causality. Our account of the dynamic shape method is based on the well-known paper by Koenderink and van Doorn in which they introduced this notion [135]. The curve evolution by the heat equation is from the first 1986 version of curve analysis proposed by Mackworth and Mokhtarian [153]. See also the paper by Horn and Weldon [109]. There were model errors in the 1986 paper [153] that were corrected by the authors in their 1992 paper [154]. There, they also proposed the correct intrinsic equation. However, this 1992 paper contains several inexact statements about the properties of the intrinsic equation. The correct theorems and proofs can be found in a paper by Grayson written in 1987 [98]. The algorithm that restores causality and locality to the dynamic shape method was discovered by Merriman, Bence, and Osher, who devised this algorithm for a totally different reason: They were looking for a clever numerical implementation of the mean curvature equation [170].

Topological change under smoothing. We have included several figures that illustrate how essential topological properties of an image change when the image is smoothed with the Gaussian. Damon has made a complete analysis of the topological behavior of critical points of an image under Gaussian smoothing [63]. This analysis had been sketched in [267].

Chapter 8

Affine Invariant Image Comparison

If a physical object has a smooth or piecewise smooth boundary, its images obtained by cameras in varying positions undergo smooth apparent deformations. These deformations are locally well approximated by affine transforms of the image plane.

In consequence the solid object recognition problem has often been led back to the computation of affine invariant image local features. Such invariant features could be obtained by normalization methods, but no fully affine normalization method exists for the time being. As a matter of fact, the scale invariance, which actually means invariance to blur, is only dealt with by methods inspired from the scale space theory, like the SIFT method. By simulating zooms out, this method normalizes the four translation, rotation and scale (blur) parameters, out of the six parameters of an affine transform. Affine normalization methods like MSER or Hessian Affine normalize with respect to all six parameters of the affine transform, but this normalization is imperfect, not dealing rigorously with blur for MSER, or not starting with affine invariant scale space extrema for Hessian Affine.

The method proposed in this chapter, affine SIFT (A-SIFT), simulates all image views obtainable by varying the two camera parameters left over by the SIFT method. Then it normalizes the other four parameters by simply using the SIFT method itself. The two additional parameters are the angles (a longitude and a latitude) defining the camera axis orientation. Mathematical arguments will be given in Chapter 9 to prove that the resulting method is fully affine invariant, up to an arbitrary precision.

Against any prognosis, simulating all views depending on the two camera orientation parameters is feasible with no dramatic computational load. The method permits to reliably identify features that have undergone tilts of large magnitude, up to 30 and more, while state-of-the-art methods do not exceed tilts of 2.5 (SIFT) or 4.5 (MSER). This chapter puts in evidence the role of high *transition tilts*: while a tilt from a frontal to an oblique view exceeding 6 is rare, higher transition tilts are common as soon as two oblique views of an object are compared (see Fig. 8.1). Thus, a fully affine invariance is required for 3D scene analysis. This fact is substantiated by many experiments.

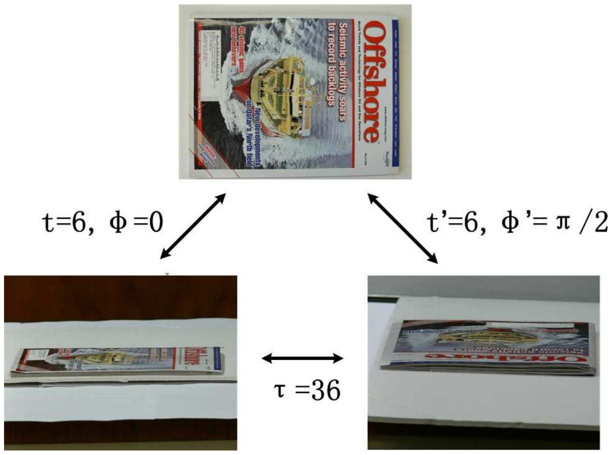

Figure 8.1: High transition tilts

Section 8.1 gives the main decomposition formula of affine maps used throughout the paper and its geometric interpretation in terms of cameras at infinity. Section 10.1 describes and discusses a method that attempts affine invariance by normalization: MSER. Section 8.2 describes the A-SIFT algorithm and discusses precursors. Section 8.4 addresses the critical sampling issues for the new-simulated parameters in A-SIFT (tilt and longitude). It then provides a complexity analysis and a fast version of the method. Section 8.3 presents and experiments the crucial notion of *transition tilt.*

8.1 The affine camera model

The general (solid) shape recognition problem starts with several photographs of a physical object, possibly taken with different cameras and view points. These digital images are the *query* images. Given other digital images, the *search* images, the question is whether some of them contain, or not, a view of the object taken in the query image. A solid object's view can deform from an image to another for two obvious reasons: First, because it underwent some physical deformation, and second, because the change of camera position induced an apparent deformation.

Image distortions arising from viewpoint changes can be locally modeled by affine planar transforms, provided the object's boundaries are piecewise smooth. In other terms, a perspective effect can be modeled by a combination of several different affine transforms in different image regions (see Fig. 8.3). Indeed, by first order Taylor formula, any planar smooth deformation $(x, y) \to (X, Y) = (F_1(x, y), F_2(x, y))$ can be locally approximated around each point $(x_0, y_0) \to (X_0, Y_0)$ by the affine map

$$\begin{pmatrix} X - X_0 \\ Y - Y_0 \end{pmatrix} = \begin{bmatrix} \frac{\partial F_1}{\partial x}(x_0, y_0) & \frac{\partial F_1}{\partial y}(x_0, y_0) \\ \frac{\partial F_2}{\partial x}(x_0, y_0) & \frac{\partial F_2}{\partial y}(x_0, y_0) \end{bmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} + O\begin{pmatrix} (x - x_0)^2 + (y - y_0)^2 \\ (x - x_0)^2 + (y - y_0)^2 \end{pmatrix}. \tag{8.1}$$

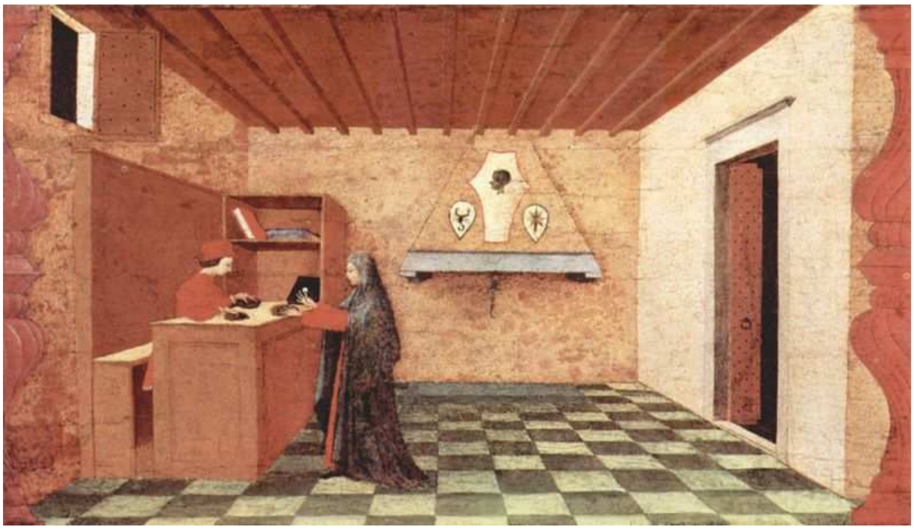

Figure 8.2: Geometric interpretation of the Taylor formula (8.1): Although the global deformation of each wall is strongly projective (a rectangle becomes a trapezoid), the it local deformation is affine: each tile on the pavement is almost a parallelogram. Indeed, projective maps are C^1 and therefore locally affine. The painting, due to Uccello, is one of the first Renaissance paintings with a correct geometric perspectives following the rules invented by Brunelleschi.

Figure 8.3: Another way to understand why the local object apparent deformations are affine. Local planar homographies are equivalent to multiple local cameras at infinity. Cameras at infinity generate affine deformations of planar objects. This is true even if the object under observation is curved, because it is then locally planar. Thus, the overall apparent deformation of the object is C^1, and Formula (8.1) applies.

Thus, all object deformations and all camera motions are locally approximated by affine transforms. For example, in the case of a flat object, the deformation induced by a camera motion is a planar homographic transform, which is smooth and therefore locally tangent to affine transforms.

The converse statement is true: *any affine transform with positive determinant can be interpreted as the apparent deformation induced on a planar object by a camera motion, the camera being assumed far away from the object.* Thus, under the local smoothness assumption of the object's boundary, the (local) deformation model of an image $u(x,y)$ under a deformation of the object or under a camera motion is

$$u(x,y) \to u(ax+by+e, cx+dy+f),$$

where the mapping

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

is any affine transform of the plane with positive determinant. The above statements rely on the next crucial following decomposition formula.

Theorem 8.1. *Any linear planar map whose matrix A has strictly positive determinant, and which is not a similarity, has a unique decomposition*

$$A = H_\lambda R_1(\psi) T_t R_2(\phi) = \lambda \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \qquad (8.2)$$

where $\lambda > 0$, λt is the determinant of A, R_i are rotations, $\phi \in [0,\pi[$, and T_t is a tilt, namely a diagonal matrix with a first eigenvalue equal to $t \geq 1$ and the second one equal to 1.

Proof. Consider the real symmetric positive semi-definite matrix A^tA, where A^t denotes the transposed matrix of A. By classic spectral theory there is an orthogonal transform O such that $A^tA = ODO^t$ where D a diagonal matrix with ordered eigenvalues $\lambda_1 \geq \lambda_2$. Set $O_1 = AOD^{-\frac{1}{2}}$. Then

$$O_1O_1^t = AOD^{-\frac{1}{2}}D^{-\frac{1}{2}}O^tA^t = AOD^{-1}O^tA^t = A(A^tA)^{-1}A^t = I.$$

Thus, there are orthogonal matrices O_1 and O such that

$$A = O_1 D^{\frac{1}{2}} O^t. \qquad (8.3)$$

Since the determinant of A is positive, the product of the determinants of O and O_1 is positive. If both determinants are positive, then O and O_1 are rotations and we can write $A = R(\psi)DR(\phi)$. If ϕ is not in $[0,\pi[$, changing ϕ into $\phi - \pi$ and ψ into $\psi + \pi$ ensures that $\phi \in [0,\pi[$. If the determinants of O and O_1 are both negative, replacing O and O_1 respectively by $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} O$ and $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} O_1$ makes them into rotations without altering (8.3), and we can as above ensure $\phi \in [0,\pi[$ by adapting ϕ and ψ. The final decomposition is obtained by taking for λ the smaller eigenvalue of $D^{\frac{1}{2}}$. □

Exercise 8.1. The aim of the exercise is to show the uniqueness of the decomposition (8.2). Assume there are two decompositions $\lambda R_1 \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} R_2 = \lambda' R_1' \begin{bmatrix} t' & 0 \\ 0 & 1 \end{bmatrix} R_2'$. Using the uniqueness of the eigenvalues of a matrix show first that $\lambda = \lambda'$, $t = t'$. You will obtain a relation of the form $R_1 D R_2 = D$ where D is diagonal and R_1 and R_2 are rotations. Deduce from this relation that $R_1 D^2 R_1^t = D^2$. Deduce from this last relation the form of R_1, conclude carefully. ■

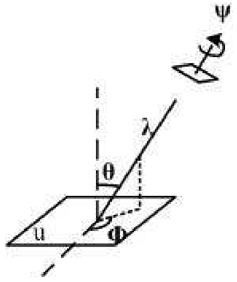

Figure 8.4: Geometric interpretation of the decomposition formula (8.2).

Exercise 8.2. Consider two cameras looking at a flat square piece of landscape which is assimilated to an infinite resolution image $u_0(x, y)$ (See Fig. 8.4). The first camera is very far above the landscape and looking down perpendicularly to the landscape.

(i) Assuming the first camera is pin-hole, show that the generated image is a square image $u_0(\mu R(\psi)(x, y))$. Consider the coordinate system $(O, \vec{\imath}, \vec{\jmath}, \vec{k})$ such that $(\vec{\imath}, \vec{\jmath})$ are the coordinate vectors in the square image u_0, parallel to the image sides, and O is the image center.

(ii) Assume a second pinhole camera has its optical axis pointing down to O. Assume its optical axis is supported by the unit vector with coordinates $(\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$. Assume again that this camera is very far from the square piece of landscape, so the light rays coming from the landscape to the camera are almost parallel. Thus the image formation on this second camera is assimilated to an orthogonal projection of the landscape u_0 onto a plane passing by the camera center C and orthogonal to the optical axis. Taking adequate coordinates on this coordinate plane, show that the generated image is $u_0 \left(R(\psi_1) T_{t_1} R(\phi_1) \begin{pmatrix} x \\ y \end{pmatrix} \right)$ for some values of ψ_1, ϕ_1, t, that you will relate to ϕ, ψ, and θ.

■

Fig. 8.4 shows a camera motion interpretation of this affine decomposition: ϕ and $\theta = \arccos 1/t$ are the viewpoint angles and ψ parameterizes the camera spin. Thus, this figure illustrates the four main parameters in the affine image deformation caused by a camera motion, starting from a frontal view u. The camera is assumed to stay far away from the image. The camera can first move parallel to the object's plane: this motion induces a translation $\mathcal{T}$ that is not represented here. The camera can rotate around its optical axis (rotation parameter ψ). Its optical axis can take a θ angle with respect to the normal to the image plane u. This parameter is called *latitude*. The plane containing the normal and the new position of the optical axis makes an angle ϕ with a fixed vertical plane. This angle is called *longitude.* Last but not least, the camera can move forward or backward. This is the zoom parameter λ. The motion of a frontal view $\lambda = 1$, $t = 1$, $\phi = \psi = 0$ to a slanted view corresponds to the image deformation $u(x, y) \to u(A(x, y))$ given by (8.2).

8.2 A-SIFT : combining simulation and normalization

The idea of combining simulation and normalization is the main successful ingredient of the SIFT method. This method normalizes rotations and translations, but simulates all zooms out of the query and of the search images. Because of the feature, it is the only fully scale invariant method.

A-SIFT simulates with enough accuracy *all* distortions caused by a variation of the direction of the optical axis of a camera (two parameters). Then it normalizes the other four by the SIFT method, or any other method that is rotation, translation, and scale invariant. More specifically, the method proceeds by the following steps. (See Fig. 8.5.)

A-SIFT algorithm

1. Each image is transformed by simulating all possible affine distortions caused by the change of orientation of the camera axis of camera from a frontal position. These distortions depend upon two parameters: the longitude ϕ and the latitude θ. The images undergo ϕ-rotations followed by tilts with parameter $t = |\frac{1}{\cos\theta}|$ (a tilt by t in the direction of x is the operation $u(x, y) \to u(tx, y)$). For digital images, the tilt is performed as t-subsampling, and therefore requires the previous application of an antialiasing filter in the direction of x, namely the convolution by a gaussian with standard deviation $c\sqrt{t^2 - 1}$. For good antialiasing, $c \simeq 0.8$, see Chapter 4.2.

2. These rotations and tilts are performed for a finite and small number of latitudes and longitudes, the sampling steps of these parameters ensuring that the simulated images keep close to any other possible view generated by other values of ϕ and θ.

3. All simulated images are compared by SIFT.

4. To be more specific, the latitudes θ are such that the associated tilts follow a geometric series 1, a, a^2, , $\ldots$, a^n, with $a > 1$. The choice $a = \sqrt{2}$ is a good compromise between accuracy and sparsity. The value n can go up to 6 or more, if the tilts are simulated on the query and the searched image, and up to 10 and more if the tilts are simulated on one image only. That way, transition tilts going up to 64 and more can be explored.

5. The longitudes ϕ are for each tilt an arithmetic series 0, b/t, $\ldots$, kb/t, where $b \simeq 72°$ seems again a good compromise, and k is the last integer such that $kb/t < 180°$.

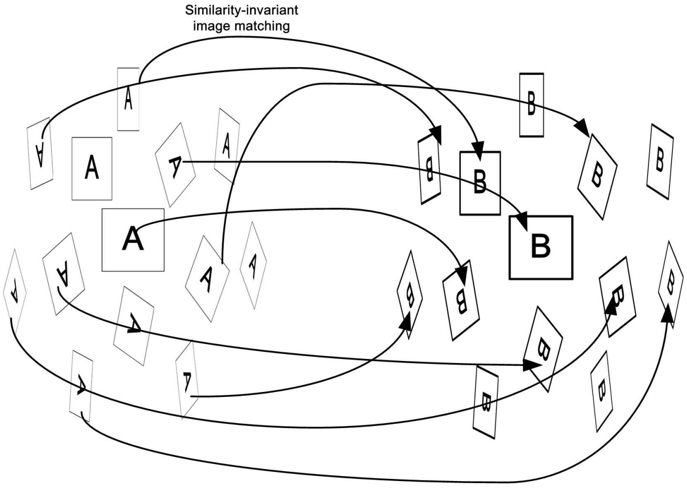

Figure 8.5: Overview of the A-SIFT algorithm. The square images A and B represent the compared images **u** and **v**. A-SIFT simulates arbitrary camera changes of direction by applying rotations followed by a tilts to both images. The simulated images, represented by the parallelograms, are then compared with an image matching algorithm like SIFT, that is invariant to similarity transformations, i.e., invariant to scale change, rotation and translation.

8.3 High transition tilts

Equation (8.2) and its geometric interpretation in Fig. 8.4 are crucial to the scopes of this study. This last figure associates any linear map A with positive singular eigenvalues with the planar deformation $u(A(x, y))$ of a frontal view $u(x, y)$, when the camera changes position. The parameter λ corresponds to a change of scale. The non critical translation parameter has been eliminated by assuming that the camera axis meets the image plane at a fixed point. Let us now consider the case where *two* camera positions, not necessarily frontal are at stake, corresponding to two different linear maps A and B. (Again, the translation parameter is left out of the discussion by fixing the intersection of the camera axis with image plane.) This physical situation is the generic one; when taking several snapshots of a scene, there is no particular reason why objects would be taken frontally. The resulting images are $u_1(x, y) = u(A(x, y))$ and $u_2(x, y) = u(B(x, y))$. Let us now take one of these images as *reference image*, and the other one as search image.

Definition 8.2. *Given two views of a planar image, $u_1(x, y) = u(A(x, y))$ and $u_2(x, y) = u(B(x, y))$, we call transition tilt $\tau(u_1, u_2)$ and transition rotation $\phi(u_1, u_2)$ the unique parameters such that*

$$BA^{-1} = H_\lambda R_1(\psi) T_\tau R_2(\phi), \tag{8.4}$$

with the notation of Formula (8.2).

It is an easy check that the transition tilt is symmetric, namely $\tau(u_1, u_2) = \tau(u_2, u_1)$. Fig. 8.6 illustrates the affine transition between two images taken

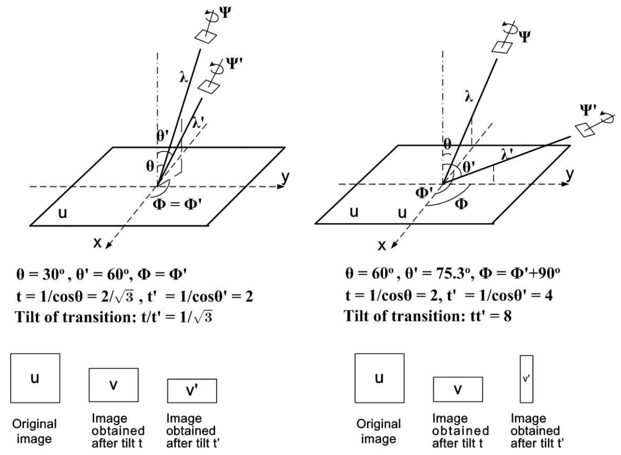
Ψ
Ψ'
λ
λ'
θ'
θ
y
u
Φ = Φ'
x
θ = 30°, θ' = 60°, Φ = Φ'
t = 1/cosθ = 2/√3 , t' = 1/cosθ' = 2
Tilt of transition: t/t' = 1/√3
u
v
v'
Original image
Image obtained after tilt t
Image obtained after tilt t'
Ψ
λ
Ψ'
λ'
θ
θ'
y
Φ'
Φ
u
u
x
θ = 60°, θ' = 75.3°, Φ = Φ'+90°
t = 1/cosθ = 2, t' = 1/cosθ' = 4
Tilt of transition: tt' = 8
u
v
v'
Original image
Image obtained after tilt t
Image obtained after tilt t'

Figure 8.6: Illustration of the difference between absolute tilt and transition tilt.

from different viewpoints, and in particular the difference between absolute tilt and transition tilt. The camera is first put in two positions corresponding to absolute tilts t and t', but with $\phi = \phi'$. The transition tilt between the resulting images v and v' is $\tau = t'/t$, assuming $t' = \max(t', t)$. On the second illustration of Fig. 8.6, the tilts are made in two orthogonal directions: $\phi = \phi' + \pi/2$. Then an easy calculation shows that the transition tilt between v and v' is the product $\tau(v, v') = tt'$. Thus, *two moderate absolute tilts can lead to a large transition tilt!* In the first case considered in the figure, the transition tilt is $\sqrt{3}$ and therefore smaller than the absolute tilts. In the second case, the tilt is $tt' = 8$. Since in realistic cases the tilt can go up to 6 or even 8, is easily understood that the *transition tilt* can go up to 36, 84, and more.

Exercise 8.3. The aim of the exercise is to prove that, given two views $u = Au_0$ and $v = Bu_0$ of a same image, with absolute tilts s and t satisfying $t \geq s$, the transition tilt $\tau = \tau(BA^{-1})$ between u and v satisfies $\frac{t}{s} \leq \tau \leq st$.

(i) Set $A = H_\lambda Q_1 T_s Q_2$ and $B = H_\mu R_1 T_s R_2$, where Q_1, Q_2, R_1, R_2 are rotations, H_λ and H_μ homotheties and T_s and T_t tilts with $t \geq s$. Show first that $\tau(BA^{-1}) = \tau(T_t R_2 Q_2^{-1} T_s^{-1})$.

(ii) Deduce that if $Q_2 = R_2$, then $\tau(BA^{-1}) = \frac{t}{s}$.

(iii) Deduce also that if $R_2 Q_2^{-1} = R(\frac{\pi}{2})$ then $\tau(BA^{-1}) = st$.

(iv) Set $R(\phi) = R_2 Q_2^{-1}$. Thus $\tau(BA^{-1}) = \tau(C)$, with $C =: T_t R(\phi) T_{s^{-1}}$ Compute the matrix $C^t C$. Check that its determinant is $det(C^t C) = t^2 s^2$ and that its trace is $tr(C^t C) = (s^2 - 1)(t^2 - 1)\cos^2\varphi + s^2 + t^2$.

(v) Show that the eigenvalues $\lambda_1 \geq \lambda_2$ of $C^t C$ satisfy $\frac{t^2}{s^2} \leq \frac{\lambda_1}{\lambda_2} \leq t^2 s^2$.

(vi) Conclude.

■

Each recognition method can be characterized by its transition tilt, namely the variation in relative observation angle compatible with recognition. Fig. 8.13 shows the regions of the observation half sphere that can be attained for a given transition tilt, from a fixed viewpoint with various tilts. This figure shows perspective and zenith views of the observation half sphere. The central point of the bright region on the observation sphere is the original viewpoint from which an image has been taken. The rest of the bright region depicts the attainable observation region, namely the subset of the observation half sphere for which recognition succeeds for the given transition tilt. The latitude angle of the first image is respectively $\theta = 45,\ 60,\ 70,\ 80°$ that correspond respectively to the absolute tilts $t = \sqrt{2},\ 2,\ 2.9,\ 5.8$. The three columns show the attainable regions on the half sphere for transition tilts $t < 2.5,\ 5,\ ,$ and 40. From the strong 80°, latitude, it needs a $\tau = 40$ transition tilt to attain the rest of the sphere! SIFT and MSER only attain small regions.

Fig. 8.7 shows the A-SIFT results for a pair of images under orthogonal viewpoints (transition rotation $\phi = 90°$) that leads to an extreme transition tilt $t \approx 37$. This is not at all an exceptional situation. It just so happens that the object's planar surface is observed at the same latitude by both views with a tilt $t \simeq t' \simeq 6$. This figure shows two snapshots of a magazine lying on a table, not even really flat, and with a non lambertian surface plagued with reflections. The difference of longitudes being about 90 degrees, the transition tilt between both images is surprisingly high: $\tau = tt' \simeq 37$. Thus, it is many times larger than the transition tilt attainable with SIFT or MSER. A-SIFT finds 120 matches out of which only 4 are wrong.

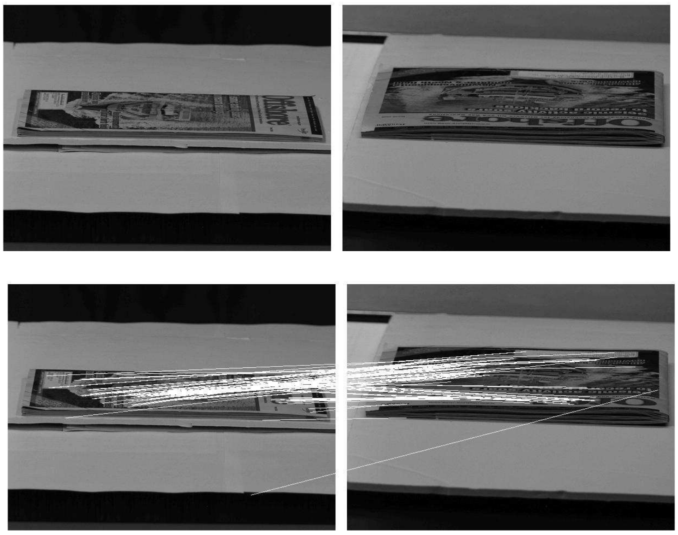

Figure 8.7: Top: Image pair with transition tilt $t \approx 37$. (SIFT, Harris-Affine, Hessian-Affine and MSER fail completely.) Bottom: A-SIFT finds 120 matches out which 4 are false. See comments in text.

The relevance of the notion of transition tilt is corroborated by the fact that the highest transition tilt τ_{max} permitting to match two images with absolute tilts t and t' is fairly independent from t and t'. It has been experimentally checked that for SIFT $\tau_{max} \simeq 2.5$ and for MSER $\tau_{max} \simeq 4$.

To demonstrate this for SIFT, the transition tilts attainable by SIFT have been explored by systematic tilt simulations and tests. The experiments have been performed in the most favorable conditions for SIFT. The seed image $\mathbf{u}_0$ is a high quality frontal view of the Graffiti series. Tilted views from this frontal view were simulated by subsampling the image in one direction by a factor $\sqrt{t}$, and oversampling the image in the orthogonal direction by the same factor. That way, the absolute tilt is t, but the image area is not decreased. A set of tilted-rotated images $\mathbf{u}_1 = \mathbf{u}_0(t_1, 0)$ and $\mathbf{u}_2 = \mathbf{u}(t_2, \phi)$ was generated by this method from $\mathbf{u}$ with absolute tilts $t_1 = (\sqrt{2})^k$, $k = 1, 2, \ldots, 5$, $t_2 = (2^{\frac{1}{4}})^l$, $l = 1, 2, \ldots, 14$, and ϕ_2 in a dense subset of $[0,\ 90°]$. The table shows for each pair t_1, t_2 the maximal longitude ϕ_{max} ensuring that $\mathbf{u}_1(t_1, 0°)$ and $\mathbf{u}_2(t_2, \phi_{max})$ match. On the right of ϕ_{max}, the table displays in each box the corresponding transition tilt $\tau(t_1, 0, t_2, \varphi_{max})$. Conspicuously enough, τ_{max} is most of the time close to 2.5. This experiment, and other similar ones, substantiate the empirical law that *SIFT works for comparing images with transition tilts smaller than 2.5.* In all of these tests, success with SIFT means that at least 20 correct SIFT descriptors, or SIFs, have been found.

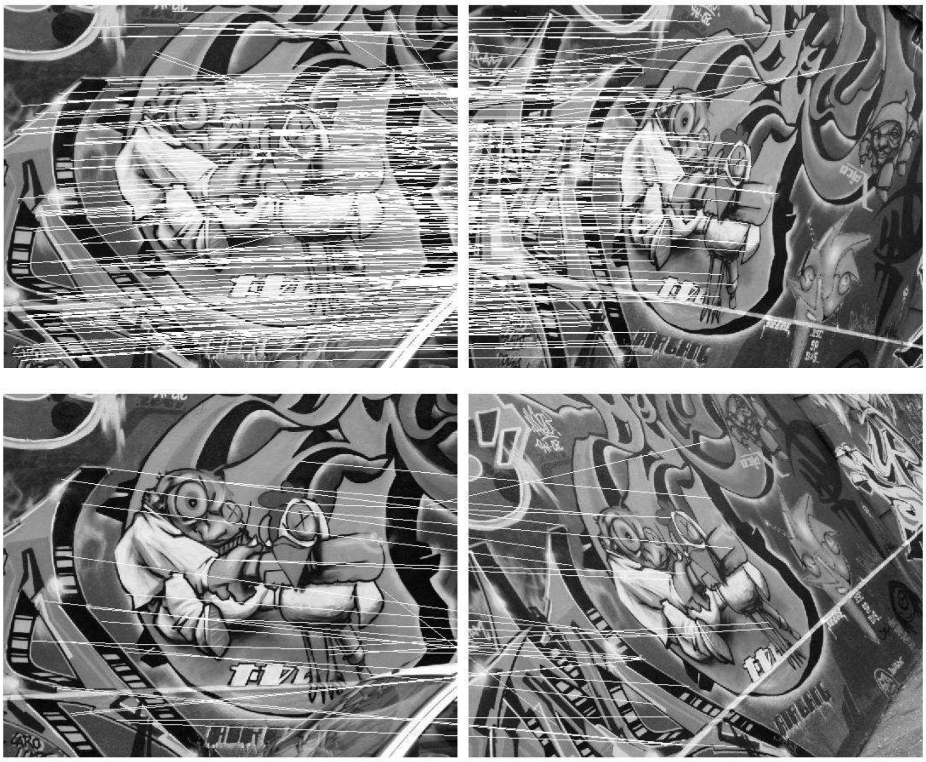

Figure 8.8: Top and bottom: SIFT detects respectively 234 and 28 matches between a frontal image and two images with tilts $t \approx 2$ and $t \approx 2.3$. This latter value is close to the limiting tilt for SIFT to work.

	$t_1 = \sqrt{2}$	$t_1 = 2$	$t_1 = 2\sqrt{2}$	$t_1 = 4$	$t_1 = 4\sqrt{2}$
$t_2 = 2^{1/4}$	90°/1.7	60°/2.2	0°/2.4		
$t_2 = 2^{1/2}$	90°/2.0	56°/2.4	11°/2.1		
$t_2 = 2^{3/4}$	90°/2.4	50°/2.6	20°/2.1	0°/2.4	
$t_2 = 2$	63°/2.6	36°/2.4	20°/2.1	9°/2.2	
$t_2 = 2 \times 2^{1/4}$	37°/2.4	30°/2.3	23°/2.3	9°/1.9	
$t_2 = 2 \times 2^{1/2}$	18°/2.6	22°/2.2	24°/2.6	12°/2.0	0°/1.4
$t_2 = 2 \times 2^{3/4}$	6°/2.4	16°/2.2	21°2.6	16°/2.5	5°/1.4
$t_2 = 4$	0°/2.8	9°/2.2	18°/2.6	14°/2.4	9°/1.8
$t_2 = 4 \times 2^{1/4}$		4°/2.4	11°/2.2	12°/2.3	8°/2.0
$t_2 = 4 \times 2^{1/2}$			6°/2.2	7°/1.9	8°/2.3
$t_2 = 4 \times 2^{3/4}$			0°/2.4	5°/2.0	8°/2.5
$t_2 = 8$				0°/2.0	7°/2.5
$t_2 = 8 \times 2^{1/4}$					4°/2.6
$t_2 = 8 \times 2^{1/2}$					3°/2.9

Table 8.1: m/n in each entry means: maximal longitude angle ϕ giving at least 20 matches by SIFT / corresponding transition tilt $\tau(t_1, t_2, \phi)$. This table shows that SIFT covers a transition tilt $\tau \approx 2.5$.

8.4 Parameter sampling and complexity

8.4.1 Sampling ranges

The camera motion depicted in Fig. 8.4 shows that ϕ should naturally cover all the directions from 0 to 2π. But, by Theorem 8.1, it is enough to simulate ϕ from 0 to π to cover all possible linear transforms.

The sampling range of the tilt parameter t determines the degree of the tilt invariance the algorithm can achieve. Image recognition under a remarkable viewpoint change in practice requires that the scene is planar and Lambertian and its structures are not squashed when observed from an oblique viewpoint. Due to these physical limitations, affine image recognition is impractical under too big a tilt t. The physical upper bound $t_{\max}$ can be obtained experimentally using some images taken from indoor and outdoor scenes, each image pair being composed of a frontal view and an oblique view.

The images used in the experiments satisfy as much as possible the physical conditions mentioned above. The indoor scene is a magazine placed on a table with the artificial illumination coming from the ceiling as shown in Fig. 8.9. The outdoor scene is a building façade with some graffiti as illustrated in Fig. 8.10. For each pair of images, the true tilt parameter t between them is obtained by manual measurement. A-SIFT is applied with very large parameter sampling ranges and small sampling steps, so that the simulated views cover accurately the true affine distortion. The A-SIFT matching results depicted in Figs. 8.9 and 8.10 show that the limit is $t_{\max} \approx 5.6$ that corresponds to a view angle $\theta_{\max} = \arccos 1/t_{\max} \approx 80°$. A-SIFT finds a large number of matches when the tilt between the frontal image and the oblique image is smaller than about 5.6. Therefore we set the tilt simulation range $t_{\max} = 4\sqrt{2}$.

Let us emphasize that when the two images under comparison are taken from orthogonal longitude angles (see Fig. 8.7 as an example), i.e., $\phi = \phi' + \pi/2$, the maximum tilt invariance A-SIFT with $t_{\max} = 4\sqrt{2}$ can achieve in theory is about $t_{\max}^2 = 32$.

However, these experiments only fix reasonable bounds for all purpose algorithms. For high resolution images, for very flat lambertian surfaces, larger tilts might be recognizable.

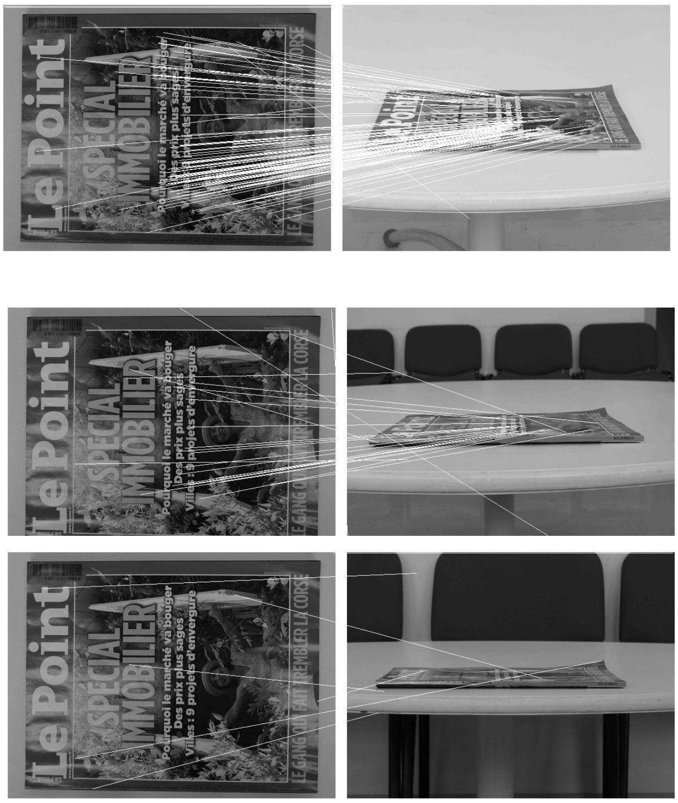

Figure 8.9: A-SIFT on an indoor scene. From top to bottom: tilt distortion t between the two images are respectively $t \approx 3, 5.2, 8.5$; the number of matches are respectively 107 (3 false), 25 (7 false), and 7 (all false).

8.4.2 Sampling steps

In order to have A-SIFT invariant to any affine transform, one needs to sample the tilt t and angle ϕ with a high enough precision. The sampling steps $\triangle t$ and $\triangle \phi$ must be fixed experimentally by testing several natural images.

The camera motion model illustrated in Fig. 8.4 indicates that the sampling precision of the latitude angle $\theta = \arccos 1/t$ should increase with θ. A geometric sampling for t satisfies this requirement. Naturally, the sampling ratio $\triangle t = t_{k+1}/t_k$ should be independent of the angle ϕ. In the sequel, the tilt sampling step is experimentally fixed to $\triangle t = \sqrt{2}$.

As can be observed from the camera motion model in Fig. 8.4, one needs a finer ϕ sampling when $\theta = \arccos 1/t$ increases: the image distortion caused

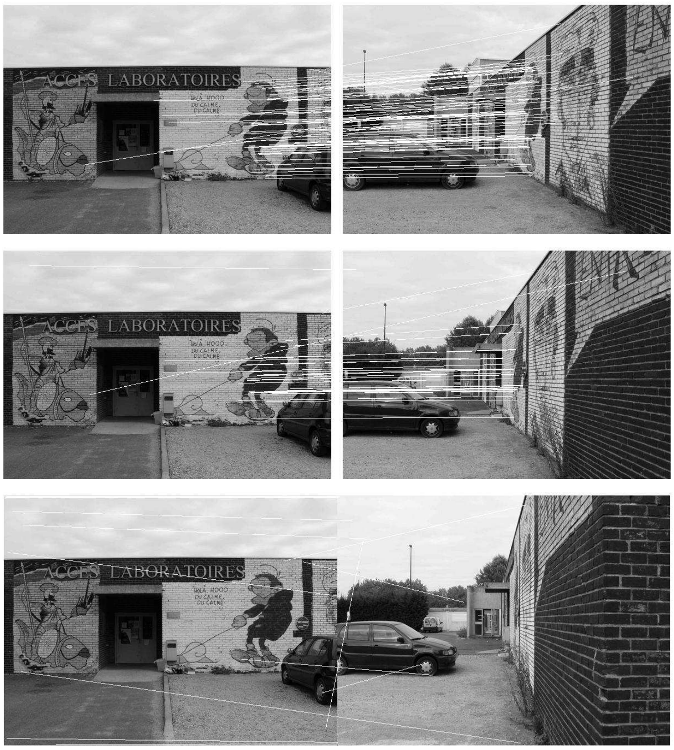

Figure 8.10: A-SIFT on an outdoor scene. From top to bottom: tilt distortion t between the two images are respectively $t \approx 3.8, 5.6, 8$; the number of matches are respectively 71 (4 false), 33 (4 false), 10 (all false).

by a fixed longitude angle displacement $\triangle\phi$, is much more drastic when the latitude angle θ increases. The longitude sampling step in the sequel will be $\triangle\phi = 2 \times \frac{36^\circ}{t} = \frac{72^\circ}{t}$.

Fig. 8.11 illustrates the sampling of the parameters $\theta = \arccos 1/t$ and ϕ. At bigger θ the sampling of θ as well as the sampling of ϕ are denser.

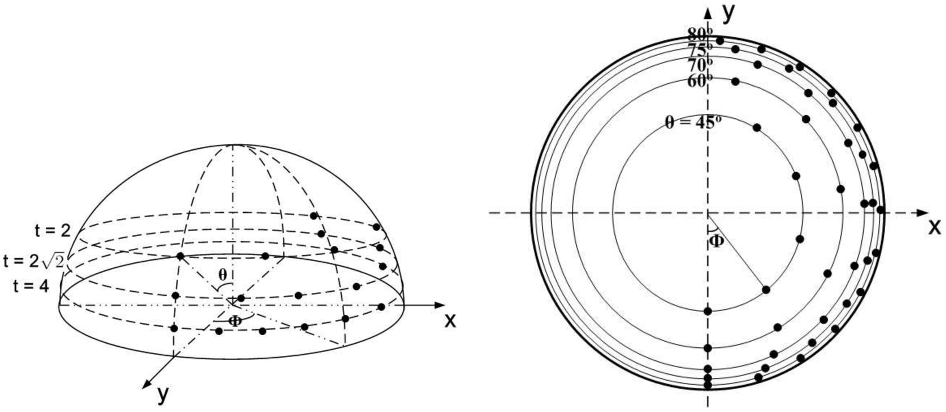

Figure 8.11: Sampling of the parameters $\theta = \arccos 1/t$ and ϕ. Black dots represent the sampling. Left: perspective illustration (only $t = 2, 2\sqrt{2}, 4$ are shown). Right: zenith view of the observation half sphere. The values of θ are indicated on the figure.

8.4.3 Acceleration with multi-resolution

The multi-resolution procedure accelerates A-SIFT by selecting the transforms that yield SIFT matches on low-resolution (LR) versions of the compared images. In case of success only, the procedure simulates the identified affine transforms on the query, and applies SIFT to compare them to the targets.

The multi-resolutions A-SIFT is summarized as follows.

1. Down-sample all compared digital images $\mathbf{u}$ and $\mathbf{v}$ by a $K \times K$ factor: $\mathbf{u}' = \mathbf{S}_K\mathbf{G}_K\mathbf{u}$ and $\mathbf{v}' = \mathbf{S}_K\mathbf{G}_K\mathbf{v}$, where $\mathbf{G}_K$ is an anti-aliasing gaussian discrete filter.

2. Low-resolution (LR) A-SIFT: perform A-SIFT between $\mathbf{u}'$ and $\mathbf{v}'$.

3. Identify the M affine transforms yielding the biggest numbers of matches between $\mathbf{u}'$ and $\mathbf{v}'$. They are retained only if the matches are meaningful. In practice, it is enough to put a threshold on the number k of matches, and $k = 15$ seems to be a good choice.

4. High-resolution (HR) A-SIFT: apply A-SIFT between $\mathbf{u}$ and $\mathbf{v}$ by simulating only the affine transforms previously identified.

Fig. 8.12 shows an example. The low-resolution A-SIFT that is applied on the 3×3 sub-sampled images finds 12 correspondences and identifies the 5 best affine transforms. The high-resolution A-SIFT finds 133 matches.

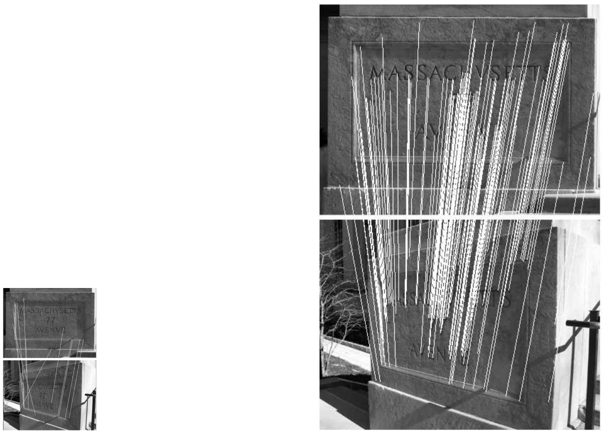

Figure 8.12: “77 Mass Ave”. Left: low-resolution A-SIFT, 12 matches, three of which are wrong. Right: high-resolution A-SIFT, 133 matches. Due to the lack of details in these images, the number of matches at low resolution is critically low.

8.4.4 A-SIFT Complexity

The complexity of the A-SIFT algorithm will be estimated under the recommended baseline configuration: The tilt and angle ranges are $[t_{\min}, t_{\max}] = [1, 4\sqrt{2}]$ and $[\phi_{\min}, \phi_{\max}] = [0°, 180°]$, and the sampling steps are $\triangle t = \sqrt{2}$, $\triangle \phi = 36° \times \frac{t}{2}$. Each t tilt is simulated by image sub-sampling in one direction by a t factor. All images are sub-sampled by a $K \times K = 3 \times 3$ factor for the low-resolution A-SIFT. Finally, the high-resolution A-SIFT simulates the $M = 5$ best affine transformations that are identified, but only in case they contain enough matches. When matching an image to a large database, the most common event is failure. Thus, the final high-resolution step is only to be taken into account when comparing images of the same scene.

The complexity of the descriptor computation is proportional to the input image area. This area is proportional to the number of simulated tilts t. Indeed, the number of ϕ simulations is proportional to t for each t, but the t sub-sampling for each tilt simulation divides the area by t. More precisely, the image area input to low-resolution A-SIFT is

$$\frac{1 + (|\Gamma_t| - 1)\frac{180°}{2\times 36°}}{K^2} = \frac{1 + 5 \times 2.5}{9} = 1.5$$

times as large as that of the original images, where $|\Gamma_t|$ is the number of tilt simulations. Thus, the complexity of the low-resolution A-SIFT is 1.5 times as much as that of a single SIFT routine, and generates 1.5 as many SIFs. Here we must distinguish two cases:

1. If the comparisons involve a large database (where most comparisons will be failures), the complexity is propositional to the number of SIFs in the queries multiplied by the number of SIFs in the targets. Since A-SIFT

introduces a a 1.5 area factor, the final complexity is simply $1.5^2 = 2.25$ times the SIFT complexity.

2. If the comparisons involve a set of images with high match likeliness, then the high resolution step is no more negligible. Then, it can only be asserted that the complexity will be less than $6.5 + 2.5 = 9$ times a SIFT routine on the same images. However, in that case, A-SIFT ensures many more detections than SIFT, because it explores many more viewpoint angles. Thus, the *complexity rate per detected SIF* might be much closer to, or even smaller than the per detection complexity in a SIFT routine.

For the high-resolution A-SIFT, this factor is $M = 5$. Therefore the total complexity of the A-SIFT is 6.5 times a SIFT routine.

The SIFT subroutines can be implemented in parallel in A-SIFT (for both the low-resolution and the high-resolution A-SIFT). Recently many authors have investigated SIFT accelerations [126, 85, 143]. A realtime SIFT implementation has been proposed in [238]. Obviously, all of these accelerations directly apply to A-SIFT.

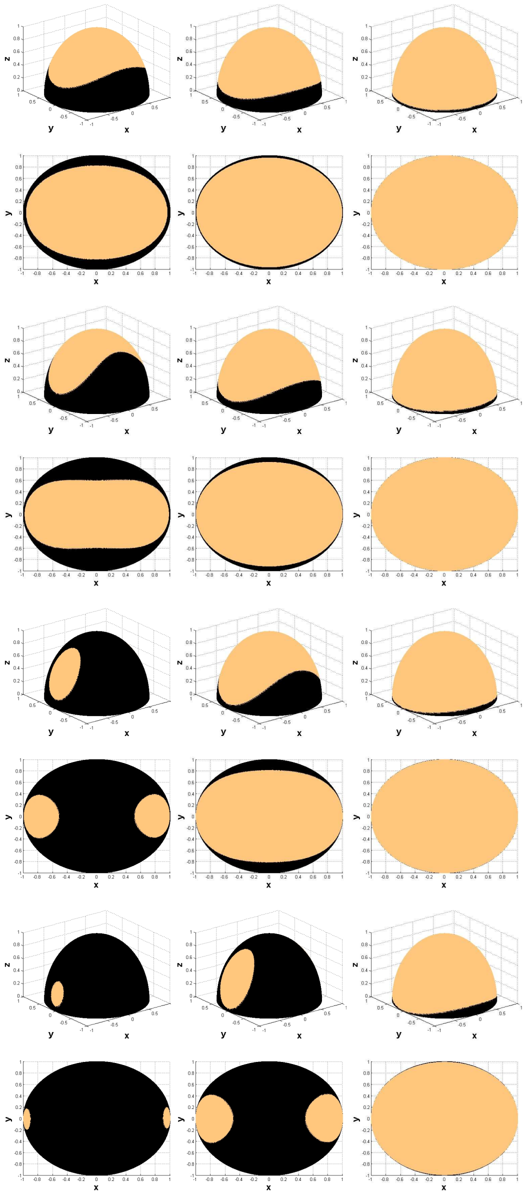

Figure 8.13: Each recognition method can be characterized by its transition tilt, namely the variation in relative observation angle compatible with recognition. This figure shows perspective and zenith views of the observation sphere. The central point of the bright region on the observation sphere is the original viewpoint from which an image has been taken. The rest of the bright region depicts, for several positions of this original point and several transition tilts, the attainable observation region, namely all other view angles in the observation sphere for which recognition succeeds for the given transition tilt. From top to bottom: latitude angle of the first image $\theta = 45,\ 60,\ 70,\ 80°$ that correspond respectively to the absolute tilts $t = \sqrt{2},\ 2,\ 2.9,\ 5.8$. From left to right: transition tilt $< 2.5,\ 5,\ 40$.

Chapter 9

The mathematical justification

This chapter gives the mathematical formalism and a mathematical proof that A-SIFT is fully affine invariant, up to sampling errors. The next chapter 10.3 is devoted to many comparative experiments where all mentioned state-of-the art algorithms are compared for their scale and tilt invariance.

In this chapter, to lighten the notation of the gaussian, G_σ will denote the convolution operator on $\mathbb{R}^2$ with the gauss kernel $\mathrm{G}_\sigma(x,y) = \frac{1}{2\pi(\mathbf{c}\sigma)^2} e^{-\frac{x^2+y^2}{2(\mathbf{c}\sigma)^2}}$, namely $\mathrm{G}u(x,y) =: (\mathrm{G} * u)(x,y)$. The constant $\mathbf{c} \geq 0.8$ is large enough to ensure that all considered images can be sampled with sampling mesh 1 after convolution with G_1. The one dimensional gaussians will be denoted by $\mathrm{G}^x_\sigma(x,y) = \frac{1}{\sqrt{2\pi}\mathbf{c}\sigma} e^{-\frac{x^2}{2(\mathbf{c}\sigma)^2}}$ and $\mathrm{G}^y_\sigma(x,y) = \frac{1}{\sqrt{2\pi}\mathbf{c}\sigma} e^{-\frac{y^2}{2(\mathbf{c}\sigma)^2}}$. G_σ still satisfies the semigroup property

$$\mathrm{G}_\sigma \mathrm{G}_\beta = \mathrm{G}_{\sqrt{\sigma^2+\beta^2}}. \tag{9.1}$$

The proof of the next formula is a mere change of variables in the integral defining the convolution.

$$\mathrm{G}_\sigma H_\gamma u = H_\gamma \mathrm{G}_{\sigma\gamma} u. \tag{9.2}$$

Exercise 9.1. Prove (9.2). ■

Using the above notation, the next paragraph formalizes the SIFT method.

9.0.5 The image formation model

As developed in Section 8.1, the whole image comparison process, based on local features, can proceed as though images where (locally) obtained by using digital cameras that stand far away, at infinity. The geometric deformations induced by the motion of such cameras are affine maps. A model is also needed for the two main camera parameters not deducible from its position, namely sampling and blur. The digital image is defined on the camera CCD plane. The pixel width can be taken as length unit, and the origin and axes chosen so that the camera pixels are indexed by $\mathbb{Z}^2$. The associated image sampling operator will be denoted by $\mathbf{S}_1$. The digital initial image is always assumed well-sampled and obtained by a gaussian blur with standard deviation 0.6. (See

[184] for a detailed analysis of why this model is sufficient and coherent for most digital images, and compatible with the SIFT method.) In all that follows, u_0 denotes the (theoretical) infinite resolution image that would be obtained by a frontal snapshot of a plane object with infinitely many pixels. The digital image obtained by any camera at infinity is $\mathbf{u} = \mathbf{S}_1 \mathrm{G}_1 A \mathcal{T} u_0$, where A is *any* linear map with positive singular values and $\mathcal{T}$ any plane translation. Thus we can summarize the general image formation model with cameras at infinity as follows.

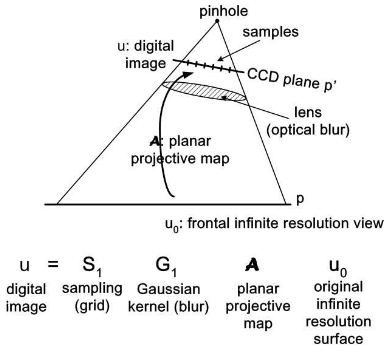

Figure 9.1: The projective camera model $u = \mathbf{S}_1 \mathrm{G}_1 \mathcal{A} u_0$. $\mathcal{A}$ is a planar projective transform (a homography). G_1 is an anti-aliasing gaussian filtering. $\mathbf{S}_1$ is the CCD sampling.

Definition 9.1. Image formation model. *Digital images of a planar object whose frontal infinite resolution image is u_0, obtained by a digital camera far away from the object, satisfy*

$$\mathbf{u} =: \mathbf{S}_1 \mathrm{G}_1 A \mathcal{T} u_0 \tag{9.3}$$

where A is any linear map and $\mathcal{T}$ any plane translation. G_1 *denotes a gaussian kernel broad enough to ensure no aliasing by 1-sampling, namely $I\mathbf{S}_1 \mathrm{G}_1 A \mathcal{T} u_0 = \mathrm{G}_1 A \mathcal{T} u_0$.*

The formal description of A-SIFT will be by far simpler if sampling issues do not interfere. All operations and all reasoning will be made with continuous well sampled images. It is easy to deduce afterwards the discrete operators acting on samples. $\mathcal{T}$ denotes an arbitrary translation, $\mathbf{R}$ an arbitrary rotation, $\mathbf{H}_\lambda$ an arbitrary homothety, and G an arbitrary gaussian convolution, all applied to continuous images. In the particular case in the digital image formation model (9.3) where A is a frontal view of u_0, $A = HR\mathcal{T}$ is the composition of a translation $\mathcal{T}$, a homothety H, and a rotation R. Thus the digital image is $\mathbf{u} = \mathbf{S}_1 \mathrm{G}_1 H \mathcal{T} R u_0$.

9.0.6 Inverting tilts

We shall denote by $*_y$ the 1-D convolution convolution operator in the y-direction. When we write $\mathrm{G}*_y$, we mean that G is a one-dimensional gaussian, depending on y, and the 1-D convolution means

$$\mathrm{G} *_y u(x,y) =: \int \mathrm{G}^y(z)u(x,y-z)dz.$$

There are *three different notions of tilt*, that we must carefully distinguish.

Definition 9.2. *Given $t > 1$, the tilt factor, define*

- *the absolute tilt : $T_t^x u_0(x,y) =: u_0(tx,y)$. In case this tilt is made in the y direction. It will be denoted by $T_t^y u_0(x,y) =: u_0(x,ty)$;*
- *the continuous tilt (taking into account camera blur): $\mathbb{T}_t^x v =: T_t^x \mathrm{G}^x_{\sqrt{t^2-1}} *_x v$. In case the simulated tilt is done in the x direction, it is denoted $\mathbb{T}_t^y v =: T_t^y \mathrm{G}^y_{\sqrt{t^2-1}} *_y v$.*
- *the digital tilt (transforming a digital image u into a digital image) : $\mathbf{u} \to \mathbf{S}_1 \mathbb{T}_t^x I \mathbf{u}$. This is the one that is used in the algorithm. It is correct because, as we shall see, the simulated tilt yields a blur permitting $\mathbf{S}_1$-sampling.*

If u_0 is an infinite resolution image observed with a t camera tilt in the x direction, the observed image is $\mathrm{G}_1 T_t^x u_0$. Our main problem is to reverse such tilts. This operation is in principle impossible, because absolute tilts do not commute with blur. However, the next lemma shows that $\mathbb{T}_t^y$ is actually a pseudo inverse to T_t^x.

Lemma 9.3. *One has*

$$\mathbb{T}_t^y = H_t \mathrm{G}^y_{\sqrt{t^2-1}} *_y (T_t^x)^{-1}.$$

Proof Since $(T_t^x)^{-1}u(x,y) = u(\frac{x}{t},y)$,

$$\left(\mathrm{G}^y_{\sqrt{t^2-1}} *_y (T_t^x)^{-1}u\right)(x,y) = \int \mathrm{G}_{\sqrt{t^2-1}}(z)u(\frac{x}{t}, y-z)dz.$$

Thus

$$H_t\left(\mathrm{G}^y_{\sqrt{t^2-1}} *_y (T_t^x)^{-1}u\right)(x,y) = \int \mathrm{G}_{\sqrt{t^2-1}}(z)u(x,ty-z)dz =$$

$$\left(\mathrm{G}^y_{\sqrt{t^2-1}} *_y u\right)(x,ty) = \left(T_t^y \mathrm{G}^y_{\sqrt{t^2-1}} *_y u\right)(x,y).$$

□

The meaning of the next result is that a tilted image $\mathrm{G}_1 T_t^x u$ can be tilted back by tilting in the orthogonal direction. The price to pay is a t zoom out. The second relation in the theorem means that the application of the simulated tilt to an image that can be well sampled by $\mathbf{S}_1$ yields an image that keeps that well sampling property.

Theorem 9.4. *Let $t \geq 1$. Then*

$$\mathbb{T}_t^y(\mathrm{G}_1 T_t^x) = \mathrm{G}_1 H_t; \qquad (9.4)$$

$$\mathbb{T}_t^y \mathrm{G}_1 = \mathrm{G}_1 T_t^y. \qquad (9.5)$$

Proof By Lemma 9.3, we recall that

$$\mathbb{T}_t^y = H_t \mathrm{G}^y_{\sqrt{t^2-1}} *_y (T_t^x)^{-1}.$$

Thus,

$$\mathbb{T}_t^y(\mathrm{G}_1 T_t^x) = H_t \mathrm{G}^y_{\sqrt{t^2-1}} *_y ((T_t^x)^{-1}\mathrm{G}_1 T_t^x). \tag{9.6}$$

By a variable change in the integral defining the convolution, it is an easy check that

$$(T_t^x)^{-1}\mathrm{G}_1 T_t^x u = \left(\frac{1}{t}\mathrm{G}_1(\frac{x}{t}, y)\right) * u, \tag{9.7}$$

and by the separability of the 2D gaussian in two 1D gaussians,

$$\frac{1}{t}\mathrm{G}_1(\frac{x}{t}, y) = \mathrm{G}_t(x)\mathrm{G}_1(y). \tag{9.8}$$

From (9.7) and (9.8) one obtains

$$(T^x)^{-1}\mathrm{G}_1 T_t^x u = ((\mathrm{G}_t^x(x)\mathrm{G}_1^y(y)) * u = \mathrm{G}_t^x(x) *_x \mathrm{G}_1^y(y) *_y u,$$

which implies

$$\mathrm{G}^y_{\sqrt{t^2-1}} *_y (T^x)^{-1}\mathrm{G}_1 T_t^x u = \mathrm{G}^y_{\sqrt{t^2-1}} *_y (\mathrm{G}_t^x(x) *_x \mathrm{G}_1^y(y) *_y u) = \mathrm{G}_t u.$$

Indeed, the 1D convolutions in x and y commute and $\mathrm{G}_t^y * \mathrm{G}^y_{\sqrt{t^2-1}} = \mathrm{G}_t^y$ by the Gaussian semigroup property (9.1). Substituting the last proven relation in (9.6) yields

$$\mathbb{T}_t^y \mathrm{G}_1 T_t^x u = H_t \mathrm{G}_t u = \mathrm{G}_1 H_t u.$$

The second relation (9.5) follows immediately by noting that $H_t = T_t^y T_t^x$. □

Exercise 9.2. Prove (9.7). ■

9.0.7 Proof that A-SIFT works

The meaning of Theorem 9.4 is that we can design an exact algorithm that simulates all inverse tilts for comparing two images. After interpolation, A-SIFT handles two images $u = \mathrm{G}_1 A\mathcal{T}_1 u_0$ and $v = \mathrm{G}_1 B\mathcal{T}_2 u_0$ that are two snapshots from different view points of a flat object whose front infinite resolution image is denoted by u_0. For a sake of simplicity, we break the symmetry, and set $\tilde{u}_0 =: A\mathcal{T}_1 u_0$, so that $u = \mathrm{G}_1\tilde{u}_0$ and $v = \mathrm{G}_1 B\mathcal{T}_2\mathcal{T}_1^{-1}A^{-1}\tilde{u}_0 = \mathrm{G}_1 BA^{-1}\mathcal{T}\tilde{u}_0$ for a translation $\mathcal{T}$ that depends on $\mathcal{T}_1$, $\mathcal{T}_2$, and A. Let us use the decomposition given by (8.2),

$$BA^{-1} = R_1 T_t^x H_\lambda R_2,$$

where R_1, R_2 are rotations, H_λ a zoom, and $T_t^x(x, y) = (tx, y)$ is the transition tilt from u to v. In summary A-SIFT has to compare the interpolated images

$$v = \mathrm{G}_1 R_1 T_t^x H_\lambda R_2 \mathcal{T}\tilde{u}_0 \text{ and } u = \mathrm{G}_1\tilde{u}_0.$$

The A-SIFT formal algorithm

The following algorithm, where image sampling issues are eliminated by interpolation, is actually a proof that A-SIFT manages to compare u and v obtained from u_0 by arbitrary camera positions at infinity. In this ideal algorithm, a ‘“dense enough” set of rotations and tilts is applied to v, so that each one of the simulated rotation-tilts is “close enough” to any other rotation-tilt. In the mathematical setting, this approximation must be infinitesimal. In the practical empirical setting, we’ll have to explore how dense the sets of rotations and tilts must be (see Section 8.4).

A-SIFT Algorithm (formal)

1. Apply a dense set of all possible rotations (and therefore also a rotation close to R_1^{-1}) to v. Thus, some of the simulated images will be arbitrary close to $v \rightarrow R_1^{-1}\mathrm{G}_1 R_1 H_\lambda T_t^x R_2 \mathcal{T}\tilde{u}_0 = \mathrm{G}_1 T_t^x H_\lambda R_2 \mathcal{T}\tilde{u}_0$;

2. apply in continuation a dense set of simulated tilts $\mathbb{T}_t^y$, and therefore also one arbitrary close to the right one $\mathbb{T}_t^y = T_t^y \mathrm{G}^y_{\sqrt{t^2-1}} *_y$, to $R_1^{-1}v = \mathrm{G}_1 T_t^x H_\lambda R_2 \mathcal{T}\tilde{u}_0$. By Theorem 9.4 we have the commutation $T_t^y \mathrm{G}_1 = \mathrm{G}_1 T_t^y$, which yields

$$\mathbb{T}_t^y R_1^{-1} v = \mathrm{G}_1 H_t H_\lambda R_2 \mathcal{T}\tilde{u}_0 = \mathrm{G}_1 H_{t\lambda} R_2 \mathcal{T}\tilde{u}_0;$$

3. perform a SIFT comparison of $\mathrm{G}_1 H_{t\lambda} R_2 \mathcal{T}\tilde{u}_0$, which is a frontal view of $\tilde{u}_0$, with $u = \mathrm{G}_1\tilde{u}_0$ which also is a frontal view of $\tilde{u}_0$.

The above algorithm description is also a proof of the following consistency theorem, since the SIFT method finds all SIFs common to two frontal views (Theorem 5.5).

Theorem 9.5. *Let $u = \mathrm{G}_1 A \mathcal{T}_1 u_0$ and $v = B\mathcal{T}_2 u_0$ be two images obtained from an infinite resolution image u_0 by cameras at infinity with arbitrary position and focal lengths. Then A-SIFT, applied with a dense set of tilts and longitudes, simulates two views of u and v that are obtained from each other by a translation, a rotation, and a camera zoom. As a consequence, these images match by the SIFT algorithm.*

Remark 9.6. *Even if the above proof, and the statement of Lemma* **??**, *deal with asymptotic statements when the sampling steps tend to infinity or when the SIFT scales tend to infinity, the approximation rate is very quick, a fact that can only be checked experimentally. This fact is actually extensively verified by the huge amount of experimental evidence on SIFT, that shows first that the recognition of scale invariant features (SIFs) is robust to a substantial variation of latitude and longitude, and second that the scale invariance is quite robust to moderate errors on scale. Section 8.4 has evaluated the adequate sampling rates and ranges for tilts and longitudes.*

Simulating midway tilts

The algorithm of Section 9.0.7 can be implemented in several ways. In the above description, the transition tilt T_t^x is directly inverted on one of the images. This strategy is consistent, but not optimal. As we have seen, the transition tilt can be very large. It is preferable to simulate moderate tilts on two images that large tilts on one of them. To this aim a *midway image* can be reached from both images by applying a $\sqrt{t}$ tilt to one of them and a $\sqrt{t}$ tilt to the other one. The only change to the formal algorithm will be that rotations and tilts are applied to both images, not just to one of them.

Midway A-SIFT (formal)

1. Apply a dense set of all possible rotations to both images, and therefore R_2 to u and R_1^{-1} to v;
2. apply in continuation a dense set of simulated tilts $\mathbb{T}_t^x$ in a fixed $[0, t_{max}]$ range;
3. perform a SIFT comparison of all pairs of resulting images.

Let us now prove that this algorithm works, namely that two of the simulated images are deduced from each other by a similarity. The query and target images are $u = \mathrm{G}_1 A \mathcal{T}_1 u_0$ and $v = \mathrm{G}_1 B \mathcal{T}_2 u_0$. By the usual decomposition of a linear map (8.2),

$$BA^{-1} = R_1 T_t^x R_2 H_\lambda = (R_1 T_{\sqrt{t}^x})(T^x_{\sqrt{t}} R_2 H_\lambda).$$

Notice that by the relation

$$\mathbb{T}_t^x R(-\frac{\pi}{2}) = R(\frac{\pi}{2})\mathbb{T}_t^y, \tag{9.9}$$

the algorithm also simulates tilts in the y direction, up to $R(\frac{\pi}{2})$ rotation. In particular, the above algorithm applies:

1. $\mathbb{T}^x_{\sqrt{t}} R_2$ to $\mathrm{G}_1 A \mathcal{T}_1 u_0$, which by (9.5) yields $\tilde{u} = \mathrm{G}_1 T^x_{\sqrt{t}} R_2 A \mathcal{T}_1 u_0 =: \mathrm{G}_1 \tilde{A} \mathcal{T}_1 u_0$;
2. $R(\frac{\pi}{2})\mathbb{T}^y_{\sqrt{t}} R_1^{-1}$ to $\mathrm{G}_1 B \mathcal{T}_2 u_0$, which by (9.5) yields $\mathrm{G}_1 R(\frac{\pi}{2}) T^y_{\sqrt{t}} R_1^{-1} B \mathcal{T}_2 u_0 =:$ $\mathrm{G}_1 \tilde{B} \mathcal{T}_2 u_0$.

Let us show that $\tilde{A}$ and $\tilde{B}$ only differ by a similarity. Indeed,

$$\tilde{B}^{-1} R(\frac{\pi}{2}) H_{\sqrt{t}} \tilde{A} = B^{-1} R_1 T^y_{\sqrt{t}^{-1}} T^x_{\sqrt{t}} H_{\sqrt{t}} R_2 A =$$

$$B^{-1} R_1 T^y_{\sqrt{t}^{-1}} T^x_{\sqrt{t}} H_{\sqrt{t}} R_2 A = B^{-1} R_1 T_t^x R(\frac{\pi}{2}) R_2 A = B^{-1}(BA^{-1})A = I,$$

where I is the identity. It follows that $\tilde{B} = R(\frac{\pi}{2}) H_{\sqrt{t}} \tilde{A}$. Thus,

$$\tilde{u} = \mathrm{G}_1 \tilde{A} \mathcal{T}_1 u_0 \text{ and } \tilde{v} = \mathrm{G}_1 R(\frac{\pi}{2}) H_{\sqrt{t}} \tilde{A} \mathcal{T}_2 u_0,$$

that are two of the simulated images, are deduced from each other by a rotation and a $\sqrt{t}$ zoom. It follows that their SIFs are identical as soon as the scale of the SIF exceeds $\sqrt{t}$.

□

Exercise 9.3. There is a (non crucial) error in the above proof. Read this proof carefully, find the error, and adjust the proof. ■

9.0.8 Conclusion on the algorithms

The above descriptions have neglected the sampling issues, but care was taken that input images and output images be always written in the $\mathrm{G}_1 u$ form. For the digital input images, that always have the form $\mathbf{u} = \mathbf{S}_1 \mathrm{G}_1 u_0$, the Shannon interpolation algorithm is I is first applied, to give back $I\mathbf{S}_1 \mathrm{G}_1 u_0 = \mathrm{G}_1 u_0$. For the output images, that always have the form $\mathrm{G}_1 v$, the sampling $\mathbf{S}_1$ gives back a digital image.

Thus, the descriptions of the formal algorithm A-SIFT and of its "midway" version are changed into a digital algorithm by:

- replacing everywhere the inputs $\mathrm{G}_1 u$ by their digital version $\mathbf{S}_1 \mathrm{G}_1 u$;
- by applying digital rotations to digital images : $\mathbf{u} \to \mathcal{R}\mathbf{u} =: \mathbf{S}_1 R I \mathbf{u}$;
- by applying digital tilts as defined in Def. 9.2, namely $\mathbf{u} \to \mathbf{S}_1 \mathcal{T}_t^x I \mathbf{u}$.

That way, the formal algorithms are transformed into digital algorithms. The proofs need not be repeated, since by Shannon interpolation and sampling, it is equivalent to talk about $\mathbf{S}_1 \mathrm{G}_1 u_0$ or about $\mathrm{G}_1 u_0$.

Clearly the midway algorithm is better, because it only needs simulating tilts that are square roots of the real transition tilts. Thus, all of the forthcoming discussion will focus on the midway version, that we'll simply call A-SIFT.

Chapter 10

Experiments on affine invariant methods

10.1 Affine normalization methods: are they fully affine invariant?

Since the affine transform depends upon six parameters, it is out of the question to just simulate all of them and compare the original image to all deformed images by all possible affine deformations. However, *simulation* can be a solution for a few parameters: the SIFT method actually simulates zooms out.

The other way that has been tried by many authors is *normalization*. Normalization is a magic method that, given a patch that has undergone an unknown affine transform, transforms the patch into a standardized one, where the effect of the affine transform has been eliminated (see Fig. 10.1). Normalization by translation is easily achieved: A patch around (x_0, y_0) is translated back to a patch around $(0, 0)$. A rotational normalization requires a circular patch. In this patch, a principal direction is found, and the patch is rotated so that this principal direction coincides with a fixed direction. Thus, of the six parameters in an affine transform, at least three are easily eliminated by normalization.

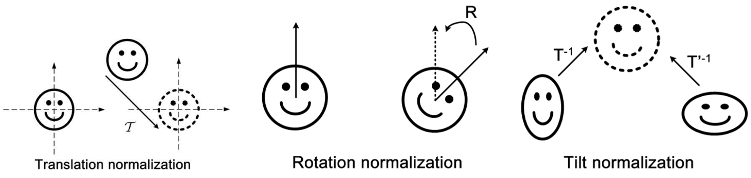

Figure 10.1: Normalization methods can eliminate the effect of a a class of affine transforms by associating the same standard patch to all transformed patches.

However, when it comes to the other three parameters, things get difficult and controversial. Two methods have been recently proposed to perform a level lines based full affine normalization: MSER [168] and LLD [189]. Both of them apply to image level lines, or to image pieces of level lines, an affine normalization in the spirit of the translation and rotation normalization explained above. We shall focus on MSER, but the discussion applies to LLD as well.

10.2 Global Normalization and Encoding

10.2.1 Global Affine Normalization

Classical shape normalization methods are based on the inertia matrix normalization. We shall use Cohignac's presentation of this method, given in [41]. Denote by $\mathbf{1}_{\mathcal{F}}$ the indicator function of a shape domain $\mathcal{F}$. The shape is usually associated a weight function defined on the shape domain, $u(x,y) = \varphi(x,y)\mathbf{1}_{\mathcal{F}}$. Classically, $\varphi(x,y)$ is the restriction of an image to the shape extracted from it, or the restriction to the shape of the image gradient, or the restriction to the shape of the image gradient direction, etc. Thus when we talk about "the shape $\mathcal{F}$", we actually talk about $u(x,y)$. Of course, when $\varphi(x,y)=1$, the whole analysis of this section boils down to the analysis of the proper shape $u = \mathbf{1}_{\mathcal{F}}$. In all that follows, a convenient abbreviation is $\mathcal{F}$ for $u(x,y)$. If A is linear or affine map, $A\mathcal{F}$ denotes the function Au defined as usual by $Au(x,y) = u(A(x,y))$. In order to achieve translation invariance of the normalized representation, it may be assumed that $\mathcal{F}$ has been previously translated so that its barycenter weighted by $u(x,y)$ is at the origin of the image plane.

Exercise 10.1. Show that this last assumption amounts to assume that

$$\mu_{1,0}(\mathcal{F}) =: \int_{I\!R^2} xu(x,y)dxdy = 0 \ \text{ and } \ \mu_{0,1}(\mathcal{F}) =: \int_{I\!R^2} yu(x,y)dxdy = 0.$$

More precisely, give a formula for the weighted barycenter $b(\mathcal{F})$ of $\mathcal{F}$ and show that the barycenter is covariant by any affine transform, namely $b(A\mathcal{F}) = Ab(\mathcal{F})$ for every linear transform A. Show that the weighted area of $\mathcal{F}$ defined by $\mu_{0,0}(\mathcal{F}) =: \int_{I\!R^2} u(x,y)dxdy$ satisfies $\mu_{0,0}(A\mathcal{F}) = |det(A)|\mu_{0,0}(\mathcal{F})$. ■

The moment of order (p,q) (p and q natural integers) of the shape $\mathcal{F}$ (weighted by u) is defined by

$$\mu_{p,q}(\mathcal{F}) = \int_{I\!R^2} x^p y^q u(x,y)dxdy.$$

Let $S_{\mathcal{F}}$ be the following 2×2 positive-definite, symmetric matrix

$$S_{\mathcal{F}} = \frac{1}{\mu_{0,0}} \begin{pmatrix} \mu_{2,0} & \mu_{1,1} \\ \mu_{1,1} & \mu_{0,2} \end{pmatrix},$$

where $\mu_{i,j} = \mu_{i,j}(\mathcal{F})$. By the uniqueness of Cholesky factorization [94], $S_{\mathcal{F}}$ may be uniquely decomposed as $S_{\mathcal{F}} = B_{\mathcal{F}} B_{\mathcal{F}}^{\mathrm{t}}$ where $B_{\mathcal{F}}$ is a lower-triangular real matrix with positive diagonal entries.

Definition 10.1. *The* pre-normalized shape *associated to $\mathcal{F}$ is the shape $\mathcal{F}' = B_{\mathcal{F}}^{-1}(\mathcal{F})$.*

The aim is to prove that the pre-normalized solid shape is invariant to affine transformations, up to a rotation.

Lemma 10.2. *Let A be a non-singular 2×2 matrix. Then $S_{A\mathcal{F}} = AS_{\mathcal{F}}A^{\mathrm{t}}$.*

Proof. Let a, b, c and d be real numbers such that:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

The moment of order $(2, 0)$ associated to the solid shape $A\mathcal{F}$ is

$$\begin{aligned} \mu_{2,0}(A\mathcal{F}) &= \det(A) \int_{\mathbb{R}^2} (ax + by)^2 u(x, y) dx dy \\ &= \det(A)(a^2 \mu_{2,0} + 2ab\mu_{1,1} + b^2 \mu_{0,2}). \end{aligned}$$

The same computation for moments of order $(0, 2)$ and $(1, 1)$ yields

$$\begin{aligned} \mu_{0,2}(A\mathcal{F}) &= \det(A)(c^2 \mu_{2,0} + 2cd\mu_{1,1} + d^2 \mu_{0,2}), \\ \mu_{1,1}(A\mathcal{F}) &= \det(A)(ac\mu_{2,0} + bd\mu_{0,2} + (ad + bc)\mu_{1,1}). \end{aligned}$$

Since $\mu_{0,0}(A\mathcal{F}) = \det(A)\mu_{0,0}$, one can easily check that $S_{A\mathcal{F}} = AS_{\mathcal{F}}A^{\mathrm{t}}$. □

Exercise 10.2. Make the calculation proving that $S_{A\mathcal{F}} = AS_{\mathcal{F}}A^{\mathrm{t}}$. ■

Lemma 10.3. *Let X_0 be a 2×2 invertible matrix. Then, for any 2×2 matrix X: $XX^{\mathrm{t}} = X_0 {X_0}^{\mathrm{t}}$ if and only if there exists an orthogonal matrix Q such that $X = X_0 Q$.*

Proof. Since X_0 is invertible, $XX^{\mathrm{t}} = X_0 {X_0}^{\mathrm{t}}$ iff $X_0^{-1} X (X_0^{-1} X)^{\mathrm{t}} = \mathrm{Id}_2$. Letting $Q = X_0^{-1} X$ yields the result. □

Proposition 10.4. *The pre-normalized solid shape is invariant to any invertible, planar, linear transformation $(x, y)^{\mathrm{t}} \mapsto A(x, y)^{\mathrm{t}}$, up to an orthogonal transformation. Moreover, if $\det(A) > 0$, the invariance holds up to a rotation.*

Proof. Since A is a 2×2 non singular matrix, following Lemma 10.2, $S_{A\mathcal{F}} = AS_{\mathcal{F}}A^{\mathrm{t}}$. By letting $B_{\mathcal{F}}$ be the lower-triangular matrix of Cholesky's decomposition of $B_{\mathcal{F}}$, it follows that $S_{A\mathcal{F}} = AB_{\mathcal{F}}(AB_{\mathcal{F}})^{\mathrm{t}}$. Now, since $S_{A\mathcal{F}}$ is a 2×2 positive-definite, symmetric matrix, Cholesky factorization yields $S_{A\mathcal{F}} = B_{A\mathcal{F}} {B_{A\mathcal{F}}}^{\mathrm{t}}$, where $B_{A\mathcal{F}}$ is a 2×2 non-singular, lower-triangular real matrix. Then, by Lemma 10.3, $B_{A\mathcal{F}} = AB_{\mathcal{F}}Q$, where Q is a 2×2 orthogonal matrix. Hence, $B_{A\mathcal{F}}^{-1} A\mathcal{F} = (AB_{\mathcal{F}}Q)^{-1} A\mathcal{F} = Q^{-1} B_{\mathcal{F}}^{-1} A^{-1} A\mathcal{F} = Q^{-1} B_{\mathcal{F}}^{-1} \mathcal{F}$, which proves the invariance of $\mathcal{F}' = B_{\mathcal{F}}^{-1}\mathcal{F}$ to planar isomorphisms, up to an orthogonal transformation. Finally, notice that if $\det(A) > 0$, then $\det(Q) > 0$. □

Exercise 10.3. Prove that a closed form for $B_{\mathcal{F}}^{-1}$ in terms of the moments of $\mathcal{F}$ can be computed by taking the inverse of $B_{\mathcal{F}}$, the lower-triangular matrix given by the Cholesky decomposition of $S_{\mathcal{F}}$,

$$B_{\mathcal{F}}^{-1} = \sqrt{\mu_{0,0}} \begin{pmatrix} \frac{1}{\sqrt{\mu_{2,0}}} & 0 \\ -\frac{\mu_{1,1}}{\mu_{2,0}\sqrt{\mu_{0,2} - \frac{\mu_{1,1}^2}{\mu_{2,0}}}} & \frac{1}{\sqrt{\mu_{0,2} - \frac{\mu_{1,1}^2}{\mu_{2,0}}}} \end{pmatrix}.$$

The pre-normalized solid shape $\mathcal{F}' = B_{\mathcal{F}}^{-1}\mathcal{F}$ is then an affine invariant representation of $\mathcal{F}$ *modulo* a rotation. In order to obtain a full affine invariant representation, only a reference angle is needed. This can be achieved, for instance, by computing in polar coordinates

$$\varphi = \mathrm{Arg}\left(\int_0^{2\pi}\int_0^{+\infty}(B_{\mathcal{F}}^{-1}u)(r,\theta)e^{i\theta}rdrd\theta\right),$$

then rotating $\mathcal{F}'$ by $-\varphi$.

Putting all the steps together, the support of the affine invariant normalization of $\mathcal{F}$ is the set of points (x_N, y_N) given by

$$\begin{pmatrix} x_N \\ y_N \end{pmatrix} = \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} B_{\mathcal{F}}^{-1} \begin{pmatrix} x - \mu_{1,0} \\ y - \mu_{0,1} \end{pmatrix},$$

for all $(x, y) \in \mathcal{F}$.

10.2.2 Maximally Stable Extremal Regions (MSER)

The MSER method introduced by Matas et al. [168] attempts to achieve affine invariance by selecting the most robust connected components of upper and lower level sets as image features.

Extremal regions is the name given by the authors to the connected components of upper or lower level sets. Maximally stable extremal regions, or MSERs, are defined as maximally contrasted regions in the following way. let $Q_1, ..., Q_{i-1}, Q_i, ...$ be a sequence of nested extremal regions, i.e. $Q_i \subset Q_{i+1}$ where Q_i is defined by a threshold at level i or, in other terms, Q_i is an upper (resp. lower) level set at level i. An extremal region in the list Q_{i_0} is said to be maximally stable if the area variation $q(i) =: |Q_{i+1} \setminus Q_{i-1}|/|Q_i|$ has a local minimum at i_0, where $|Q|$ denotes the area of a region $|Q|$. Clearly the above measure is a measure of contrast along the boundary ∂Q_i of Q_i. Indeed, assuming that u is C^1 and that the grey level increment between i and $i + 1$ is infinitesimal, the area $|Q_{i+1} \setminus Q_{i-1}|$ varies least when $\int_{\partial Q_i} |\nabla u|$ is maximal. The MSER extraction is a first step of image matching. Once MSERs are computed, the affine normalization of Section 10.2.1 is performed on the MSERs before they can be compared. The fact that the method is not fully scale invariant is easily explained with the experiment of Fig. 10.2. In MSER the scale normalization is based on the size (area) of the detected extremal regions. However, scale change is not just a homothety: it involves a blur followed by sub-sampling. The blur changes drastically the size of the regions. As pointed out in [42] this entails a strong lack of scale invariance. It could only be compensated by simulating actual blur on the images, as made by the SIFT method.

10.3 Experiments

A-SIFT image matching performance will be compared with the state-of-the-art approaches with the detectors DoG [151], Hessian-Affine, Harris-Affine [172, 175] and MSER [168] all coded by the most popular SIFT descriptor [151]. The MSER detector followed by the correlation descriptor as proposed in the original work [168] is also included in the comparison, whose performance will be shown

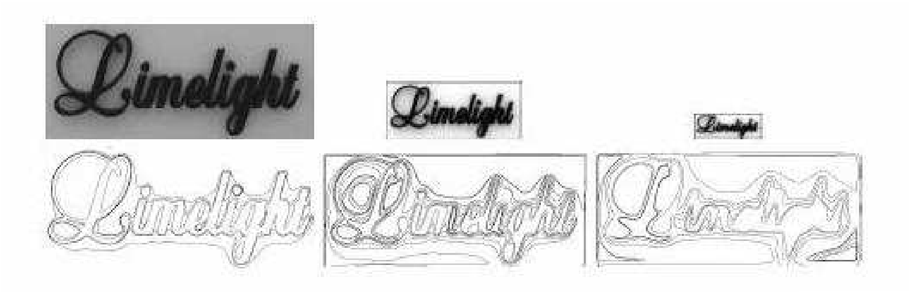

Figure 10.2: **Shapes change with distance: The level lines not stable by down-sampling. This is the main problem with level lines methods (MSER).**

slightly worse than that of the MSER detector followed by the SIFT descriptor. For simplicity, in the text the methods will be named respectively SIFT, Harris-Affine, Hessian-Affine and MSER for short. (The MSER detector followed by the SIFT and the correlation descriptors are sometimes denoted as MSER+SIFT and MSER+Corr. By MSER alone, we mean the MSER detector followed by either of the two descriptors.)

The Lowe [150] reference software was used for DoG with SIFT. For all the other methods we used the Hessian-Affine, Harris-Affine and MSER descriptor code provided by the authors and combined them with the SIFT descriptor implemented by Mikolajczyk, all downloadable from [171].

Applications of A-SIFT and comparisons with the other methods will also be performed for video object tracing and symmetry detection.

The experiments will show images taken from different viewpoints with varying tilts, zooms, and transition tilts. Correspondences will be connected by white segments. Note that the parallelism or coherent directions of the connecting lines usually indicates that most correspondences are correct.

All images under comparison have a low resolution 600×450. As reported under each figure, A-SIFT applied an image sub-sampling of factor $K \times K$ with $K = 3$ for most images. A very few cases where objects of interest are too small will be shown. In those cases A-SIFT only works with a more conservative subsampling $K = 2$, and in one case only with $K = 1$.

10.3.1 Exploring tilts and zooms

Fig. 10.3 illustrates the two settings that we have adopted to make systematic comparisons respectively for evaluating the maximum absolute tilt and transition tilt attained by each algorithm. A magazine and a painting shown in Fig. 10.4 were photographed for the experiments. Unlike SIFT and A-SIFT, the Hessian-Affine, Harris-Affine and MSER detectors are not robust to scale change as shown in Fig. 10.6. Therefore the pairs of images under comparison were chosen free of scale change so that the evaluation is focused on the tilt invariance.

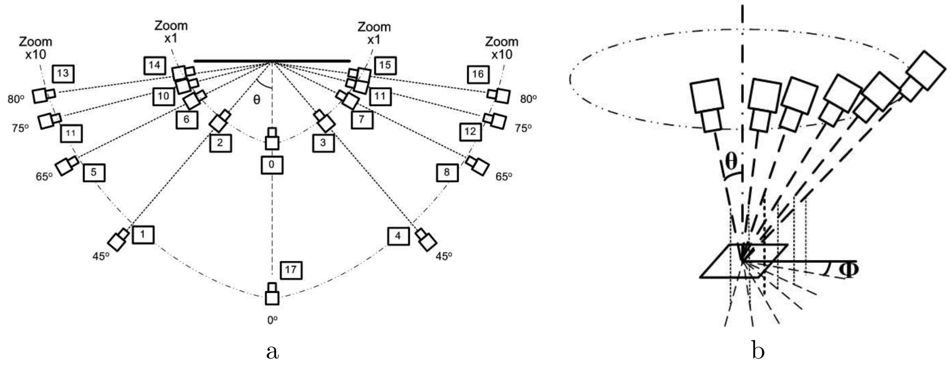

Figure 10.3: The settings adopted to systematic comparison. Left: absolute tilt test. An object is photographed with a latitude angle θ (between the camera axis and the normal to the object) that varies from 0 degree (frontal view) to 80 degrees, from distances varying between 1 and 10, which is the maximum focal distance change. Right: transition tilt test. An object is photographed with a longitude angle ϕ (between the camera axis projected on the object plane and a fixed direction thereon) that varies from 0 degree to 90 degrees, from a fixed distance.

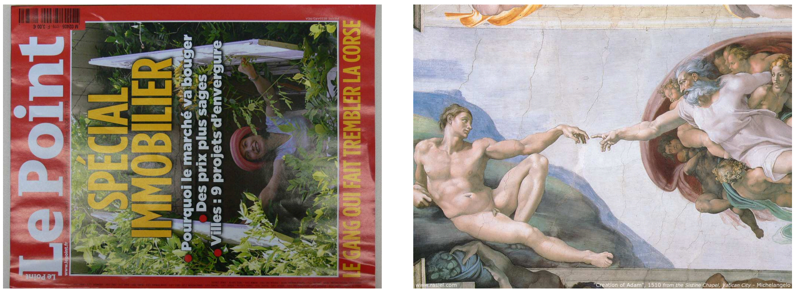
Le Point
SPÉCIAL IMMOBILIER
Pourquoi le marché va bouger
Des prix plus sages
Villes : 9 projets d'envergure
LE GANG QUI FAIT TREMBLER LA CORSE

Figure 10.4: The magazine cover and the painting are photographed in the experiments.

10.3.2 Absolute Tilt Tests

The painting illustrated in Fig. 10.4 is photographed with a reflex camera, with distances varying between ×1 and ×10, which is the maximum focal distance change, and with viewpoint angles between the camera axis and the normal to the poster that varies from 0 degree (frontal view) to 80 degrees. It is clear that beyond 80 degrees, to establish a correspondence between the frontal image and the extreme viewpoint becomes absolutely haphazard. Even when the photo acquisition conditions and the image resolution are excellent, with such a big view angle change the observed surface becomes in general reflective, and the image in the resulting photo is totally different from the frontal view. Nevertheless, A-SIFT works until 80 degrees, and it would be unrealistic to insist on bigger angles.

Table 10.1 summarizes performance of each algorithm in terms of number of correct matches. Some matching results are illustrated in Figs. 10.7 to 10.10.

One remarks first that MSER, which uses maximally stable level lines as features, obtains for most time much less correspondences than other methods whose features are based on local maxima in the scale-space. This has been confirmed by LLD, a novel image matching approach independently developed at ENS Cachan that applies also level lines as features [191, 189]. Let us recall that robust image matching requires a sufficiently big number of correspondences.

For images taken at short distance as illustrated in Figs. 10.7 and 10.8, tilt varies on the same flat object because of the perspective effect, as illustrated in Fig. 10.5. The number of SIFT correspondences drops dramatically when the angle is bigger than 65 degrees (that corresponds to a tilt $t \approx 2.3$) and it fails completely when the angle exceeds 75 degrees (tilt $t \approx 3.8$). At 65 and 75 degrees, as shown in Fig. 10.8, most matches are located on the side closer to the camera where the actual tilt is smaller. The performance of Harris-Affine and Hessian-Affine degrades considerably when the angle goes over 75 degrees (tilt $t \approx 3.8$). The MSER correspondences remain at a small number with a noticeable decline over 65 degrees (tilt $t \approx 2.4$). A-SIFT works perfectly until 80 degrees (tilt $t \approx 5.8$).

Images taken at a camera-object distance multiplied by 10, as shown in Figs. 10.9 and 10.10, exhibits less perspective effects but contains less meaningful pixels at big angles. For these images the SIFT performance drops considerably: recognition is possible only with angles smaller than 45 degrees. The performance of Harris-Affine and Hessian-Affine declines clear when the angle goes from 45 to 65 degrees and beyond 65 degrees they fail completely. MSER struggles at the angle of 45 degrees and fails at 65 degrees. A-SIFT again functions perfectly until 80 degrees.

Rich in highly contrasted regions, the magazine shown in Fig. 10.4 is more favorable to MSER. Table 10.2 shows the result of a similar experiment performed with the magazine, with the latitude angles from 50 to 80 degrees on one side and with the camera focus distance ×4. Fig. 10.11 shows the result with 80-degree angle. The performance of SIFT, Harris-Affine and Hessian-Affine drops dramatically with the angle going from 50 to 60 degrees (tilt t from 1.6 to 2). Beyond 60 degrees (tilt $t = 2$) they all fail completely. MSER finds many correspondences until 70 degrees (tilt $t \approx 2.9$). The number drops considerably when the angle exceeds 70 degrees and becomes too small at 80 degrees (tilt $t \approx 5.8$) for robust recognition. A-SIFT works perfectly until 80

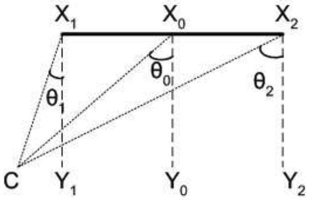

Figure 10.5: When the camera view angle is large, the absolute tilt of a plane object can vary considerably in the same image.

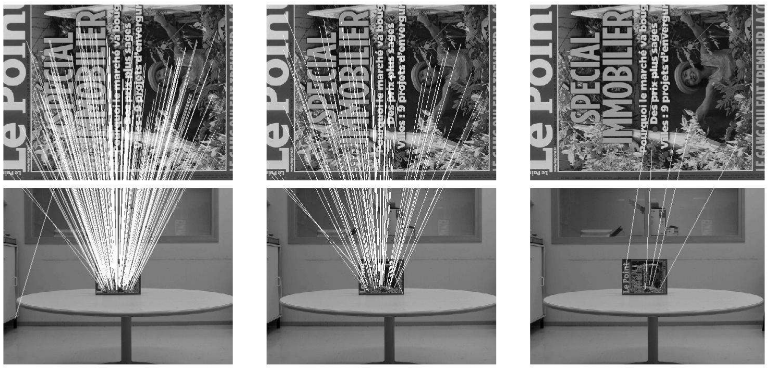

Figure 10.6: Robustness to scale change. A-SIFT (shown), SIFT (shown), Harris-Affine (shown), Hessian-Affine, MSER+Corr and MSER+SIFT find respectively 221, 86, 4, 3, 3 and 4 correct matches. Harris-Affine, Hessian-Affine and MSER are not robust to scale change. A-SIFT is implemented with $K = 2$, which means that $K = 3$ doesn't work.

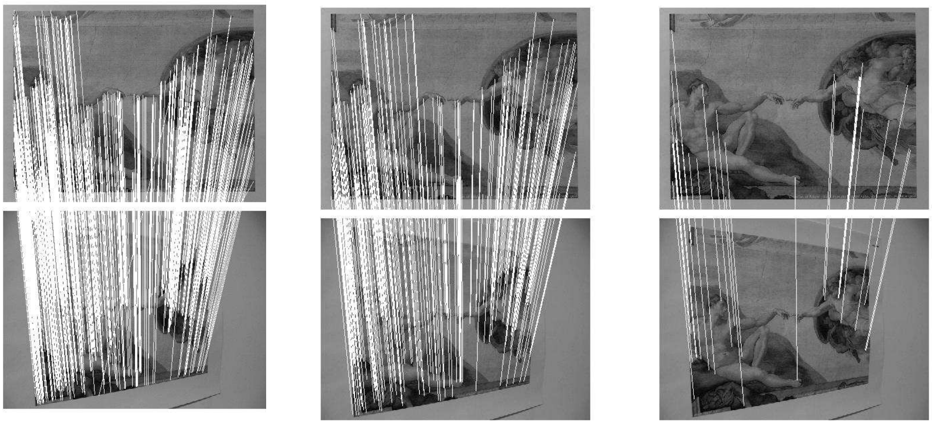

Figure 10.7: Correspondences between the painting images taken from short distance (zoom ×1) at frontal view and at $-45°$ angle. The absolute tilt varies: $t = 2$ (middle), $t < 2$ (left part), $t > 2$ (right part). A-SIFT (shown), SIFT (shown), Harris-Affine (shown), Hessian-Affine, MSER+Corr and MSER+SIFT find respectively 624, 236, 28, 15, 7 and 11 correct matches. A-SIFT is implemented with $K = 3$.

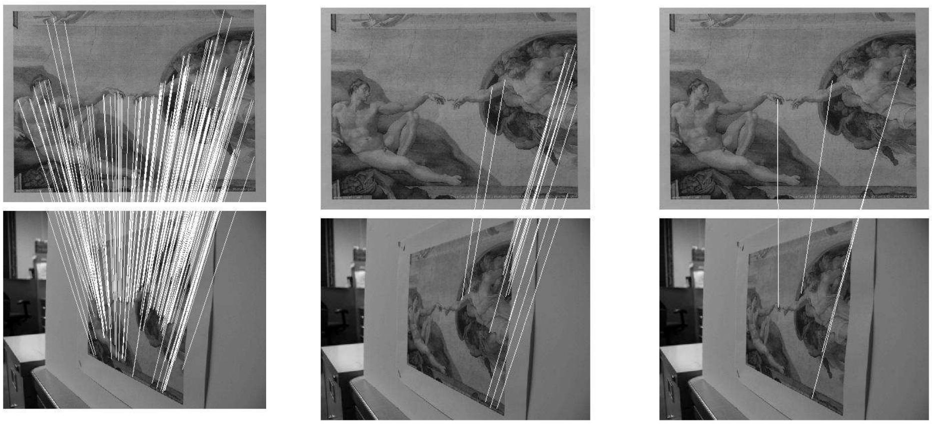

Figure 10.8: Correspondences between the painting images taken from short distance (zoom ×1) at frontal view and at 75° angle. The local absolute tilt varies: $t = 4$ (middle), $t < 4$ (right part), $t > 4$ (left part). A-SIFT ($K = 3$, shown), SIFT (shown), Harris-Affine, Hessian-Affine, MSER+Corr and MSER+SIFT (shown) find respectively 202, 15, 3, 1, 5 and 5 correct matches.

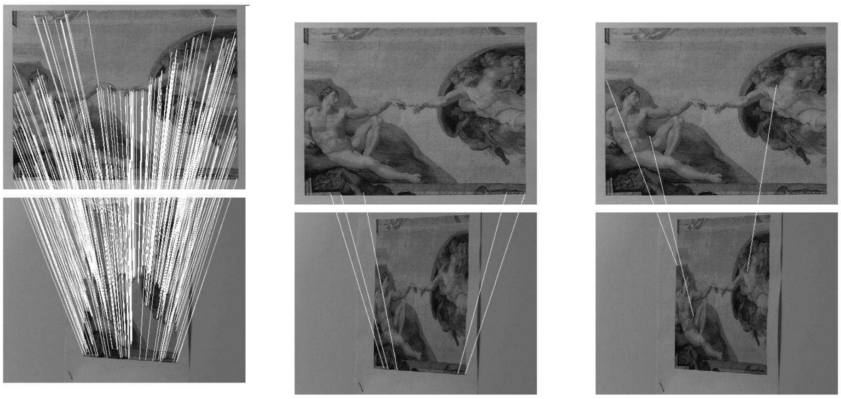

Figure 10.9: Correspondences between long distance snapshots (zoom ×10) at frontal and 65° angle, absolute tilt $t = 2.4$. A-SIFT ($K = 3$, shown), SIFT (shown), Harris-Affine (shown), Hessian-Affine, MSER+Corr and MSER+SIFT find respectively 341, 5, 3, 0, 3 and 4 correct matches.

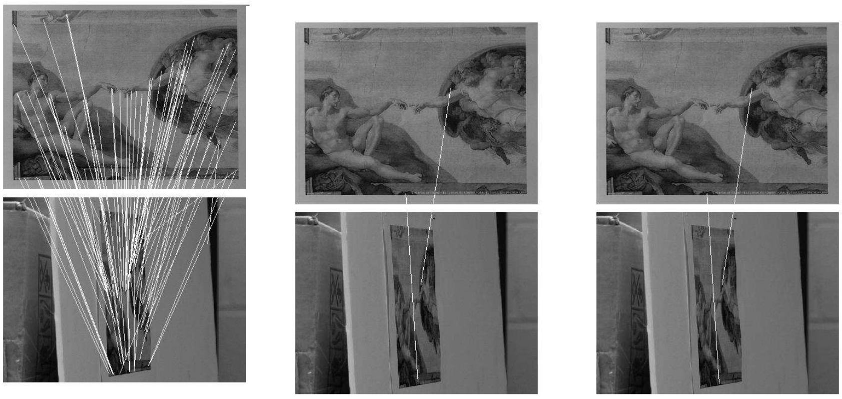

Figure 10.10: Correspondences between long distance views (zoom ×10), frontal view and 80° angle, absolute tilt $t = 5.8$. A-SIFT ($K = 3$, shown), SIFT, Harris-Affine, Hessian-Affine, MSER+Corr (shown) and MSER+SIFT (shown) find respectively 75, 1, 1, 0, 2 and 2 correct matches.

Z × 1	SIFT	Haraff	Hesaff	MSER+Corr	MSER+SIFT	A-SIFT
−80°/5.8	1	16	1	3	4	110
−75°/3.9	24	36	7	4	3	281
−65°/2.3	117	43	36	6	5	483
−45°/1.4	245	83	51	9	13	559
45°/1.4	195	86	26	10	12	428
65°/2.3	92	58	32	10	11	444
75°/3.9	15	12	7	7	7	203
80°/5.8	2	6	6	5	5	204

Z × 10	SIFT	Haraff	Hesaff	MSER+Corr	MSER+SIFT	A-SIFT
−80°/5.8	1	3	0	4	4	116
−75°/3.9	0	3	0	6	6	265
−65°/2.3	10	22	16	7	10	542
−45°/1.4	182	68	45	18	19	722
45°/1.4	171	54	26	14	15	707
65°/2.3	5	12	5	5	6	468
75°/3.9	2	1	0	4	4	152
80°/5.8	3	0	0	4	2	110

Table 10.1: Absolute tilt invariance comparison. Summary of the results of the experiments that compare A-SIFT with SIFT, Harris-Affine (HarAff), Hessian-Affine (HesAff), MSER coded by the correlation descriptor (MSER+Corr) and by the SIFT descriptor (MSER+SIFT) for viewpoint angles between 45 and 80 degrees. Top: images taken with zoom ×1. Bottom: images taken with zoom ×10. (The camera-object distance is 10 times bigger.) The latitude angles and the absolute tilts are listed in the left column. With zoom ×1, the actual tilt varies on the same object varies around the marked value due to the perspective effect.

degrees.

The above experiments lead us to the following conclusion of the maximum absolute tilts of the approaches under comparison. SIFT exceeds hardly an absolute of 2 and the limit is about 2.5 for Harris-Affine and Hessian-Affine. The performance of MSER depends heavily on the types of image. For images with highly contrasted regions, MSER reaches an absolute tilt about 5. However if the images do not contain highly contrasted regions, the performance of MSER is very limited even under small tilts. For A-SIFT, an absolute tilt of 5.8 that corresponds to an extreme viewpoint angle of 80 degrees does not pose any difficulty to achieve robust recognition.

10.3.3 Transition Tilt Tests

The magazine shown in Fig. 10.4 is place face-up and photographed by a reflex camera and makes two sets of images. As illustrated in Fig. 10.3-b, for each image set, the camera with a fixed latitude angle θ, i.e., a fixed absolute tilt t of respectively 2 and 4, circles around with the longitude angle ϕ going from 0 to 90 degrees. The camera focus distance is ×4. In each set the images have

	SIFT	Haraff	Hesaff	MSER+Corr	MSER+SIFT	A-SIFT
50°/1.6	267	131	144	129	150	1692
60°/2.0	20	29	39	88	117	1012
70°/2.9	1	2	2	48	69	754
80°/5.8	0	0	0	10	17	267

Table 10.2: Absolute tilt invariance comparison. Summary of the results of the experiments that compare A-SIFT with SIFT, Harris-Affine (HarAff), Hessian-Affine (HesAff), MSER coded by the correlation descriptor (MSER+Corr) and by the SIFT descriptor (MSER+SIFT) for viewpoint angles between 50 and 80 degrees. The latitude angles and the absolute tilts are listed in the left column.

the same absolute tilt t while the transition tilt τ (with respect to the image taken at $\phi = 0$ degree) goes from 1 to t^2 when ϕ goes from 0 to 90 degrees. To evaluate the maximum transition tilt, the images taken at $\phi \neq 0$ are matched against the one taken at $\phi = 0$.

Table 10.3 compares the performance of the algorithms. The number of matches under the absolute tilt $t = 2$ shows clearly that performance of SIFT drops dramatically when the transition tilt goes from 1.3 to 1.7. With a transition tilt over 2.1, SIFT fails completely. Similarly a considerable performance decline is observed for Harris-Affine and Hessian-Affine when the transition tilt goes from 1.3 to 2.1. Hessian-Affine slightly outperform Harris-Affine but both methods fail completely when the transition tilt goes above 3. MSER and A-SIFT works stably until the transition tilt goes to 4. A-SIFT distinguishes itself by finding more than 10 times as many as those of MSER that cover a much larger area, as illustrated in Fig. 10.12.

Under an absolute tilt $t = 4$, SIFT, Harris-Affine and Hessian-Affine struggle at a transition tilt of 1.9 having comparable number of matches. They fail completely when the transition tilt goes bigger. MSER works stably until a transition tilt of 7.7. Over this value, the number of correspondences is too small for reliable recognition. A-SIFT works perfectly.

The transition tilt is a crucial notion that evaluates the degree of affine invariance of the image comparison algorithms. With the ordinary viewpoint changes the transition tilt goes easily to a big value (above 16 for example). The experiments above show that the maximum transition tilt, about 2 for SIFT and 2.5 for Harris-Affine and Hessian-Affine, is by far not enough. MSER enables reliable recognition until a transition tilt of about 10, under the condition that the images under comparison are free of scale change and contain highly contrasted regions. The limit of A-SIFT goes beyond 16 largely. Images that have undergone transition tilts up to 30 and more can be reliably recognized by A-SIFT, an example being illustrated in Fig. 8.7.

10.3.4 Comparative experiments

Fig. 10.18 compares the A-SIFT image matching with SIFT, Harris-Affine, Hessian-Affine and MSER. Table 10.4 summarizes the results. Fig. 10.13 shows images of a building facade taken from very different viewpoints. The transformation of the rectangle facade on the left to a trapezia on the right indicates

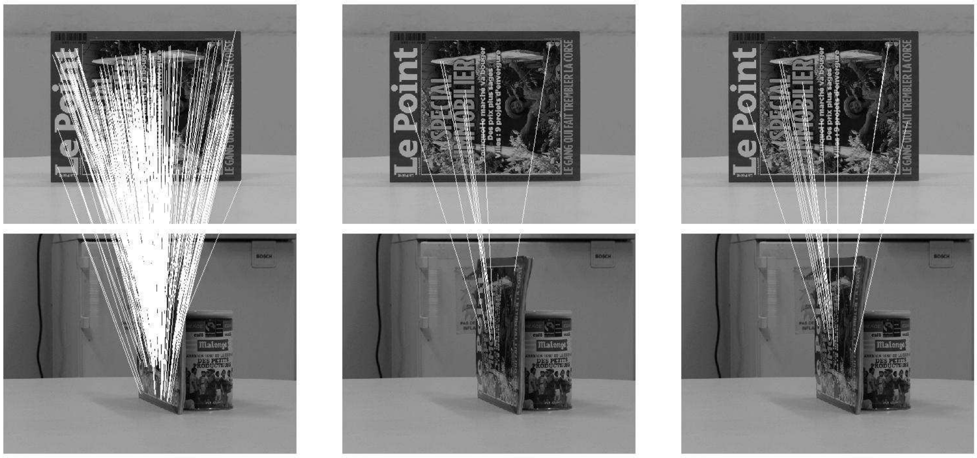

Figure 10.11: Correspondences between images taken with zoom ×4, frontal view and 80° angle, absolute tilt $t = 5.8$. A-SIFT (shown, $K = 3$), SIFT, Harris-Affine, Hessian-Affine, MSER+Corr (shown) and MSER+SIFT (shown) find respectively 349, 0, 0, 0, 10 and 17 correct matches.

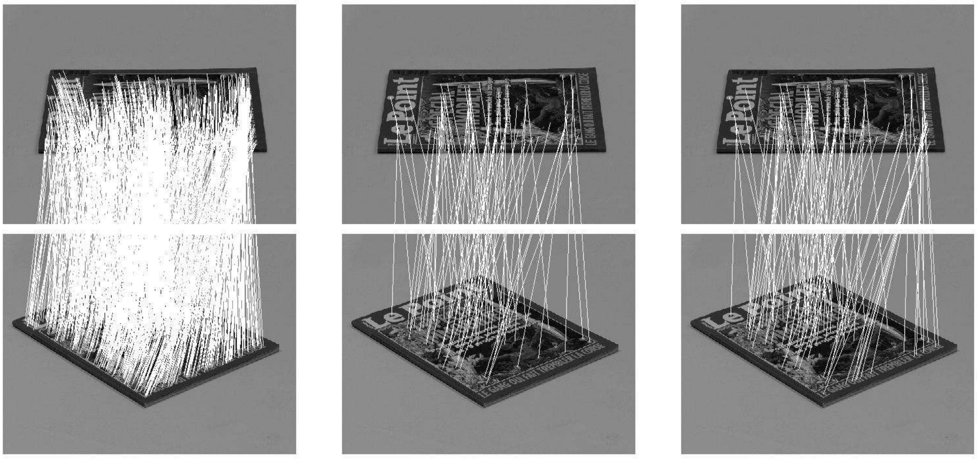

Figure 10.12: Correspondences with absolute tilts $t_1 = t_2 = 2$ and longitude angles $\phi_1 = 0°$ and $\phi_2 = 50°$, transition tilt $\tau = 3$. A-SIFT ($K = 3$, shown), SIFT, Harris-Affine, Hessian-Affine, MSER+Corr (shown) and MSER+SIFT (shown) find respectively 881, 2, 0, 2, 70 and 87 correct matches.

ϕ_2/τ $t = 2$	SIFT	Haraff	Hesaff	MSER+Corr	MSER+SIFT	A-SIFT
10°/1.3	408	233	176	110	124	1213
20°/1.7	49	75	84	96	122	1173
30°/2.1	5	24	32	85	103	1048
40°/2.5	3	13	29	71	88	809
50°/3.0	3	1	3	70	87	745
60°/3.4	2	0	1	50	62	744
70°/3.7	0	0	0	34	51	557
80°/3.9	0	0	0	40	51	589
90°/4.0	0	0	1	41	56	615

$\phi_2/\tau t = 4$	SIFT	Haraff	Hesaff	MSER+Corr	MSER+SIFT	A-SIFT
10°/1.9	22	32	14	38	49	1054
20°/3.3	4	5	1	32	39	842
30°/5.3	3	2	1	24	32	564
40°/7.7	0	0	0	22	28	351
50°/10.2	0	0	0	15	19	293
60°/12.4	1	0	0	12	17	145
70°/14.3	0	0	0	6	13	90
80°/15.6	0	0	0	6	12	106
90°/16.0	0	0	0	5	9	88

Table 10.3: Transition tilt invariance comparison. Summary of the results of the experiments that compare A-SIFT with SIFT, Harris-Affine (HarAff), Hessian-Affine (HesAff), MSER coded by the correlation descriptor (MSER+Corr) and by the SIFT descriptor (MSER+SIFT). The affine parameters of the two images are $\phi_1 = 0°$, $t_1 = t_2 = 2$ (above), $t_1 = t_2 = 4$ (below). ϕ_2 and the transition tilts τ are in the left column.

that the transformation is not affine, but strongly perspective. Nevertheless, since a projective transformation can be locally modeled by affine transforms, a large number of correspondences is established by A-SIFT. All the other methods fail. Fig. 10.14 shows the results of the standard test pair Graffiti 1 and Graffiti 6 proposed by Mikolajczyk [171]. A-SIFT finds 724 correspondences, out of which 3 are false. SIFT, Harris-Affine and Hessian-Affine find respectively 0, 3 and 1 correct correspondences: the $\tau = 3.2$ transition tilt is just a bit too large. MSER+Corr and MSER+SIFT find respectively 50 and 70 correct correspondences. Proposed by Matas et al. in their online demo [167] as a standard image to test MSER [168], the images in Fig. 10.15 show a number of containers placed on a desktop [1]. A-SIFT finds 194 correct correspondences. SIFT, Harris-Affine, Hessian-AFfine, MSER+Corr and MSER+SIFT find respectively 10, 23, 11, 16 and 22 correct correspondences. Let us note that images in Figs. 10.14 and 10.15 provide optimal conditions for MSER: the camera-object distances are similar and well contrasted shapes are present. But let us recall that MSER fails under large scale changes or when well contrasted shapes are not present. Fig. 10.16 contains two orthogonal road signs taken under a view change that makes a transition tilt $\tau \approx 2.6$. A-SIFT successfully matches the two signs

[1] We thank Michal Perdoch for having kindly provided us with the images.

finding 50 correspondences while all the other methods totally fail. The monument shown in Fig. 10.17 has undergone a viewpoint change of latitude angle 65 degrees (tilt t=2.4). A-SIFT works perfectly and finds 101 correct correspondences. SIFT struggles by establishing 13 correspondences. Harris-Affine, Hessian-Affine, MSER+Corr and MSER+SIFT fail finding only 2, 2, 5 and 4 matches. In Fig. 10.18 is the stump taking from different viewpoints which makes a transition tilt $\tau \approx 2.6$. A-SIFT achieved success finding 168 correct correspondences while all the other methods fail.

Fig. 10.19 illustrates a complex scene in a coffee room in which divers objects are presented. A-SIFT recognizes the scene by finding 125 correspondences over six non-coplanar objects. SIFT finds 11 correspondence on the wall and 2 on the box over the fridge. Other methods fail finding at most a few correspondences on the wall. Fig. 10.20 shows a coffee can that has been rotated for about 120 degrees. Over the cylinder transition tilt varies continuously and reaches big values. A-SIFT identifies 287 correspondences that cover almost all visible common areas. SIFT fails completely due to the too large viewpoint change. Harris-Affine and Hessian Affine fail by finding 3 and 6 correspondences. MSER finds a small number of correspondences that cover only part of letters that provide highly contrasted regions but it does not catch anything on the image part on the lower half of the can. The Palace of Versaille in Fig. 10.19 undertakes a viewpoint change of about 50 degrees. A-SIFT detect 67 matches uniformly distributed on the facade that can be viewed in the two images. SIFT finds 26 correspondences, mostly located in the closer end where the transition tilt is smaller due to the perspective effect. The other methods fail completely by finding zero or sporadic correspondences.

Fig./τ	SIFT	Haraff	Hesaff	MSER1	MSER2	A-SIFT
Fig. A/3.0	0	0	1	3	3	58
Fig. 10.13/3.8	0	0	1	0	2	68
Fig. 10.14/3.2	0	3	1	50	50	721
Fig. 10.15/[1.6,3.0]	10	23	11	16	22	254
Fig. 10.16/2.6	0	0	0	0	1	50
Fig. B/15.0	0	0	0	0	0	78
Fig. 10.17/2.4	13	2	2	5	4	101
Fig. 10.18/2.6	1	2	1	6	6	168
Fig. C/[1.6, ∞]	26	7	2	3	4	143
Fig. 10.19/[1.5, 3.3]	13	0	3	5	2	125
Fig. 10.20/[2.3, ∞]	0	6	3	12	22	287
Fig. D/[2, ∞]	19	5	7	7	13	123
Fig. 10.21/1.8	67	26	2	1	0	4

Table 10.4: Summary of the results of the experiments that compare A-SIFT with SIFT, Harris-Affine (HarAff), Hessian-Affine (HesAff), MSER coded by the correlation descriptor (MSER+Corr) and by the SIFT descriptor (MSER+SIFT). The transition tilts or their ranges are listed in the left column. The figures with Latin numbers are not shown in the text.

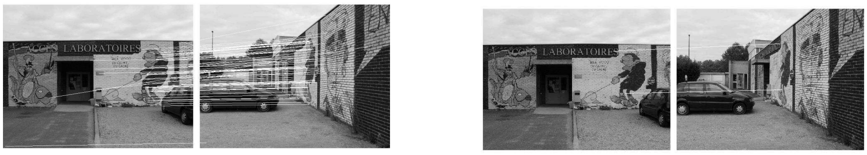

Figure 10.13: Image matching: Facade. Absolute (and transition) tilt $t = 3.8$ ($\theta = 75°$). A-SIFT, SIFT, Harris-Affine, Hessian-Affine, MSER+Corr and MSER+SIFT find respectively 68, 0, 1, 1, 0 and 2 correct matches. A-SIFT is implemented with $K = 2$, which means that $K = 3$ doesn't work Results shown: A-SIFT and MSER+SIFT

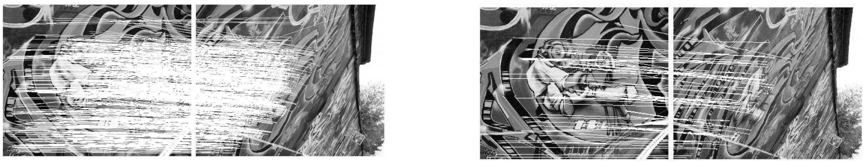

Figure 10.14: Image matching between Graffiti 1 and Graffiti 6. Transition tilt: $\tau \approx 3.2$. From top to bottom, left to right: A-SIFT, SIFT, Harris-Affine, Hessian-Affine, MSER+Corr and MSER+SIFT find respectively 721, 0, 3, 1, 50 and 70 correct matches. A-SIFT is implemented with $K = 3$. Results shown: A-SIFT and MSER+SIFT

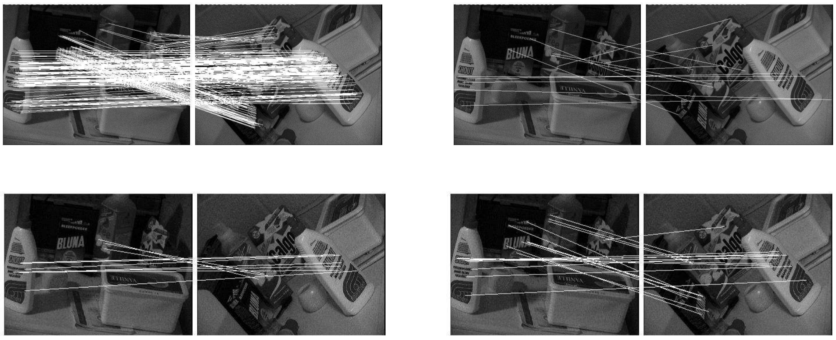

Figure 10.15: Image matching (images proposed by Matas et al [167]). Transition tilt: $\tau \in [1.6, 3.0]$. From top to bottom, left to right: A-SIFT (shown), SIFT (shown), Harris-Affine, Hessian-Affine (shown), MSER+Corr and MSER+SIFT (shown) find respectively 254, 10, 23, 11, 16 and 22 correct matches. A-SIFT is implemented with $K = 3$.

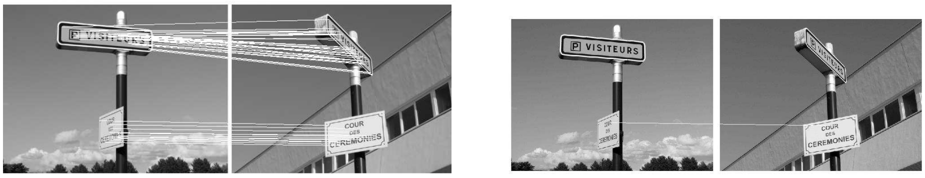

Figure 10.16: Image matching: road signs. Transition tilt $\tau \approx 2.6$. A-SIFT (shown), SIFT, Harris-Affine, Hessian-Affine, MSER+Corr and MSER+SIFT (shown) find respectively 50, 0, 0, 0, 0 and 1 correct matches. A-SIFT is implemented with $K = 3$.

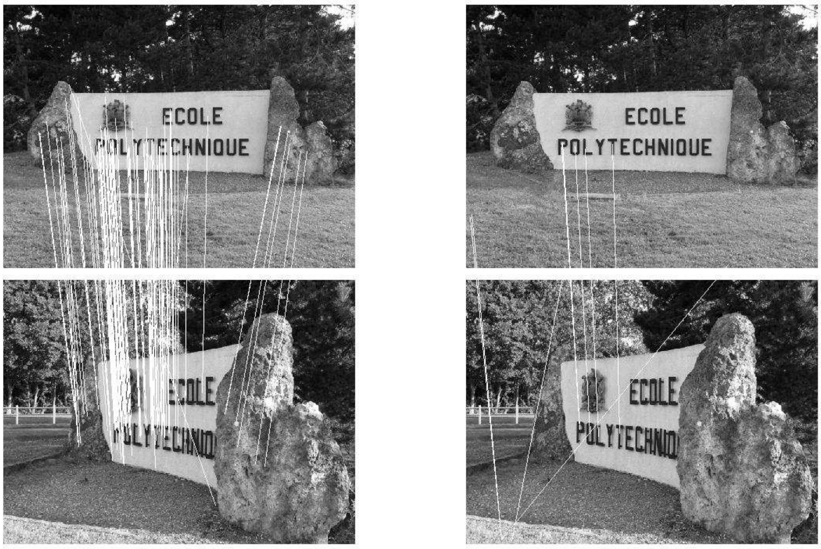

Figure 10.17: Image matching: Ecole Polytechnique. Absolute (and transition) tilt $t \approx 2.4$ ($\theta = 65°$). A-SIFT (shown), SIFT, Harris-Affine, Hessian-Affine, MSER+Corr and MSER+SIFT (shown) find respectively 101, 13, 2, 2, 5 and 4 correct matches. A-SIFT is implemented with $K = 3$.

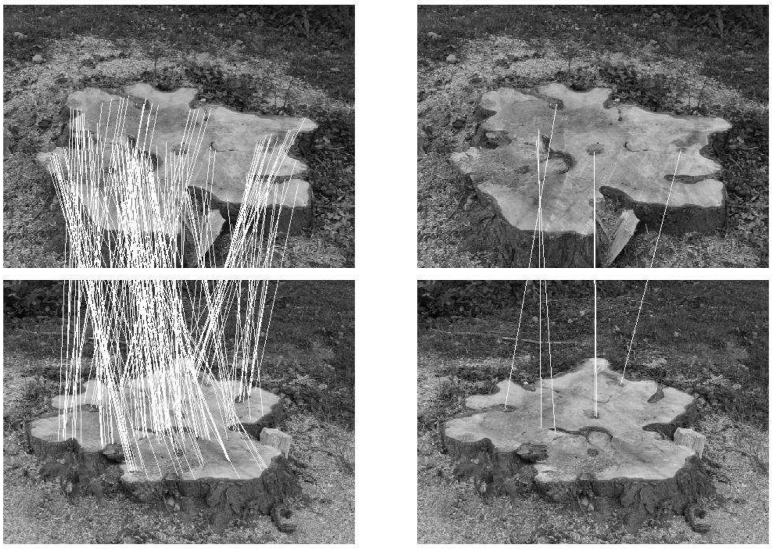

Figure 10.18: Image matching: stump. Transition tilt $\tau \approx 2.6$. A-SIFT (shown), SIFT, Harris-Affine, Hessian-Affine, MSER+Corr and MSER+SIFT (shown) find respectively 168, 1, 2, 1, 6 and 6 correct matches. A-SIFT is implemented with $K = 3$.

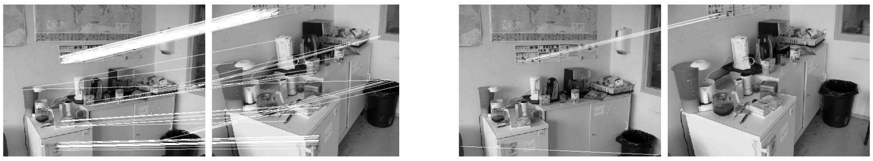

Figure 10.19: Image matching: coffee room. Transition tilt $\tau \in [1.5, 3.3]$. A-SIFT (shown), SIFT, Harris-Affine, Hessian-Affine, MSER+Corr and MSER+SIFT (shown) find respectively 125, 13, 0, 3, 5 and 2 correct matches. A-SIFT is implemented with $K = 1$, which gives one of the very few examples where details in one of the images are so small that $K = 2$ and $K = 3$ do not work.

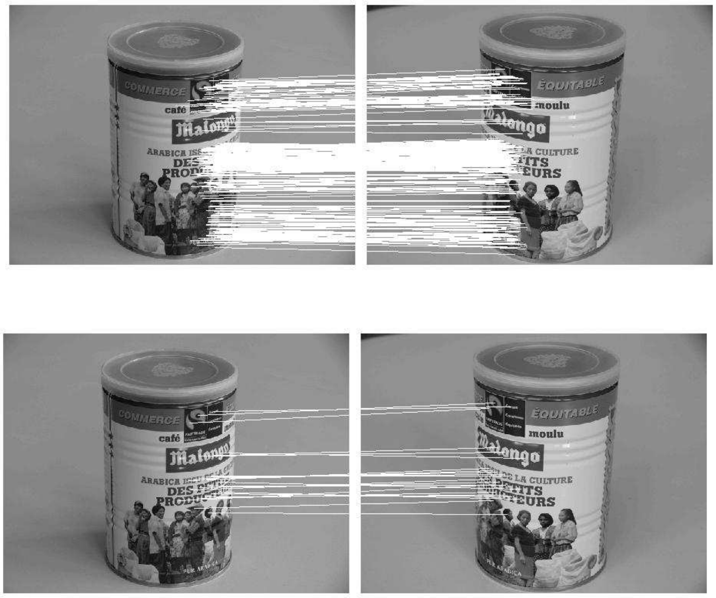

Figure 10.20: Image matching: can. Transition tilt $\tau \in [2.3, \infty]$. A-SIFT (shown), SIFT, Harris-Affine, Hessian-Affine, MSER+Corr and MSER+SIFT (shown) find respectively 287, 0, 6, 3, 12 and 22 correct matches. A-SIFT is implemented with $K = 2$, which means that $K = 3$ doesn't work

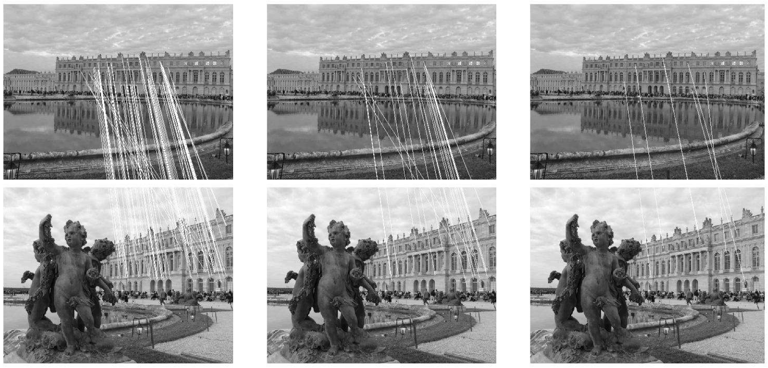

Figure 10.21: Image matching: Palace of Versailles. Transition tilt $\tau = 1.8$. A-SIFT (shown), SIFT (shown), Harris-Affine, Hessian-Affine, MSER+Corr and MSER+SIFT (shown) find respectively 67, 26, 2, 1, 0 and 4 correct matches. A-SIFT is implemented with $K = 3$.

10.3.5 Symmetry detection in perspective

Symmetry detection has drawn considerable attention in computer vision and has been used for numerous applications such as image indexing, completion of occluded shapes, object detection, facial image analysis and visual attention (see, for example, [56] for a survey). The image projection is usually approximated by plane affine transforms for symmetry detection in perspective [185]. Some recent works apply SIFT, MSER and other affine-invariant detectors and descriptors to detect bilateral symmetry [152, 56]. Conversely, symmetry has been used to extract affine-invariant image features [15].

The image matching algorithm can be used to detect bilateral symmetry in an image $\mathbf{u}$, by simply looking for correspondences between $\mathbf{u}(x, y)$ and its flipped version $\mathbf{u}(-x, y)$. After being flipped, symmetric structures become either identical if taken in frontal view, or identical up to an oblique view otherwise. A correspondence between $\mathbf{u}(x, y)$ and $\mathbf{u}(-x, y)$ therefore connects a pair of bilateral symmetrical points in $\mathbf{u}(x, y)$. Fig. 10.22 shows some examples of bilateral symmetry detection obtained by A-SIFT, SIFT, Harris-Affine, Hessian-Affine, MSER+Corr and MSER+SIFT. A-SIFT that has the best performance on affine invariant image matching, results in the best symmetry detection in perspective.

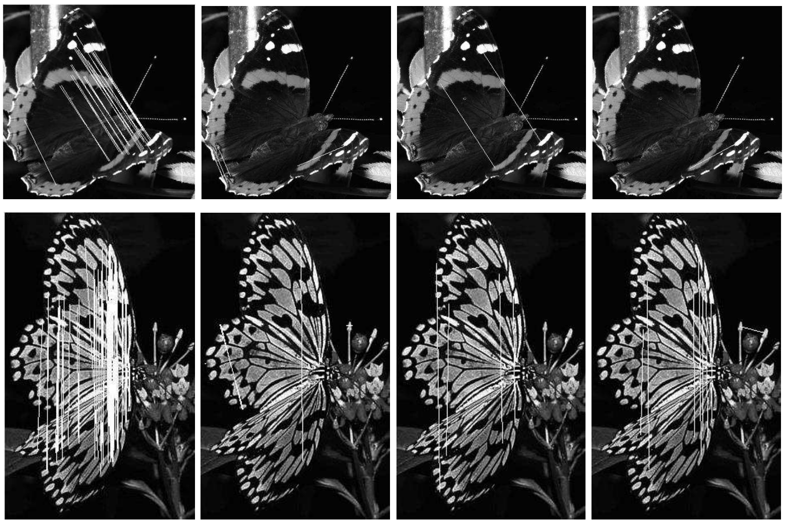

Figure 10.22: Symmetry detection in perspective. From left to right: detection results by A-SIFT, SIFT, Hessian-Affine, MSER+SIFT.

10.4 Comments and references

David Pritchard's master thesis was a first step toward A-SIFT. Quoting [205] in his 2003 master thesis on cloth parameters and motion capture:

> Cloth strongly resists stretching, but permits substantial bending; folds and wrinkles are a distinctive characteristic of cloth. This

> behaviour means that sections of the cloth are often seen at oblique angles, leading to large affine distortions of features in certain regions of the cloth. Unfortunately, SIFT features are not invariant to large affine distortions.(...) To compensate for this, we use an expanded set of reference features. We generate a new reference image by using a 2 x 2 transformation matrix T to scale the reference image by half horizontally. We repeat three more times, scaling vertically and along axes at 45 degrees, as shown in Figure 5.3. This simulates different oblique views of the reference image. For each of these scaled oblique views, we collect a set of SIFT features. Finally, these new SIFT features are merged into the reference feature set. When performing this merge, we must adjust feature positions, scales and orientations by using T-1. This approach is compatible with the recommendations made by Lowe for correcting SIFT's sensitivity to affine change.

In recent years local image detectors have bloomed. They can be classified by their incremental invariance properties. All of them are translation invariant. The Harris point detector [105] is also rotation invariant. The Harris-Laplace, Hessian-Laplace and the DoG (Difference-of-Gaussian) region detectors [172, 175, 151, 80] are invariant to rotations and changes of scale. Some moment-based region detectors [146, 25] including the Harris-Affine and Hessian-Affine region detectors [173, 175], an edge-based region detector [246, 245], an intensity-based region detector [244, 245], an entropy-based region detector [124], and two level line-based region detectors MSER ("maximally stable extremal region") [168] and LLD ("level line descriptor") [187, 189, 191] are designed to be invariant to affine transformations. MSER, in particular, has been demonstrated to have often better performance than other affine invariant detectors, followed by Hessian-Affine and Harris-Affine [177].

The mentioned methods have a varying complexity. Measured in terms of their processing times, the fastest is MSER, followed by Harris-Affine and Hessian-Affine. SIFT is ten times slower, but a recent acceleration has been proposed in [97], that equals it to the other detectors.

However, the mentioned affine invariant detectors aren't fully affine invariant. As pointed out in [151], they start with initial feature scales and locations selected in a non-affine invariant manner. As shown in [191], MSER and LLD are not fully scale invariant, because they do not take into account the drastic changes of level lines due to blur. This is also the case for other image local descriptors, such as the distribution-based shape context [27], the geometric histogram [18] descriptors, the derivative-based complex filters [25, 224], and the moment invariants [250].

There is, however, at least one method dealing with scale (and therefore with blur) in a fully satisfactory way. In his milestone paper [151], Lowe has proposed a scale-invariant feature transform (SIFT) descriptor that is invariant to image scaling and rotation and partially invariant to illumination and viewpoint changes. Although SIFT is *a priori* less invariant to affine transforms than other descriptors such as Hessian-Affine and Harris-Affine [172, 175], its performance turns out to be comparable, as we shall see in many experiments. Furthermore, SIFT is really scale invariant (a mathematical proof of this fact is given in [184]). A number of SIFT variants and extensions, including PCA-SIFT [126], GLOH (gradient location-orientation histogram) [176] and SURF (speeded up robust features) [26], that claim to have improved robustness and distinctiveness with scaled-down complexity have been developed ever since [85, 143]. Demonstrated to be superior to many other descriptors [174, 176], SIFT and its variants have been popularly applied for scene recognition [79, 182, 217, 251, 96, 226, 265, 183] and detection [86, 194], robot localization [28, 227, 196, 120], image registration [264], image retrieval [104], motion tracking [247, 128], 3D modeling and reconstruction [211, 252], building panoramas [3, 34], photo management [263, 141, 235, 51], as well as symmetry detection [152].

Part II

Contrast-Invariant Image Analysis

Chapter 11

Contrast-Invariant Classes of Functions and Their Level Sets

This chapter is about one of the major technological contributions of mathematical morphology, namely the representation of images by their upper level sets. As we shall see in this chapter, this leads to a handy contrast invariant representation of images.

Definition 11.1. *Let $u \in \mathcal{F}$. The level set of u at level $0 \leq \lambda \leq 1$ is denoted by $\mathcal{X}_\lambda u$ and defined by*

$$\mathcal{X}_\lambda u = \{\mathbf{x} \mid u(\mathbf{x}) \geq \lambda\}.$$

Strictly speaking, we have called level sets what should more properly be called upper level sets. Several level sets of a digital image are shown in Figure 11.1 and all of the level sets of a synthetic image are illustrated in Figure 11.2. The reconstruction of an image from its level sets is illustrated in Figure 11.3. Two important properties of the level sets of a function follow directly from the definition. The first is that the level sets provide a complete description of the function. Indeed, we can reconstruct u from its level sets $\mathcal{X}_\lambda u$ by the formula

$$u(\mathbf{x}) = \sup\{\lambda \mid \mathbf{x} \in \mathcal{X}_\lambda u\}.$$

This formula is called *superposition principle* as u is being reconstructed by "superposing" its level sets.

Exercise 11.1. Prove the superposition principle. ■

The second important property is that level sets of a function are globally invariant under contrast changes. We say that two functions u and v have the same level sets globally if for every λ there is μ such that $\mathcal{X}_\mu v = \mathcal{X}_\lambda u$, and conversely. Now suppose that a contrast change $g : \mathbb{R} \to \mathbb{R}$ is continuous and increasing. Then it is not difficult to show that $v = g(u)$ and u have the same level sets globally.

Exercise 11.2. Check this last statement for any function u and any continuous increasing contrast change g. ■

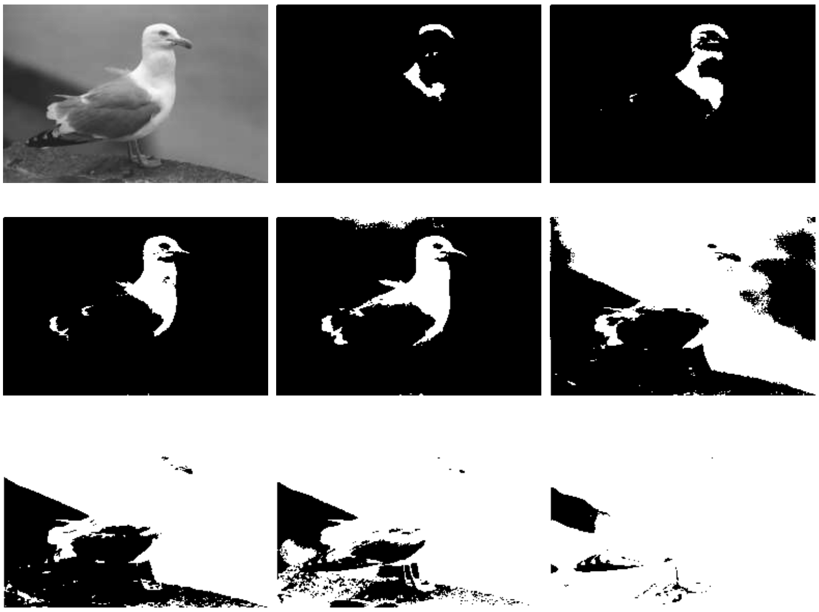

Figure 11.1: Level sets of a digital image. Left to right, top to bottom: We first show an image with range of gray levels from 0 to 255. Then we show eight level sets in decreasing order from $\lambda = 225$ to $\lambda = 50$, where the grayscale step is 25. Notice how essential features of the shapes are contained in the boundaries of level sets, the level lines. Each level set (which appears as white) is contained in the next one, as guaranteed by Proposition 11.2.

Conversely, we shall prove that if the level sets of a function $v \in \mathcal{F}$ are level sets of u, then there is a continuous contrast change g such that $v = g(u)$. This justifies the attention we will dedicate to level sets, as they turn out to contain all of the contrast invariant information about u.

11.1 From an image to its level sets and back

In the next proposition, for a sake of generality, we consider bounded measurable functions on S_N, not just functions in $\mathcal{F}$.

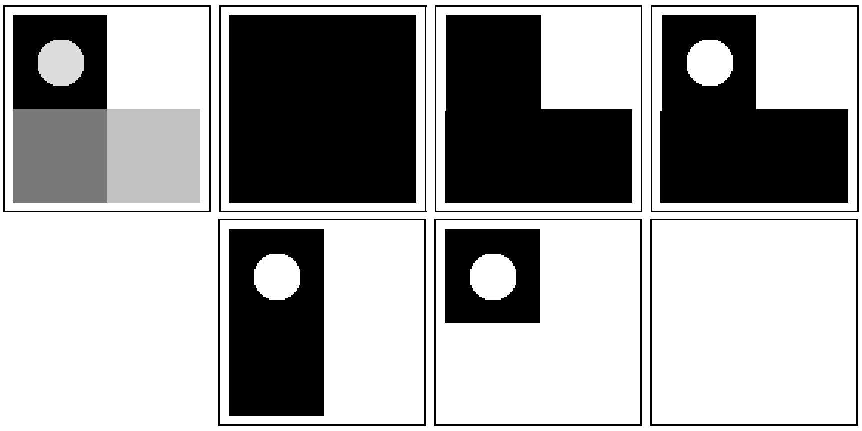

Figure 11.2: A simple synthetic image and all of its level sets (in white) with decreasing levels, from left to right and from top to bottom.

Proposition 11.2. *Let X_λ denote the level sets $\mathcal{X}_\lambda u$ of a bounded measurable function $u : S_N \to \mathbb{R}$. Then the sets X_λ satisfy the following two structural properties:*

(i) *If $\lambda > \mu$, then $X_\lambda \subset X_\mu$. In addition, there are two real numbers $\lambda_{max} \geq \lambda_{min}$ so that $X_\lambda = S_N$ for $\lambda < \lambda_{min}$, $X_\lambda = \emptyset$ for $\lambda > \lambda_{max}$.*

(ii) *$X_\lambda = \bigcap_{\mu<\lambda} X_\mu$ for every $\lambda \in \mathbb{R}$.*

Conversely, if $(X_\lambda)_{\lambda\in\mathbb{R}}$ is a family of sets of $\mathcal{M}$ that satisfies (i) and (ii), then the level sets of the function u defined by superposition principle,

$$u(\mathbf{x}) = \sup\{\lambda \mid \mathbf{x} \in X_\lambda\} \tag{11.1}$$

satisfy $\mathcal{X}_\lambda u = X_\lambda$ for all $\lambda \in \mathbb{R}$ and $\lambda_{min} \leq u \leq \lambda_{max}$.

Proof. The first part of Relation (i) follows directly from the definition of upper level sets. The second part of (i) works with $\lambda_{min} = \inf u$ and $\lambda_{max} = \sup u$. The relation (ii) follows from the equivalence $u(\mathbf{x}) \geq \lambda \Leftrightarrow u(\mathbf{x}) \geq \mu$ for every $\mu < \lambda$.

Conversely, take a family of subsets $(X_\lambda)_{\lambda\in\mathbb{R}}$ satisfying (i) and (ii) and define u by the superposition principle. Let us show that $X_\lambda = \mathcal{X}_\lambda u$. Take first $\mathbf{x} \in X_\lambda$. Then it follows from the definition of u that $u(\mathbf{x}) \geq \lambda$, and hence $\mathbf{x} \in \mathcal{X}_\lambda u$. Thus, $X_\lambda \subset \mathcal{X}_\lambda u$. Conversely, let $\mathbf{x} \in \mathcal{X}_\lambda u$. Then $u(\mathbf{x}) = \sup\{\nu \mid \mathbf{x} \in X_\nu\} \geq \lambda$. Consider any $\mu < \lambda$. Then there exists a μ' such that $\mu < \mu' \leq \sup\{\nu \mid \mathbf{x} \in X_\nu\}$ and $\mathbf{x} \in X_{\mu'}$. It follows from (i) that $\mathbf{x} \in X_\mu$. Since μ was any number less that λ, we conclude by using (ii) that $\mathbf{x} \in \bigcap_{\mu<\lambda} X_\mu = X_\lambda$. It is easily checked that $\lambda_{min} \leq u \leq \lambda_{max}$. □

Exercise 11.3. Check the last statement of the preceding proof, that $\lambda_{min} \leq u \leq \lambda_{max}$. ■

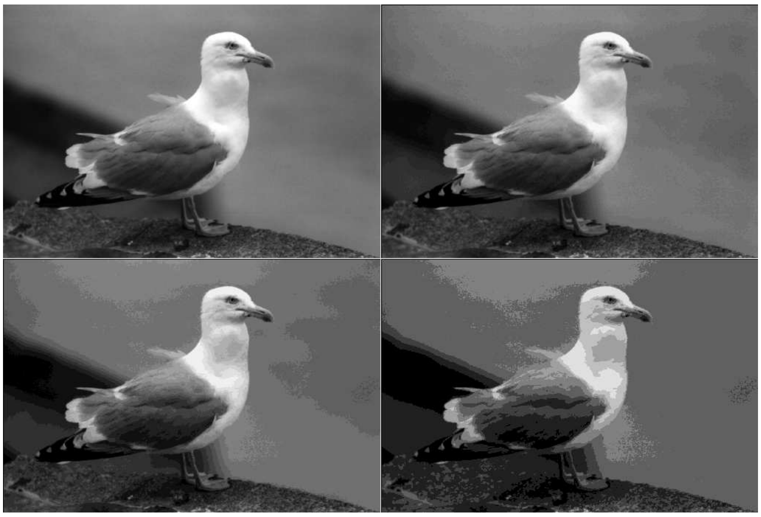

Figure 11.3: Reconstruction of an image from its level sets: an illustration of Proposition 3.2. We use four different subsets of the image's level sets to give four reconstructions. Top, left: all level sets; top, right: all level sets whose gray level is a multiple of 8; bottom, left: multiples of 16; bottom, right: multiples of 32. Notice the relative stability of the image shape content under these drastic quantizations of the gray levels.

11.2 Contrast changes and level sets

Practical aspects of contrast changes are illustrated in Figures 11.4, 11.5, 11.6, and 11.7, which illustrate how insensitive our perception of images is to contrast changes, even when they are flat on some interval. When this happens, some information on the image is even lost, as several grey levels melt together.

Definition 11.3. *Any nondecreasing continuous surjection $g:\ \mathbb{R} \to \mathbb{R}$ will be called a contrast change.*

Exercise 11.4. Remark that $g(s) \to \pm\infty$ as $s \to \pm\infty$. Check that if $u \in \mathcal{F}$ and g is a contrast change, then $g(u) \in \mathcal{F}$. ■

In case g is increasing, g has an inverse contrast change g^{-1}. In case g is flat on some interval, we shall be happy with a pseudo-inverse for g.

Definition 11.4. *The pseudo-inverse of any contrast change $g : \mathbb{R} \to \mathbb{R}$ is defined by*

$$g^{(-1)}(\lambda) = \inf\{r \in \mathbb{R} \ \mid g(r) \geq \lambda\}.$$

Exercise 11.5. Check that g^{-1} is finite on $\mathbb{R}$ and tends to $\pm\infty$ as $s \to \pm\infty$. Give an example of g such that g^{-1} is not continuous. ■

Exercise 11.6. Compute and draw $g^{(-1)}$ for the function $g(s) = \max(0, s)$. Notice that such a function is ruled out by our conditions at infinity for contrast changes. ■

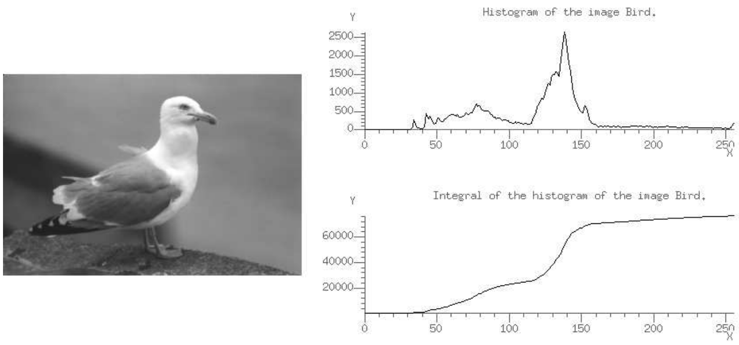

Figure 11.4: The histogram of the image Bird. For each $i \in \{0, 1, \dots, 255\}$, we display (above, right) the function $h(i) = \text{Card}\,\{\mathbf{x} \mid u(\mathbf{x}) = i\}$. The function below is the cumulative histogram, namely the primitive of h defined by $H_u(i) = \text{Card}\,\{\mathbf{x} \mid u(\mathbf{x}) \leq i\}$. The shape of h provides an indication about the overall contrast of the image and about the contrast change imposed by the sensors. See Chap. 12 for manipulations of the cumulative histogram.

Lemma 11.5. *Let $g : \mathbb{R} \to \mathbb{R}$ be a contrast change. Then for every $\lambda \in \mathbb{R}$, $g(g^{(-1)})(\lambda) = \lambda$ and*

$$g(s) \geq \lambda \textit{ if and only if } s \geq g^{(-1)}(\lambda). \tag{11.2}$$

Proof. The first relation follows immediately from the continuity of g. If $g(s) \geq \lambda$, then $s \geq g^{(-1)}(\lambda)$ by the definition of $g^{(-1)}(\lambda)$. Conversely, if $s \geq g^{(-1)}(\lambda)$, then $g(s) \geq g(g^{(-1)}(\lambda)) = \lambda$ and thus $g(s) \geq \lambda$. □

Theorem 11.6. *Let $u \in \mathcal{F}$ and g be a contrast change. Then any level set of $g(u)$ is a level set of u. More precisely, for $\lambda \in \mathbb{R}$,*

$$\mathcal{X}_\lambda g(u) = \mathcal{X}_{g^{(-1)}(\lambda)} u. \tag{11.3}$$

Proof. The proof is read directly from Lemma 11.5 by taking $s = u$. □

The next result is a converse statement to Theorem 11.6.

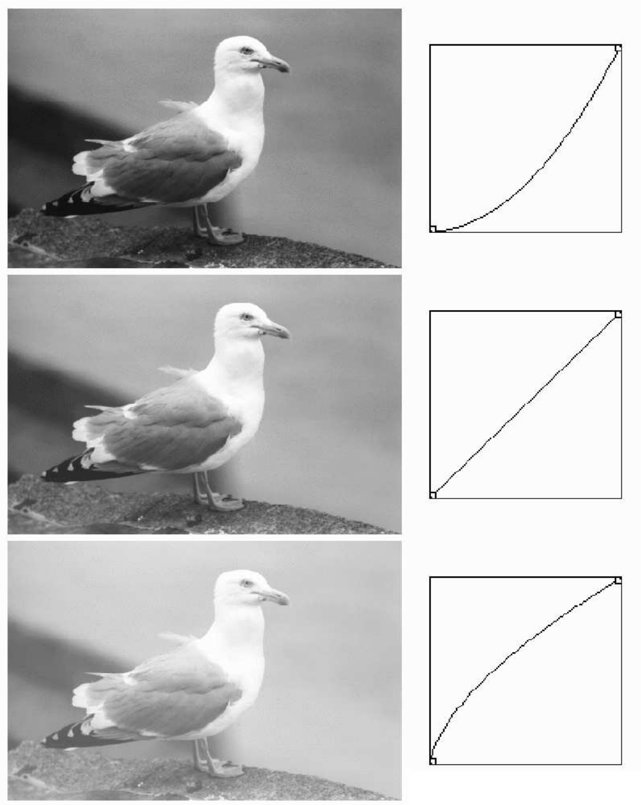

Figure 11.5: Contrast changes and an equivalence class of images. The three images have exactly the same level sets and level lines, but their level sets are mapped onto three different gray-level scales. The graphs on the right are the graphs of the contrast changes $u \mapsto g(u)$ that have been applied to the initial gray levels. The first one is concave; it enhances the darker parts of the image. The second one is the identity; it leaves the image unaltered. The third one is convex; it enhances the brighter parts of the image. Software allows one to manipulate the contrast of an image to obtain the best visualization. From the image analysis viewpoint, image data should be considered as an equivalence class under all possible contrast changes.

Theorem 11.7. *Let u and $v \in \mathcal{F}$ such that every level set of v is a level set of u. Then $v = g(u)$ for some contrast change g.*

Proof. One can actually give an explicit formula for g, namely, for every $\mu \in u(S_N)$,

$$g(\mu) = \sup\{\lambda \in v(S_N) \mid \mathcal{X}_\mu u \subset \mathcal{X}_\lambda v\}. \tag{11.4}$$

For $\mu \notin u(S_N)$, we can easily extend g into an nondecreasing function such that $g(\pm\infty) = \pm\infty$). (Take (e.g.) g piecewise affine). Note that $\nu > \mu$ implies that $g(\nu) \geq g(\mu)$. Let us first show that $\inf v \leq g(\mu) \leq \sup v$. Set

$$\Lambda := \{\lambda \mid \mathcal{X}_\mu u \subset \mathcal{X}_\lambda v\}.$$

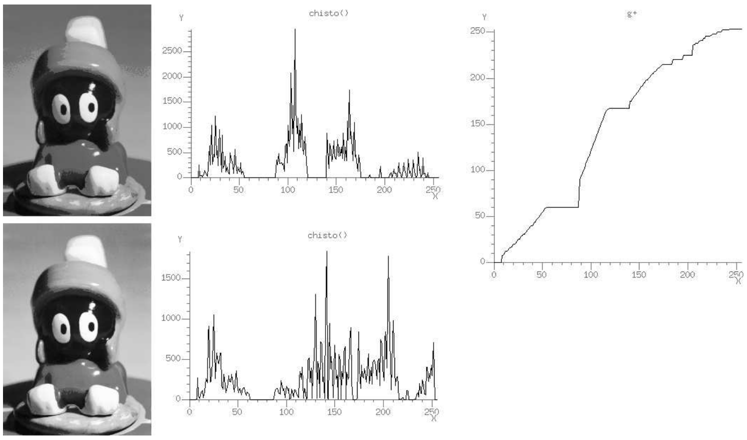

Figure 11.6: The two images (left) have the same set of level sets. The contrast change that maps the upper image onto the lower image is displayed on the right. It corresponds to one of the possible g functions whose existence is stated in Corollary 3.14. The function g may be locally constant on intervals where the histogram of the upper image is zero (see top, middle graph). Indeed, on such intervals, the level sets are invariant.

Λ is not empty because $\mathcal{X}_{\inf v} = S_N$ and therefore $\inf v \in \Lambda$. Thus $g(\mu) = \sup \Lambda \geq \inf v$. On the other hand $\mathcal{X}_{\sup v+\varepsilon} v = \emptyset$ for every $\varepsilon > 0$. Since $\mu \in u(S_N)$, $\mathcal{X}_\mu u \neq \emptyset$ and therefore $g(\mu) = \sup \Lambda \leq \sup v$.

Step 1: Proof that $\boldsymbol{v(\mathbf{x}) \geq g(u(\mathbf{x}))}$. By Proposition 11.2($i$) Λ has the form $(-\infty, \sup \Lambda)$ or $(-\infty, \sup \Lambda]$. But by Proposition 11.2(ii), $\mathcal{X}_{\sup \Lambda} v = \bigcap_{\lambda < \sup \Lambda} \mathcal{X}_\lambda v$, and this implies by the definition of Λ that $g(\mu) = \sup \Lambda \in \Lambda$. Thus,

$$\mathcal{X}_\mu u \subset \mathcal{X}_{g(\mu)} v. \tag{11.5}$$

Given $\mathbf{x} \in S_N$, let $\mu = u(\mathbf{x})$ in (11.5). Then,

$$\mathcal{X}_{u(\mathbf{x})} u \subset \mathcal{X}_{g(u(\mathbf{x}))} v.$$

Since $\mathbf{x} \in \mathcal{X}_{u(\mathbf{x})} u$, we conclude that $\mathbf{x} \in \mathcal{X}_{g(u(\mathbf{x}))} v = \{\mathbf{y} \mid v(\mathbf{y}) \geq g(u(\mathbf{x}))\}$.

Step 2: Proof that $\boldsymbol{v(\mathbf{x}) \leq g(u(\mathbf{x}))}$. Given $\mathbf{x} \in S_N$, we translate the assumption with $\lambda = v(\mathbf{x})$ as follows: There exists a $\mu(\mathbf{x}) \in \mathbb{R}$ such that

$$\mathcal{X}_{v(\mathbf{x})} v = \{\mathbf{y} \mid u(\mathbf{y}) \geq \mu(\mathbf{x})\} = \mathcal{X}_{\mu(\mathbf{x})} u. \tag{11.6}$$

Since $\mathbf{x} \in \mathcal{X}_{v(\mathbf{x})} v$, we know that $\mathbf{x} \in \mathcal{X}_{\mu(\mathbf{x})} u$. Thus, $u(\mathbf{x}) \geq \mu(\mathbf{x})$, and $\mathcal{X}_{u(\mathbf{x})} u \subset \mathcal{X}_{\mu(\mathbf{x})} u = \mathcal{X}_{v(\mathbf{x})} v$. This last relation implies by the definition of g that $v(\mathbf{x}) \leq g(u(\mathbf{x}))$.

Step 3: Proof that $\boldsymbol{g}$ is continuous. Recall that the image of a connected set by a continuous function is connected. Thus $u(S_N)$ is an interval of $\mathbb{R}$ and so is $v(S_N)$. Since $g(u) = v$, $g(u(S_N)) = v(S_N)$ is an interval. Now, a nondecreasing function is continuous on an interval if and only if its range is connected. Thus g is continuous on $u(S_N)$ and so is its extension to $\mathbb{R}$. □

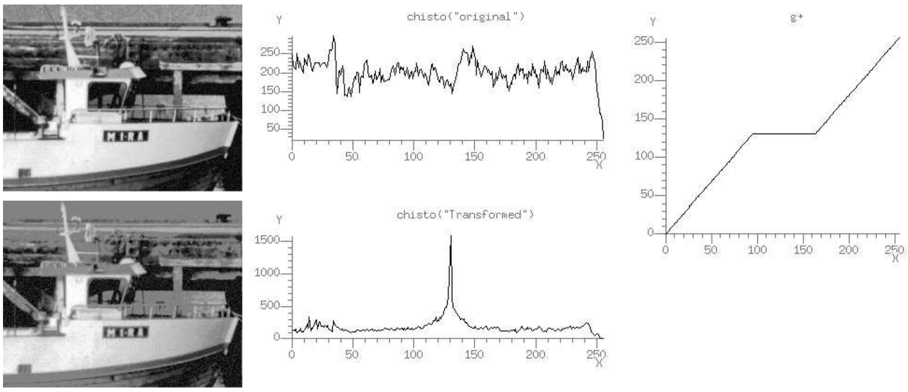

Figure 11.7: The original image (top, left) has a strictly positive histogram (all gray levels between 0 and 255 are represented). Therefore, if any contrast change g that is not strictly increasing is applied, then some data will be lost. Every level set of the transformed image $g(u)$ is a level set of the original image; however, the original image has more level sets than the transformed image.

Exercise 11.7. Prove the last statement in the theorem, namely that "a nondecreasing function is continuous on an interval if and only if its range is connected". ■

Exercise 11.8. By reading carefully the steps 1 and 2 of the proof of Theorem 11.7, check that this theorem applies with u and v just bounded and measurable on S_N. Then one has still has $v = g(u)$ with g defined in the same way. Of course g is still nondecreasing but not necessarily continuous. Find a simple example of functions u and v such that g is not continuous. ■

11.3 Exercises

Exercise 11.9. This exercise gives a way to compute the function g such that $v = g(u)$ defined in the proof of Theorem 11.7 in terms of the repartition functions of u and v. Let G be a Gauss function defined on $\mathbb{R}^N$ such that $\int_{\mathbb{R}^N} G(\mathbf{x})d\mathbf{x} = 1$. For every measurable subset of $\mathbb{R}^N$, set $|A|_G := \int_A G(\mathbf{x})d\mathbf{x}$. Let u be a bounded continuous function on $\mathbb{R}^N$. We can associate with u its *repartition function* $h_u(\lambda) := |\mathcal{X}_\lambda u|_G$. Show that $h_u : \lambda \in [\inf u, \sup u] \to h_u(\lambda)$ is strictly decreasing. Show that it can have jumps but is left-continuous, that is $h_u(\lambda) = \lim_{\mu\uparrow\lambda} h_u(\mu)$. Define for every non increasing function h a pseudo inverse by $h^{((-1))}(\mu) := \sup\{\lambda \mid h(\lambda) \geq \mu\}$. Show that $h^{((-1))}$ is non increasing and that $h^{((-1))} \circ h(\mu) \geq \mu$, and that if h is left-continuous, $h \circ h^{((-1))}(\mu) \geq \mu$. Using (11.4) prove that $g = h_v^{((-1))} \circ h_u$.
Hint: prove that $g(\mu) = \sup\{\lambda \mid |\mathcal{X}_\mu u|_G \leq |\mathcal{X}_\lambda v|_G\}$. ■

Exercise 11.10. Let u be a real-valued function. If $(\mu_n)_{n\in\mathbb{N}}$ is an increasing sequence that tends to λ, prove that

$$\mathcal{X}_\lambda u = \bigcap_{n\in\mathbb{N}} \mathcal{X}_{\mu_n} u \tag{11.7}$$

$$\{\mathbf{x} \mid u(\mathbf{x}) > \lambda\} = \bigcup_{\mu>\lambda} \mathcal{X}_\mu u. \tag{11.8}$$

■

11.4 Comments and references

Contrast invariance and level sets. It was Wertheimer who noticed that the actual local values of the gray levels in an image could not be relevant information for the human visual system [260]. Contrast invariance is one of the fundamental model assumptions in mathematical morphology. The two basic books on this subject are Matheron [169] and Serra [228, 230]. See also the fundamental paper by Serra [229]. Ballester et al. defined an “image intersection” whose principle is to keep all pieces of bilevel sets common to two images [22]. (A bilevel set is of the form $\{\mathbf{x} \mid \lambda \le u(\mathbf{x}) \le \mu\}$.) Monasse and Guichard developed a *fast level set transform* (FLST) to associate with every image the inclusion tree of connected components of level sets [181]. They show that the inclusion trees of connected upper and lower level sets can be fused into a single inclusion tree; among other applications, this tree can be used for image registration. See Monasse [180].

Contrast changes. The ability to vary the contrast (to apply a contrast change) of a digital image is a very useful tool for improving image visualization. Professional image processing software has this capability, and it is also found in popular software for manipulating digital images. For more about contrast changes that preserve level sets, see [48]. Many reference on contrast-invariant operators are given at the end of Chapter 13.

Chapter 12

Specifying the contrast of images

Midway image equalization means any method giving to a pair of images a similar histogram, while maintaining as much as possible their previous grey level dynamics. The comparison of two images is one of the main goals of computer vision. The pair can be a stereo pair, two images of the same object (a painting for example), multi-channel images of the same region, images of a movie, etc. Image comparison is perceptually greatly improved if both images have the same grey level dynamics (which means, the same grey level histogram). Many image comparison algorithms are based on grey level and take as basic assumption that intensities of corresponding points in both images are equal. However, this assumption is generally false for stereo pairs, and deviations from this assumption cannot even be modeled by affine transforms [58]. Consequently, if we want to compare visually and numerically two images, it is useful to give them first the same dynamic range and luminance.

In all of this applicative chapter the images $u(\mathbf{x})$ and $v(\mathbf{x})$ are defined on a domain which is the union of M pixels. The area of each pixel is equal to 1. The images are discrete in space and values: they attain values in a finite set $\mathbb{L}$ and they are constant on each pixel of the domain. We shall call such images *discrete images*. The piecewise constant interpolation is a very bad image interpolation. It is only used here for a fast handling of image histograms. For other scopes, better interpolation methods are of course necessary.

Definition 12.1. *Let u be a discrete image. We call cumulative histogram of u the function $H_u : \mathbb{L} \to \mathbb{M} := [0, M] \cap \mathbb{N}$ defined by*

$$H_u(l) =: \mathrm{meas}(\{\mathbf{x} \mid u(\mathbf{x}) \leq l\}).$$

This cumulative histogram is a primitive of the *histogram* of the image $h(l) = \mathrm{meas}(\{\mathbf{x} \mid u(\mathbf{x}) = l\})$. Figures 11.4, 11.6 and the first line of Figure 12.1. show the histograms of some images and their cumulative histograms. In fact Figure 11.7 shows first the histogram and then the modified histogram after a contrast change has been applied. These experiments illustrate the robustness of image relevant information to contrast changes and even to the removal of some level sets, when the contrast change is flat on an interval. Such experiments suggest that one can *specify* the histogram of a given image by applying the adequate contrast change. Before proceeding, we have to define the pseudo-inverses of a discrete function.

Proposition 12.2. *Let $\varphi : \mathbb{L} \to \mathbb{M}$ be a nondecreasing function from a finite set of values into another. Define two pseudo-inverse functions for φ :*

$$\varphi^{(-1)}(l) := \inf\{s \mid \varphi(s) \geq l\} \text{ and } \varphi^{((-1))}(l) := \sup\{s \mid \varphi(s) \leq l.\}$$

Then one has the following equivalences:

$$\varphi(s) \geq l \Leftrightarrow s \geq \varphi^{(-1)}(l), \quad \varphi(s) \leq l \Leftrightarrow s \leq \varphi^{((-1))}(l) \tag{12.1}$$

and the identity

$$(\varphi^{(-1)})^{((-1))} = \varphi. \tag{12.2}$$

Proof. The implication $\varphi(s) \geq l \Rightarrow s \geq \varphi^{(-1)}(l)$ is just the definition of $\varphi^{(-1)}$. The converse implication is due to the fact that the infimum on a a finite set is attained. Thus $\varphi(\varphi^{(-1)}(l)) \geq l$ and therefore $s \geq \varphi^{(-1)}(l) \Rightarrow \varphi(s) \geq l$. The identity (12.2) is a direct consequence of the equivalences (12.1). Indeed,

$$s \leq (\varphi^{(-1)})^{((-1))}(l) \Leftrightarrow \varphi^{(-1)}(s) \leq l \Leftrightarrow s \leq \varphi(l).$$

□

Exercise 12.1. Prove that if φ is increasing, $\varphi^{(-1)} \circ \varphi(l) = l$ and $\varphi^{((-1))} \circ \varphi(l) = l$. If φ is surjective, $\varphi \circ \varphi^{(-1)} = l$ and $\varphi \circ \varphi^{((-1))}(l) = l$. ■

Proposition 12.3. *Let φ be a discrete contrast change and u a digital image. Then*

$$H_{\varphi(u)} = H_u \circ \varphi^{((-1))}.$$

Proof. By (12.1), $\varphi(u) \leq l \Leftrightarrow u \leq \varphi^{((-1))}(l)$. Thus by the definitions of H_u and $H_{\varphi(u)}$,

$$H_{\varphi(u)}(l) = \text{meas}(\{\mathbf{x} \mid \varphi(u) \leq l\}) = \text{meas}(\{\mathbf{x} \mid u(\mathbf{x}) \leq \varphi^{((-1))}(l)\}) = H_u \circ \varphi^{((-1))}(l).$$

□

Let $G : \mathbb{L} \to \mathbb{M} := [0, 1, \ldots, M]$ be any discrete nondecreasing function. Can we find a contrast change $\varphi : \mathbb{L} \to \mathbb{L}$ such that the cumulative histogram of $\varphi(u)$, $H_{\varphi(u)}$ becomes equal to G? Not quite: if for instance u is constant its cumulative histogram is a one step function and Proposition 12.3 implies that $H_{\varphi(u)}$ will also be a one step function. More generally if u attains k values, then $\varphi(u)$ attains k values or less. Hence its cumulative histogram is a step function with $k+1$ steps. Yet, at least formally, the functional equation given by Proposition 12.3, $H_u \circ \varphi^{-1} = G$, leads to $\varphi = G^{-1} \circ H_u$. We know that we cannot get true inverses but we can involve pseudo-inverses. Thus, we are led to the following definition:

Definition 12.4. *Let $G : \mathbb{L} \to \mathbb{M}$ be a nondecreasing function. We call specification of u on the cumulative histogram G the image*

$$\tilde{u} := G^{((-1))} \circ H_u(u).$$

Exercise 12.2. Prove that if G and H_u are one to one, then the cumulative histogram of $\tilde{u}$ is G. Is it enough to assume that H_u is one to one? ■

Definition 12.5. *Let, for $l \in [0, L] \cap I\!N$, $G(l) = \lfloor \frac{M}{L} l \rfloor$, where $\lfloor r \rfloor$ denotes the largest integer smaller than r. Then $\tilde{u} := G^{((-1))} \circ H_u(u)$ is called the* uniform equalization of u. *If v is another discrete image and one takes $G = H_v$, $\tilde{u} := H_v^{((-1))} \circ H_u(u)$ is called the* specification of u on v.

When H_u is one to one, one can reach by applying a contrast change to u any specified cumulative histogram G. Otherwise, the above definitions do the best that can be expected and are actually quite efficient. For instance in the "marshland experiment" (Figure 12.1) the equalized histogram and its cumulative histogram are displayed on the second row. The cumulative histogram is very close to its goal, the linear function. The equalized histogram does not look flat but a sliding average of it would look almost flat.

Yet it is quite dangerous to specify the histogram of an image with an arbitrary histogram specification. This fact is illustrated in Figures 12.1 and 12.2 where a uniform equalization erases existing textures by making them too flat (Figure 12.1) but also enhances the quantization noise in low contrasted regions and produces artificial edges or textures (see Figure 12.2).

12.1 Midway equalization

We have seen that if one specifies u on v, then u inherits roughly the histogram of v. It is sometimes more adequate to bring the cumulative histograms of u and v towards a cumulative histogram which would be "midway" between both. Indeed, if we want to compare visually and numerically two images, it is useful to give them first the same dynamic range and luminance. Thus we wish:

- From two images u and v, construct by contrast changes two images $\tilde{u}$ and $\tilde{v}$, which have a similar cumulative histogram.
- This common cumulative histogram h should stand "midway" between the previous cumulative histograms of u and v, and be as close as possible to each of them. This treatment must avoid to favor one cumulative histogram rather than the other.

Definition 12.6. *Let u and v be two discrete images. Set*

$$\Phi := \frac{1}{2}\left(H_u^{(-1)} + H_v^{(-1)}\right).$$

We call midway cumulative histogram *of u and v the function*

$$G := \Phi^{((-1))} = \left(\frac{1}{2}(H_u^{(-1)} + H_v^{(-1)})\right)^{((-1))} \tag{12.3}$$

and "midway specifications" of u and v the functions $\tilde{u} := \Phi \circ H_u(u)$ and $\tilde{v} := \Phi \circ H_v(v)$.

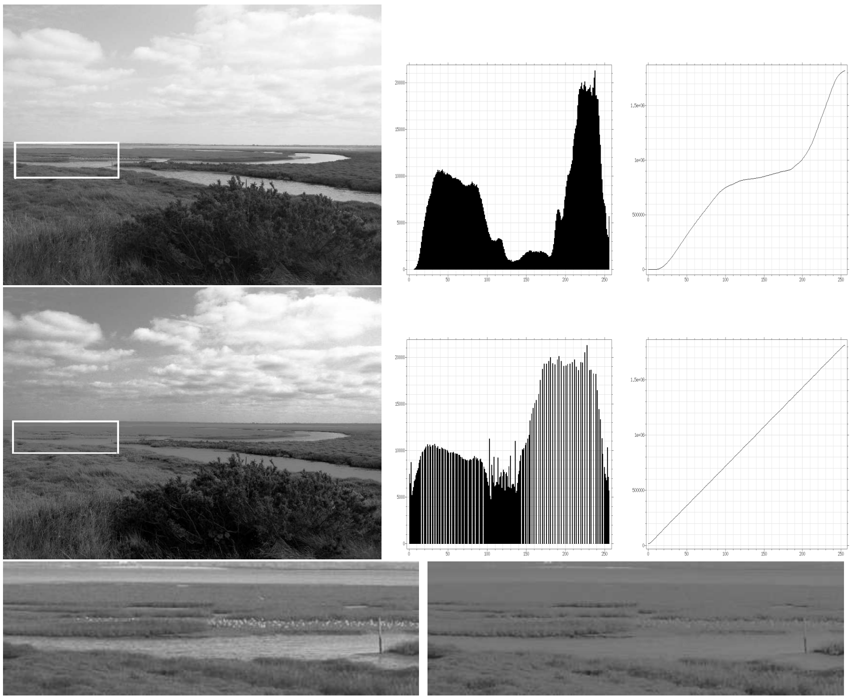

Figure 12.1: *First row: Image u, the corresponding grey level histogram h_u, and the cumulative histogram H_u. Second row: Equalized image $H_u(u)$, its histogram and its cumulative histogram. In the discrete case, histogram equalization flattens the histogram as much as possible. We see on this example that image equalization can be visually harmful. In this marshland image, after equalization, the water is no more distinguishable from the vegetation. The third row shows a zoom on the rectangular zone, before and after equalization.*

Exercise 12.3. Let u and v be two constant images, whose values are a and b. Prove that their "midway" function is the right one, namely a function w which is constant and equal to $\frac{a+b}{2}$. ■

Exercise 12.4. Prove that if we take as a definition of the midway histogram

$$G := \left(\frac{1}{2}(H_u^{((-1))} + H_v^{((-1))})\right)^{(-1)},$$

then for two constant images $u = a$ and $v = b$ the midway image is constant and equal to $[1/2(a+b) - 1]$. This proves that Definition 12.6 is better. ■

Exercise 12.5. Prove that if u is a discrete image and f and g two nondecreasing functions, then the midway image of $f(u)$ and $g(u)$ is $\frac{f(u)+g(u)}{2}$. ■

Exercise 12.6. If we want the "midway" cumulative histogram H to be a compromise between H_u and H_v, the most elementary function that we could imagine is their average, which amounts to average their histograms as well. However, the following example proves that this idea is not judicious at all.

Consider two images whose histograms are "crenel" functions on two disjoint intervals, for instance $u(\mathbf{x}) := ax$, $v(\mathbf{x}) = bx + c$. Compute a, b, c in such a way that h_u and h_v have disjoint supports. Then compute the specifications of u and v on the

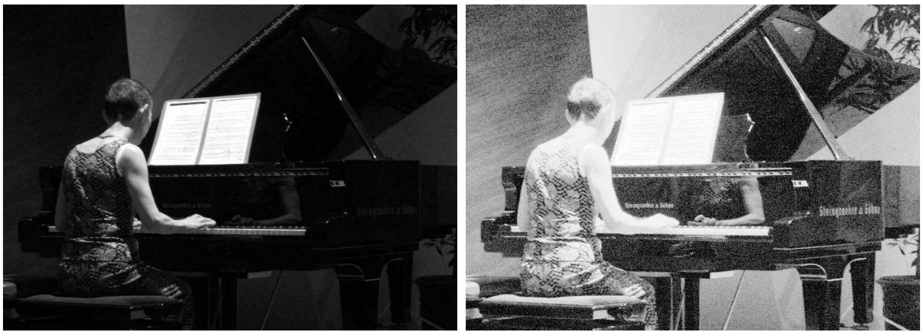

Figure 12.2: *Effect of histogram equalization on the quantization noise. On the left, the original image. On the right, the same image after histogram equalization. The effect of this equalization on the dark areas (the piano, the left part of the wall), which are low contrasted, is perceptually dramatic. We see many more details but the quantization noise has been exceedingly amplified.*

mean cumulative histogram $G := \frac{H_u + H_v}{2}$. Compare with their specifications on the midway cumulative histogram. ■

12.2 Midway equalization on image pairs

Results on a stereo pair

The top of Figure 12.3 shows a pair of aerial images in the region of Toulouse. Although the angle variation between both views is small, and the photographs are taken at nearly the same time, we see that the lightning conditions vary significantly (the radiometric differences can also come from a change in camera settings). The second line shows the result of the specification of the histogram of each image on the other one. The third line shows both images after equalization.

If we scan some image details, as illustrated on Figure 12.4, the damages caused by a direct specification become obvious. Let us specify the darker image on the brightest one. Then the information loss, due to the reduction of dynamic range, can be detected in the brightest areas. Look at the roof of the bright building in the top left corner of the image (first line of Figure 12.4): the chimneys project horizontal shadows on the roof. In the specified image, these shadows have almost completely vanished, and we cannot even discern the presence of a chimney anymore. In the same image after equalization, the shadows are still entirely recognizable, and their size reduction remains minimal. The second line of Figure 12.4 illustrates the same phenomenon, observed in the bottom center of the image. The structure present at the bottom of the image has completely disappeared after specification and remains visible after midway equalization. These examples show how visual information can be lost by specification and how midway algorithms reduce significantly this loss.

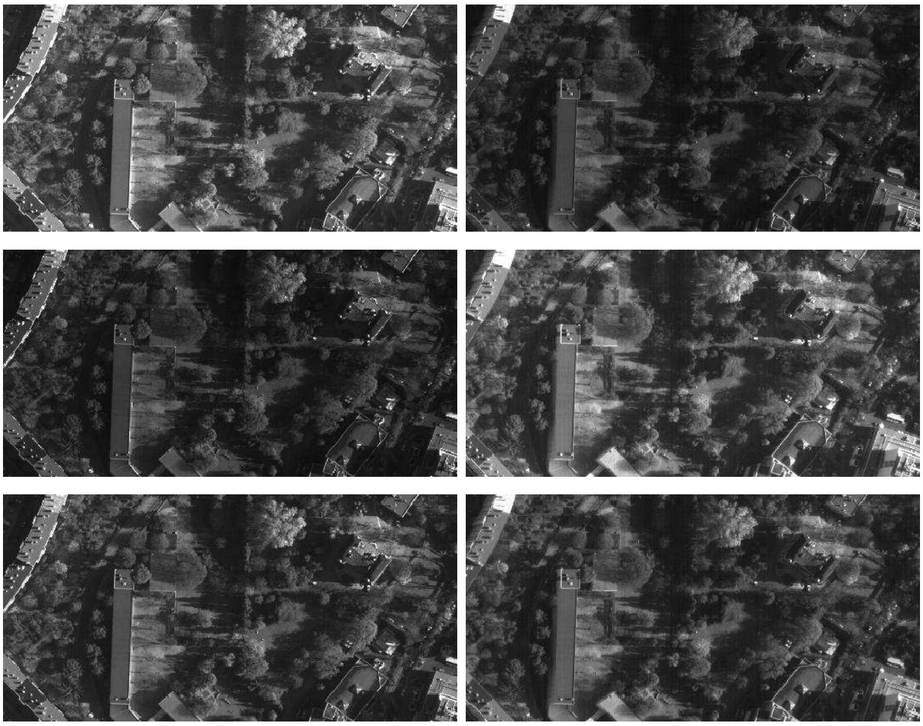

Figure 12.3: Stereo pair: two pieces of aerial images of a region of Toulouse. Same images after specification of their histograms on each other (left: the histogram of the first image has been specified on the second, and right: the histogram of the second image has been specified on the first). Stereo pair after midway equalization.

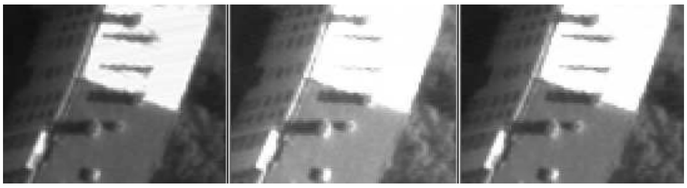

Figure 12.4: *Extracts from the stereo pair shown on Figure 12.3. From left to right: in the original image, in the specified one, in the original image after midway equalization. Notice that no detail is lost in the midway image, in contrast with the middle image.*

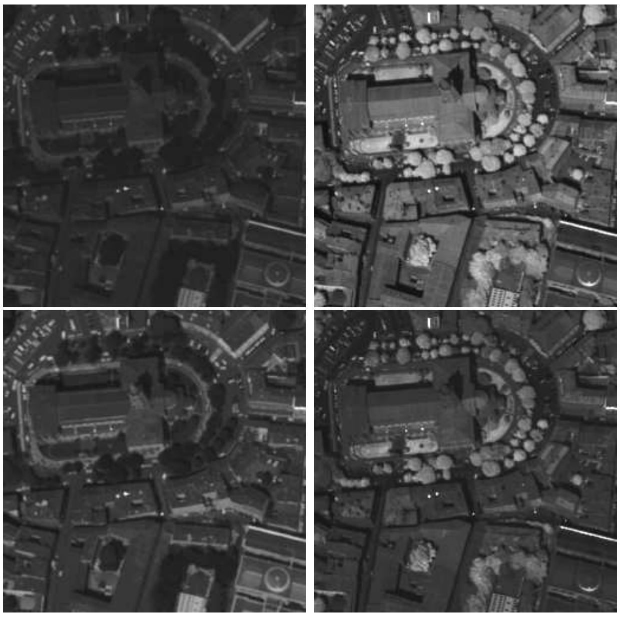

Figure 12.5: *First line: two images of Toulouse (blue and infrared channel). Second line: same images after midway equalization.*

Multi-Channel images

The top of Figure 12.5 shows two pieces of multi-channel images of Toulouse. The first one is extracted from the blue channel, and the other one from the infrared channel. The second and third line of the same figure show the same images after midway equalization. The multichannel images have the peculiarity to present contrast inversions : for instance, the trees appear to be darker than the church in the blue channel, and are naturally brighter than the church in the infrared channel. The midway equalization being limited to increasing contrast changes, it obviously cannot handle these contrast inversions. In spite of these contrast inversions, the results remain visually good, which underlines the robustness of the method gives globally a good equalization.

Photographs of the same painting

The top of Figure 12.6 shows two different snapshots of the same painting, *Le Radeau de la Méduse*[1], by Théodore Géricault (small web public versions). The second one is brighter and seems to be damaged at the bottom left. The second line shows the same couple after midway equalization. Finally, the last line of Figure 12.6 shows the difference between both images after equalization. We see clear differences around the edges, due to the fact that the original images are not completely similar from the geometric point of view.

12.2.1 Movie equalization

One can define a midway cumulative histogram to an arbitrary number of images. This is extremely useful for the removal of flicker in old movies. Flicker has multiple causes, physical, chemical or numerical. The overall contrast of successive images of the same scene in a movie oscillates, some images being dark and others bright. Our main assumption is that image level sets are globally preserved from one image to the next, even if their level evolves. This leads to the adoption of a movie equalization method preserving globally all level sets of each image. We deduce from Theorem 11.7 in the previous chapter that the correction must be a global contrast change on each image. Thus the only left problem is to *specify* a common cumulative histogram (and therefore a common histogram) to all images of a given movie scene. Noticing that the definition of G in (12.3) for two images simply derives from a mean, its generalization is easy. Let us denote $u(t,\mathbf{x})$ the movie (now a discrete time variable has been added) and by H^t the cumulative histogram function of $\mathbf{x} \to u(t,\mathbf{x})$ at time t. Since flicker is localized in time, the idea is to define a time dependent cumulative histogram function K_t^h which will be the "midway" cumulative histogram of the cumulative histograms in an interval $[t-h, t+h]$. Of course the linear scale space theory of Chapter 3 applies here. The ideal average is gaussian. Hence the following definition.

Definition 12.7. *Let $u(t,\mathbf{x})$ be a movie and denote by H_t the cumulative histogram of $u(t) : \mathbf{x} \to u(t,\mathbf{x})$. Consider a discrete version of the 1-D gaussian $G_h(t) = \frac{1}{(4\pi h)^{\frac{1}{2}}} e^{-\frac{t^2}{4h}}$. Set*

$$\Phi_{(t,l)} := \int G_h(t-s)(H_s^{(-1)})(l)ds.$$

We call "midway gaussian cumulative histogram at scale h" of the movie $u(t,\mathbf{x})$ the time dependent cumulative histogram

$$\mathbb{G}_{(t,l)} := \Phi_{(t,l)}^{((-1))} = \left(\int G_h(t-s)(H_s^{(-1)})(l)ds \right)^{((-1))} \tag{12.4}$$

and "midway specification" of the movie $u(t)$ the function $\tilde{u}(t) := \Phi \circ H_{u(t)}(u(t))$. If $H_{u(t)}$ is surjective, then $\tilde{u}(t)$ has $G_{(t,l)}$ as common cumulative histogram.

Notice that this is a straightforward extension of Definition 12.6.

[1] Muse du Louvre, Paris.

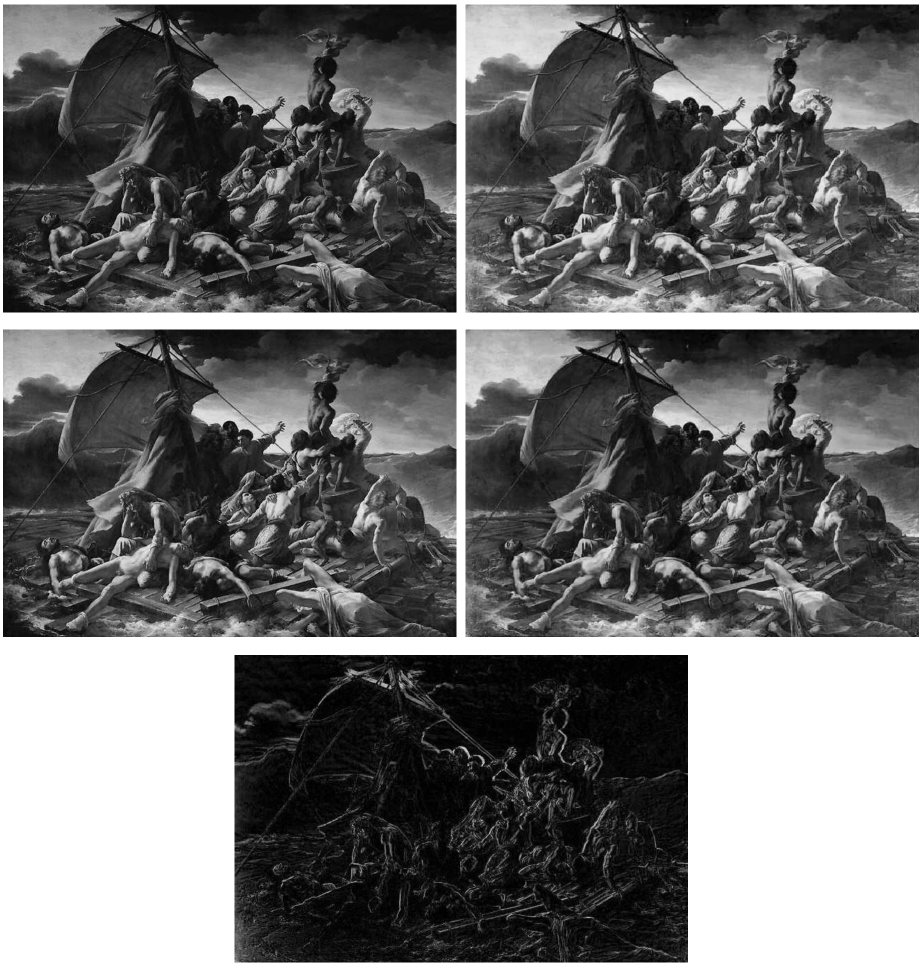

Figure 12.6: Two shots of the *Radeau de la Méduse, by Géricault.* The same images after midway equalization. Image of the difference between both images after equalization. The boundaries appearing in the difference are mainly due to the small geometric distortions between the initial images.

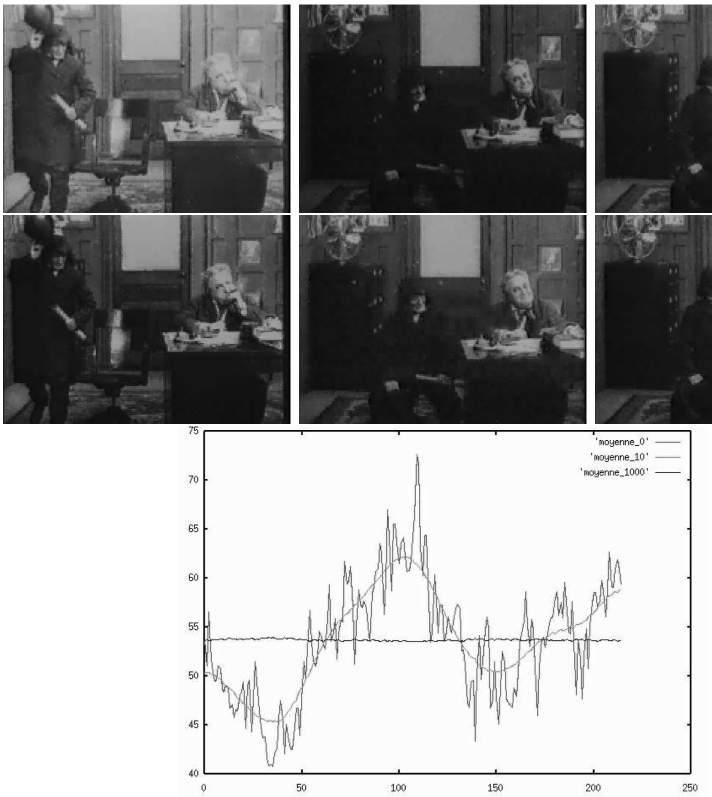

Figure 12.7: *(a) Three images of Chaplin' s film His New Job, taken at equal intervals of time. This extract of the film suffers from a severe real flicker. (b) Same images after the Scale-Time Equalization at scale* $s = 100$*. The flicker observed before has globally decreased. (c) Evolution of the mean of the current frame in time and at three different scales. The most oscillating line is the mean of the original sequence. The second one is the mean at scale* $s = 10$*. The last one, almost constant, corresponds to the large scale* $s = 1000$*. As expected the mean function is smoothed by the heat equation.*

The implementation and experimentation is easy. We simply show in Figure 12.7 three images of Chaplin' s film *His New Job*, taken at equal intervals of time. This extract of the film suffers from a severe real flicker. This flicker is corrected at the scale where, after gaussian midway equalization, the image mean becomes nearly constant through the sequence. The effects of this equalization are usually excellent. They are easily extended to color movies by processing each channel independently.

12.3 Comments and references

Histogram specification As we have seen *histogram specification* [95] can be judicious if both images have the same kind of dynamic range. For the same reason as in equalization, this method can also product contouring artifacts. The midway theory is essentially based on Julie Delons' PhD and papers [67], [68] where she defines two midway histogram interpolation methods. One of them, the *square root* method involves the definition of a square root of any nondecreasing function g, namely a function g such that $f \circ f = g$. Assume that u and v come from the same image (this intermediate image is unknown), up to two contrast changes f and f^{-1}. The function f is unknown, but satisfies formally the equality $H_u \circ f = H_v \circ f^{-1}$. Thus

$$H_u{}^{-1} \circ H_v = f \circ f.$$

It follows that the general method consists in building an increasing function f such that $f \circ f = H_u{}^{-1} \circ H_v$ and replacing v by $f(v)$ and u by $f^{-1}(u)$. This led Delon [**?**] to call this new histogram midway method, the "square root" equalization. The midway interpolation developed in this chapter uses mainly J. Delon's second definition of the midway cumulative histogram as the harmonic mean of the cumulative histograms of both images. This definition is preferable to the square root. Indeed, both definitions yield very similar results but the harmonic mean extends easily to an arbitrary number of images and in particular to movies [68]. The Cox, Roy and Hingorani algorithm defined in [58] performs a midway equalization. They called their algorithm "Dynamic histogram warping" and its aim is to give a common cumulative histogram (and therefore a common histogram) to a pair of images. Although their method is presented as a dynamic algorithm, there is a very simple underlying formula, which is the harmonic mean of cumulative histograms discovered by Delon [67].

Chapter 13

Contrast-Invariant Monotone Operators

A function operator T is monotone if $u \geq v \Rightarrow Tu \geq Tv$. A set operator $\mathcal{T}$ is monotone if $X \subset Y$ implies $\mathcal{T}X \subset \mathcal{T}Y$. We are mainly interested in monotone function operators, since they are nonlinear generalizations of linear smoothing using a nonnegative convolution kernel. We have already argued that for image analysis to be robust, the operators must also be contrast invariant. The overall theme here will be to develop the equivalence between monotone contrast-invariant function operators and monotone set operators. This equivalence is based on one of the fundamentals of mathematical morphology described in Chapter 11: A real-valued function is completely described by its level sets.

This allows one to process an image u by processing separately its level sets by some monotone set operator $\mathcal{T}$ and defining the processed image by the *superposition principle*

$$Tu = \sup\{\lambda, \mathbf{x} \in \mathcal{T}(\mathcal{X}_\lambda u)\}.$$

Such an operator is called in digital technology a *stack filter*, since it processes an image as a stack of level sets. Conversely, we shall associate with any contrast invariant monotone function operator T a monotone set operator by setting

$$\mathcal{T}(\mathcal{X}_\lambda u) = \mathcal{X}_\lambda(Tu).$$

Such a construction is called *a level set extension of* T.

Several questions arise, which will be all answered positively once the functional framework is fixed: Are stack filters contrast invariant? Conversely, is any monotone contrast invariant operator a stack filter? Is any monotone set operator the level set extension of its stack filter?

In Section 13.1 we shall make definitions precise and give some remarkable conservative properties of contrast invariant monotone operators. Section 13.2 is devoted to stack filters and shows that they are monotone and contrast invariant. Section 13.3 defines the level set extension and shows the converse statement: Any contrast invariant monotone operator is a stack filter. Section 13.4 applies this construction to a remarkable denoising stack filter due to Vincent and Serra, the area opening.

13.1 Contrast-invariance

13.1.1 Set monotone operators

We will be mostly dealing with function operators T defined on $\mathcal{F}$ and set operators $\mathcal{T}$ defined on $\mathcal{L}$, but sometimes also defined on $\mathcal{M}$. We denote by $\mathcal{D}(\mathcal{T})$ the domain of $\mathcal{T}$. Now, all set operators we shall consider in practice are defined first on subsets of $\mathbb{R}^N$.

Definition 13.1. *Let $\mathcal{T}$ a monotone operator defined on a set of subsets of $\mathbb{R}^N$. We call* standard extension of $\mathcal{T}$ to S_N *the operator, still denoted by $\mathcal{T}$, defined by*

$$\mathcal{T}(X) = \mathcal{T}(X \setminus \{\infty\}) \cup (X \cap \{\infty\}).$$

In other terms if X doesn't contain ∞, $\mathcal{T}(X)$ is already defined and if X contains ∞, $\mathcal{T}(X)$ contains it too. Thus a standard extension satisfies $\infty \in \mathcal{T}X \Leftrightarrow \infty \in X$.

Remark 13.2. *Let us examine the case where $\mathcal{T}$ is initially defined on $\mathcal{C}$, the set of all closed subsets of $\mathbb{R}^N$. There are only two kinds of sets in $\mathcal{L}$, namely*

- *compact sets of $\mathbb{R}^N$*
- *sets of the form $X = C \cup \{\infty\}$, where C is a closed set of $\mathbb{R}^N$.*

Thus the standard extension of $\mathcal{T}$ extends $\mathcal{T}$ to $\mathcal{L}$, the set of all closed (and therefore compact) subsets of S_N.

All of the usual monotone set operators used in shape analysis satisfy a small list of standard properties which it is best to fix now. Their meaning will come obvious in examples.

Definition 13.3. *We say that a set operator $\mathcal{T}$ defined on its domain $D(\mathcal{T})$ is* standard monotone *if*

- *$X \subset Y \implies \mathcal{T}X \subset \mathcal{T}Y$;*
- *$\infty \in \mathcal{T}X \iff \infty \in X$;*
- *$\mathcal{T}(\emptyset) = \emptyset$, $\mathcal{T}(S_N) = S_N$;*
- *$\mathcal{T}(X)$ is bounded in $\mathbb{R}^N$ if X is;*
- *$\mathcal{T}(X)^c$ is bounded in $\mathbb{R}^N$ if X^c is.*

Definition 13.4. *Let $\mathcal{T}$ be a monotone set operator on its domain $\mathcal{D}(\mathcal{T})$. We call* dual domain *the set*

$$\mathcal{D}(\tilde{\mathcal{T}}) := \{X \subset S_N \mid X^c \in \mathcal{D}(\mathcal{T})\}.$$

We call dual *of $\mathcal{T}$ the operator $X \to \tilde{\mathcal{T}}X = (\mathcal{T}(X^c))^c$, defined on $\mathcal{D}(\tilde{\mathcal{T}})$.*

Proposition 13.5. *$\mathcal{T}$ is a standard monotone operator if and only if $\tilde{\mathcal{T}}$ is.*

Exercise 13.1. Prove it! ∎

13.1.2 Monotone function operators

Function operators are usually defined on $\mathcal{F}$, the set of continuous functions having some limit $u(\infty)$ at infinity. We shall always assume that this limit is preserved by T, that is, $Tu(\infty) = u(\infty)$. Think that images are usually compactly supported. Thus $u(\infty)$ is the "color of the frame" for a photograph. There is no use in changing this color.

Definition 13.6. *We say that a function operator $T : \mathcal{F} \to \mathcal{F}$ is standard monotone if for all $u, v \in \mathcal{F}$,*

$$u \geq v \implies Tu \geq Tv; \quad Tu(\infty) = u(\infty). \tag{13.1}$$

Exercise 13.2. Is the operator T defined by $(Tu)(\mathbf{x}) = u(\mathbf{x}) + 1$ standard monotone? ■

Recall from Chapter 11 that any nondecreasing continuous surjection $g : \mathbb{R} \to \mathbb{R}$ is called a contrast change.

Definition 13.7. *A function operator T: $\mathcal{F} \to \mathcal{F}$ is said to be contrast invariant if for every $u \in \mathcal{F}$ and every contrast change g,*

$$g(Tu) = Tg(u). \tag{13.2}$$

Checking contrast invariance with increasing contrast changes will make our life simpler.

Lemma 13.8. *A monotone operator is contrast invariant if and only if it commutes with strictly increasing contrast changes.*

Proof. Let g be a contrast change. We can find strictly increasing continuous functions g_n and $h_n : \mathbb{R} \to \mathbb{R}$ such that $g_n(s) \to g(s)$, $h_n(s) \to g(s)$ for all s and $g_n \leq g \leq h_n$ (see Exercise 13.12.) Thus, by using the commutation of T with increasing contrast changes, we have

$$T(g(u)) \geq T(g_n(u)) = g_n(Tu) \to g(Tu) \text{ and}$$

$$T(g(u)) \leq T(h_n(u)) = h_n(Tu) \to g(Tu),$$

which yields $T(g(u)) = g(Tu)$. □

Let us give some notable properties entailed by the monotonicity and the contrast invariance.

Lemma 13.9. *Let T be standard monotone contrast invariant operator. Then for every constant function $u \equiv c$ one has $Tu \equiv c$.*

Proof. Let g be a contrast change such that $g(s) = s$ for $\inf Tu \leq s \leq \sup Tu$. Since $Tu(\infty) = u(\infty) = c$, this implies that $\inf Tu \leq c \leq \sup Tu$ and therefore $g(c) = c$, which means $g(u) = u$.. By the contrast invariance we therefore obtain $Tu = Tg(u) = g(Tu) \equiv c$. □

We have indicated several times the importance of image operators being contrast invariant. In practice, image operators are also translation invariant. For $\mathbf{x} \in \mathbb{R}^N$ we are going to use the notation $\tau_{\mathbf{x}}$ to denote the translation operator for both sets and functions: For $X \in \mathcal{M}$, $\tau_{\mathbf{x}} X = \{\mathbf{x} + \mathbf{y} \mid \mathbf{y} \in X\}$, and for $u \in \mathcal{F}$, $\tau_{\mathbf{x}} u$ is defined by $\tau_{\mathbf{x}} u(\mathbf{y}) = u(\mathbf{y} - \mathbf{x})$. Since elements of $\mathcal{M}$ can contain ∞, we specify that $\infty \pm \mathbf{x} = \infty$ when $\mathbf{x} \in \mathbb{R}^N$. This implies that $\tau_{\mathbf{x}} u(\infty) = u(\infty)$.

Definition 13.10. *A set operator $\mathcal{T}$ is said to be translation invariant if its domain is translation invariant and if for all $X \in \mathcal{D}(\mathcal{T})$ and $\mathbf{x} \in \mathbb{R}^N$,*

$$\tau_{\mathbf{x}} \mathcal{T} X = \mathcal{T} \tau_{\mathbf{x}} X.$$

A function operator T is said to be translation invariant if for all $u \in \mathcal{F}$ and $\mathbf{x} \in \mathbb{R}^N$,

$$\tau_{\mathbf{x}} T u = T \tau_{\mathbf{x}} u.$$

We say that a function operator T commutes with the addition of constants if $u \in \mathcal{F}$ and $c \in \mathbb{R}$ imply $T(u + c) = Tu + c$.

Contrast-invariant operators clearly commute with the addition of constants: Consider the contrast change defined by $g(s) = s + c$.

Lemma 13.11. *Let T be a translation-invariant monotone function operator on $\mathcal{F}$ that commutes with the addition of constants. If $u \in \mathcal{F}$ is K-Lipschitz on $\mathbb{R}^N$, namely $|u(\mathbf{x}) - u(\mathbf{y})| \le K|\mathbf{x} - \mathbf{y}|$ for all $\mathbf{x}$, $\mathbf{y}$ in $\mathbb{R}^N$, then so is Tu.*

Proof. For any $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{y} \in \mathbb{R}^N$, and $\mathbf{z} \in S_N$, we have

$$u(\mathbf{y} + \mathbf{z}) - K|\mathbf{x} - \mathbf{y}| \le u(\mathbf{x} + \mathbf{z}) \le u(\mathbf{y} + \mathbf{z}) + K|\mathbf{x} - \mathbf{y}|. \tag{13.3}$$

These inequalities work for $\mathbf{z} = \infty$ because $u(\mathbf{y} + \infty) = u(\mathbf{x} + \infty) = u(\infty)$. Thus we can write them as inequalities between functions on S_N:

$$\tau_{-\mathbf{y}} u - K|\mathbf{x} - \mathbf{y}| \le \tau_{-\mathbf{x}} u \le \tau_{-\mathbf{y}} u + K|\mathbf{x} - \mathbf{y}|. \tag{13.4}$$

Since T is monotone, we can apply T to the functions in (13.4) and preserve the inequalities, which yields

$$T(\tau_{-\mathbf{y}} u - K|\mathbf{x} - \mathbf{y}|) \le T(\tau_{-\mathbf{x}} u) \le T(\tau_{-\mathbf{y}} u + K|\mathbf{x} - \mathbf{y}|).$$

Now use the fact that T commutes with the addition of constants the translation invariance of T to obtain

$$\tau_{-\mathbf{y}}(Tu) - K|\mathbf{x} - \mathbf{y}| \le \tau_{-\mathbf{x}}(Tu)) \le T(\tau_{-\mathbf{y}} u) + K|\mathbf{x} - \mathbf{y}|).$$

Taking the values of these functions at 0 yields

$$Tu(\mathbf{y}) - K|\mathbf{x} - \mathbf{y}| \le Tu(\mathbf{x}) \le Tu(\mathbf{y}) + K|\mathbf{x} - \mathbf{y}|,$$

which is the announced result. □

We say that an operator is monotone on a set of functions if $u \ge v \Rightarrow Tu \ge Tv$. Clearly all above proofs do not depend upon the fact that the operator is standard, but just upon its translation invariance and monotonicity. Thus, by considering the proof of Lemma 13.11 and the definition of uniform continuity (Definition 1.3), one obtains the following generalizations.

Corollary 13.12. *Assume that T is a translation-invariant monotone operator on a set of uniformly continuous functions, that commutes with the addition of constants. Then Tu is uniformly continuous on $\mathbb{R}^N$ with the same modulus of continuity. In particular if u is L-Lipschitz on $\mathbb{R}^N$, then so is Tu.*

Exercise 13.3. Prove corollary 13.12.

■

13.2 Stack filters

Definition 13.13. *We say that a function operator T is obtained from a monotone set operator $\mathcal{T}$ as a* stack filter *if*

$$Tu(\mathbf{x}) = \sup\{\lambda \mid \mathbf{x} \in \mathcal{T}(\mathcal{X}_\lambda u)\} \tag{13.5}$$

for every $\mathbf{x} \in S_N$.

The relation (13.5) has practical implications. It means that Tu can be computed by applying $\mathcal{T}$ separately to each characteristic function of the level sets $\mathcal{X}_\lambda u$. This leads to the following *stack filter* algorithm.

$$\begin{tikzcd}
 & \mathcal{X}_\lambda u \arrow[r] & \mathcal{T}(\mathcal{X}_\lambda u) \arrow[dr] & \\
u \arrow[ur] \arrow[dr] & \vdots & & Tu(\mathbf{x}) = \sup\{\lambda \mid \mathbf{x} \in \mathcal{T}(\mathcal{X}_\lambda u)\}. \\
 & \mathcal{X}_\mu u \arrow[r] & \mathcal{T}(\mathcal{X}_\mu u) \arrow[ur] &
\end{tikzcd}$$

The image u is decomposed into the stack of level sets. Each level set is processed independently by the monotone operator $\mathcal{T}$. This yields a new stack of sets $\mathcal{T}(\mathcal{X}_\lambda u)$ and Formula (13.5) always defines a function Tu. Now, this construction will be perfect only if

$$\mathcal{X}_\lambda(Tu) = \mathcal{T}(\mathcal{X}_\lambda u). \tag{13.6}$$

Definition 13.14. *When (13.6) holds, we say that T "commutes with thresholds", or that T and $\mathcal{T}$ satisfy the "commutation with threshold" property.*

Of course, this commutation can hold only if $\mathcal{T}$ sends $\mathcal{L}$ into itself. A further condition which turns out to be necessary is introduced in the next definition.

Definition 13.15. *We say that a monotone set operator $\mathcal{T} : \mathcal{L} \to \mathcal{L}$ is* upper semicontinuous *if for every sequence of compact sets $X_n \in \mathcal{D}(\mathcal{T}) = \mathcal{L}$ such that $X_{n+1} \subset X_n$, we have*

$$\mathcal{T}(\bigcap_n X_n) = \bigcap_n \mathcal{T}(X_n). \tag{13.7}$$

Exercise 13.4. Show that a monotone operator $\mathcal{T} : \mathcal{L} \to \mathcal{L}$ is upper semicontinuous if and only if it satisfies, for every family $(X_\lambda)_{\lambda \in \mathbb{R}} \subset \mathcal{L}$ such that $X_\lambda \subset X_\mu$ for $\lambda > \mu$, the relation $\mathcal{T}(\bigcap_\lambda X_\lambda) = \bigcap_\lambda \mathcal{T}(X_\lambda)$. ■

Exercise 13.5. Show that a monotone operator on $\mathcal{L}$ is upper semicontinuous if and only if it satisfies (13.7) for every sequence of compact sets X_n such that $X_{n+1} \subset X_n^\circ$. Hint: Since S_N is the unit sphere in $\mathbb{R}^{N+1}$, one can endow it with the euclidian distance d in $\mathbb{R}^{N+1}$. Given a nondecreasing sequence Y_n in $\mathcal{L}$, set $X_n = \{\mathbf{x}, d(\mathbf{x}, Y_n) \leq \frac{1}{n}\}$. Then apply (13.7) to X_n and check that $\bigcap_n X_n = \bigcap_n Y_n$. ■

Exercise 13.6. Show that a monotone operator $\mathcal{T} : \mathcal{L} \to \mathcal{L}$ is upper semicontinuous if and only if it satisfies, for every family $(X_\lambda)_{\lambda \in \mathbb{R}} \subset \mathcal{L}$ such that $X_\lambda \subset X_\mu^\circ$ for $\lambda > \mu$, the relation $\mathcal{T}(\bigcap_\lambda X_\lambda) = \bigcap_\lambda \mathcal{T}(X_\lambda)$. ■

Theorem 13.16. *Let $\mathcal{T}$: $\mathcal{L} \to \mathcal{M}$ be a translation invariant standard monotone set operator. Then the associated stack filter T is translation invariant, contrast invariant and standard monotone from $\mathcal{F}$ into itself. If, in addition, $\mathcal{T}$ is upper semicontinuous, then T commutes with thresholds.*

Proof that T is monotone. One has

$$u \leq v \Leftrightarrow (\forall \lambda,\ \mathcal{X}_\lambda u \subset \mathcal{X}_\lambda v).$$

Since $\mathcal{T}$ is monotone, we deduce that

$$\forall \lambda,\ \mathcal{T}(\mathcal{X}_\lambda u) \subset \mathcal{T}(\mathcal{X}_\lambda v)$$

which by (13.5) implies $Tu \leq Tv$.

Proof that T is contrast invariant.
By Lemma 13.8 we can take g strictly increasing and therefore a bijection from $\mathbb{R}$ to $\mathbb{R}$. We notice that :
For $\lambda > g(\sup u)$, $\mathcal{X}_\lambda g(u) = \emptyset$ and therefore $\mathcal{T}(\mathcal{X}_\lambda g(u)) = \emptyset$.
For $\lambda < g(\inf u)$, $\mathcal{X}_\lambda g(u) = S_N$ and therefore $\mathcal{T}(\mathcal{X}_\lambda g(u)) = S_N$.
Thus using (13.5) we can restrict the range of λ in the definition of $T(g(u))(\mathbf{x})$:

$$\begin{aligned} T(g(u))(\mathbf{x}) &= \sup\{\lambda,\ g(\inf u) \leq \lambda \leq g(\sup u),\ \mathbf{x} \in \mathcal{T}(\mathcal{X}_\lambda g(u))\} \\ &= \sup\{g(\mu),\ \mathbf{x} \in \mathcal{T}(\mathcal{X}_{g(\mu)} g(u))\} \\ &= \sup\{g(\mu),\ \mathbf{x} \in \mathcal{T}(\mathcal{X}_\mu u)\} = g(Tu(\mathbf{x})). \end{aligned}$$

Proof that Tu belongs to $\mathcal{F}$.
T is by construction translation invariant. By Corollary 13.12, Tu is uniformly continuous on $\mathbb{R}^N$. Let us prove that $Tu(\mathbf{x}) \to u(\infty)$ as $\mathbf{x} \to \infty$. We notice that for $\lambda > u(\infty)$, $\mathcal{X}_\lambda u$ is bounded. Since $\mathcal{T}$ is standard monotone $\mathcal{T}(\mathcal{X}_\lambda u)$ is bounded too. Now, by (13.5), $Tu(\mathbf{x}) \leq \lambda$ if $\mathbf{x} \in \mathcal{T}(\mathcal{X}_\lambda u)^c$. This last condition is satisfied if $\mathbf{x}$ is large enough and we deduce that $\limsup_{\mathbf{x} \to \infty} Tu(\mathbf{x}) \leq u(\infty)$. In the same way notice that $(\mathcal{X}_\lambda u)^c$ is bounded if $\lambda < u(\infty)$. So by the same argument, we also get $\liminf_{\mathbf{x} \to \infty} Tu(\mathbf{x}) \geq u(\infty)$. $\mathcal{T}$ being standard, it is easily checked using (13.5) that $Tu(\infty) = u(\infty)$. Thus, Tu is continuous on $\mathbb{R}^N$ and at ∞ and therefore on S_N.

Proof that T commutes with thresholds, when $\mathcal{T}$ is upper semicontinuous.
Let us show that the sets $Y_\lambda = \mathcal{T}(\mathcal{X}_\lambda u)$ satisfy the properties (i) and (ii) in

Proposition 11.2. By the monotonicity of $\mathcal{T}$, $Y_\lambda \subset Y_\mu$ for $\lambda > \mu$. Since $\mathcal{T}(\emptyset) = \emptyset$, we have

$$Y_\lambda = \mathcal{T}(\mathcal{X}_\lambda u) = \mathcal{T}(\emptyset) = \emptyset$$

for $\lambda > \max u$ and, in the same way $Y_\lambda = S_N$ for $\lambda < \min u$. So Tu has the same bounds as u. This proves Property (i). As for Property (ii), we have for every λ, using the upper semicontinuity and exercise 13.4,

$$Y_\lambda = \mathcal{T}(\mathcal{X}_\lambda u) = \mathcal{T}(\bigcap_{\mu<\lambda} \mathcal{X}_\mu u) = \bigcap_{\mu<\lambda} \mathcal{T}(\mathcal{X}_\mu u) = \bigcap_{\mu<\lambda} Y_\mu.$$

So by applying the converse statement of Proposition 11.2, we deduce that

$$\mathcal{X}_\lambda(Tu) = \mathcal{T}(\mathcal{X}_\lambda u).$$

□

Exercise 13.7. Check that $Tu(\infty) = u(\infty)$, as claimed in the former proof. ■

The upper semicontinuity of $\mathcal{T}$ is necessary to ensure the commutation with thresholds. See Exercise 13.21. The assumption that $\mathcal{T}$ sends bounded sets of $\mathbb{R}^N$ on bounded sets of $\mathbb{R}^N$ and complementary sets of bounded sets onto complementary sets of bounded sets also is necessary to ensure that Tu is continuous at ∞: see Exercise 13.16.

13.3 The level set extension

Our aim here is just the converse as in the former section. We wish to associate a standard monotone set operator $\mathcal{T}$ from $\mathcal{L}$ to $\mathcal{L}$ with any contrast invariant standard monotone function operator T, in such a way that the whole machinery works, namely both operators satisfy the commutation with threshold property $\mathcal{T}(\mathcal{X}_\lambda u) = \mathcal{X}_\lambda(Tu)$ and T is the stack filter of $\mathcal{T}$.

Lemma 13.17. *Let $u \le 0$ and $v \le 0 \in \mathcal{F}$ and assume that $\mathcal{X}_0 u = \mathcal{X}_0 v \,(\neq \emptyset)$. Then there is a contrast change h such that $h(0) = 0$ and $u \ge h(v)$.*

Proof. Define

$$\tilde{h}(r) = \begin{cases} \min\{u(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}_r v\} & \text{if } \min v \le r \le 0; \\ r & \text{if } r > 0; \\ \min u - \min v + r & \text{if } r \le \min v. \end{cases}$$

Notice that $\tilde{h}(0) = 0$ and that $\tilde{h}$ is nondecreasing. The following relation holds for all $\mathbf{x} \in \mathbb{R}^N$ by the definition of $\tilde{h}$ and because $u(\mathbf{x})$ belongs to the set $\{u(\mathbf{y}) \mid v(\mathbf{y}) \ge v(\mathbf{x})\}$:

$$u(\mathbf{x}) \ge \min\{u(\mathbf{y}) \mid v(\mathbf{y}) \ge v(\mathbf{x})\} = \tilde{h}(v(\mathbf{x})).$$

We now use the compactness in S_N of the level sets of v to show that $\tilde{h}$ is continuous at zero. Let $(r_k)_{k\in\mathbb{N}}$ be an arbitrary increasing sequence tending to zero. Choose $\mathbf{x}_k \in \mathcal{X}_{r_k} v$ such that $\tilde{h}(r_k) = u(\mathbf{x}_k)$. This is possible because u is continuous and the $\mathcal{X}_{r_k} v$ are compact and nonempty. Since $\tilde{h}$ is nondecreasing, $\tilde{h}(r_k) \to \tilde{h}^-(0)$.

Let $\mathbf{x}$ be any accumulation point of the set $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$. Since the $\mathcal{X}_{r_k} v$ are compact, all the accumulation points of the set $\{x_k\}_{k\in\mathbb{N}}$ are contained in $\mathcal{X}_0 v = \bigcap_{k\in\mathbb{N}} \mathcal{X}_{r_k} v$. This means that $u(\mathbf{x}) = 0$. But $\lim u(\mathbf{x}_k) = u(\mathbf{x})$ by the continuity of u, and we conclude that $\tilde{h}^-(0) = 0$. At this point $\tilde{h}$ satisfies the announced requirements for h, except that it is not always continuous for all $r < 0$. This is easily fixed by choosing a continuous nondecreasing function h such that $\tilde{h} \geq h$ and $h(0) = 0$. One way to do this is to take $h(r) = (1/|r|) \int_{2r}^{r} \tilde{h}(s)\,\mathrm{d}s$ for $r < 0$. Then $u(\mathbf{x}) \geq \tilde{h}(v(\mathbf{x})) \geq h(v(\mathbf{x}))$ as announced. □

Exercise 13.8. Prove that $h(r) = (1/|r|) \int_{2r}^{r} \tilde{h}(s)\,\mathrm{d}s$ is indeed nondecreasing and continuous for $r \leq 0$ and that $\tilde{h} \geq h$. Find examples of functions u and v defined on S_1 for which $\tilde{h}$ is not continuous. ■

Definition 13.18 (and proposition (Evans-Spruck)). [1] *Given a contrast invariant monotone operator T on $\mathcal{F}$, we call level set extension of T the set operator defined in the following way : for any $X \in \mathcal{L}$, take $u \leq 0$ such that $\mathcal{X}_0 u = X$ and set*

$$\mathcal{T}(X) = \mathcal{X}_0 T(u).$$

Then $\mathcal{T}(X)$ does not depend upon the particular choice of u.

Proof. The proof follows directly from Lemma 13.17: Take u and $v \in \mathcal{F}$ such that $u \leq 0$, $v \leq 0$, and $\mathcal{X}_0 u = \mathcal{X}_0 v$. Let h be a contrast change such that $h(0) = 0$ and $u \geq h(v)$. Since T is monotone and contrast invariant one has by Lemma 13.1.2 $Tu \leq 0$, and $Tu \geq Th(v) = h(Tv)$. Using the fact that $h(0) = 0$, we obtain that $Tv(\mathbf{x}) = 0$ implies that $Tu(\mathbf{x}) = 0$. By interchanging the roles of u and v, $Tu(\mathbf{x}) = 0$ implies that $Tv(\mathbf{x}) = 0$. We conclude that $\mathcal{X}_0 Tu = \mathcal{X}_0 Tv$. □

Exercise 13.9. Definition 13.18 would'nt be complete if we did not prove that for any $X \in \mathcal{L}$ we can find $u \leq 0$ in $\mathcal{F}$ such that $\mathcal{X}_0 u = X$. Hint: Since S_N is the unit sphere in $\mathbb{R}^{N+1}$, one can endow it with the euclidian distance d in $\mathbb{R}^{N+1}$. Use the distance function $d(\mathbf{x}, X)$ to define u. This distance function is continuous: see Exercise 13.18. ■

Theorem 13.19 (Evans–Spruck). *Let T be a contrast-invariant monotone operator on $\mathcal{F}$ and $\mathcal{T}$ its level set extension on $\mathcal{L}$. Then $\mathcal{T}$ is monotone, T and $\mathcal{T}$ satisfy the commutation with thresholds $\mathcal{T}\mathcal{X}_\lambda u = \mathcal{X}_\lambda T u$ for all $\lambda \in \mathbb{R}$, T is the stack filter associated with $\mathcal{T}$ and $\mathcal{T}$ is upper semicontinuous on $\mathcal{L}$. In addition, if T is standard, then so is $\mathcal{T}$.*

Proof. **Commutation with thresholds:** Given u and λ, let g be a continuous contrast change such that $g(s) = \min(s, \lambda) - \lambda$ on the range of u, which is a compact interval of $\mathbb{R}$. We then have $\mathcal{X}_0 g(u) = \mathcal{X}_\lambda u$. Using this relation, the level set extension and the contrast invariance of T,

$$\mathcal{T}(\mathcal{X}_\lambda u) = \mathcal{T}(\mathcal{X}_0 g(u)) = \mathcal{X}_0(T(g(u))) = \mathcal{X}_0(g(Tu)) = \mathcal{X}_\lambda(Tu).$$

[1] What we are doing here is related to the scheme originally introduced by Osher and Sethian as a numerical method for front propagation [199]. We briefly described their work in the Introduction (see page 26).

Proof of the stack filter property: This is an immediate consequence of the superposition principle and the commutation with thresholds :

$$Tu(\mathbf{x}) = \sup\{\lambda \mid \mathbf{x} \in \mathcal{X}_\lambda Tu\} = \sup\{\lambda \mid \mathbf{x} \in \mathcal{T}(\mathcal{X}_\lambda u)\}.$$

Proof that $\mathcal{T}$ is upper semicontinuous on $\mathcal{L}$: By the result of Exercise 13.5, it is enough to consider a sequence $(X_n)_{n\geq 1}$ in $\mathcal{L}$ such that $X_{n+1} \subset X_n^\circ$. By Lemma 13.20 below there is a function $u \in \mathcal{F}$ such that $\mathcal{X}_{1-\frac{1}{n}}u = X_n$ and $\mathcal{X}_1 u = \bigcap_n X_n$. Finally, using twice the just proven commutation of thresholds,

$$\mathcal{T}(\bigcap_n X_n) = \mathcal{T}(\mathcal{X}_1 u) = \mathcal{X}_1(Tu) = \bigcap_n \mathcal{X}_{1-\frac{1}{n}} Tu = \bigcap_n \mathcal{T}(\mathcal{X}_{1-\frac{1}{n}} u) = \bigcap_n \mathcal{T}(X_n).$$

Proof that $\mathcal{T}$ is standard if T is: Recall that T is standard if $Tu(\infty) = u(\infty)$. By using the commutation with thresholds, all of the standard properties for $\mathcal{T}$ are straightforward. For instance, taking some $u \in \mathcal{F}$,

$$\mathcal{T}(\emptyset) = \mathcal{T}(\mathcal{X}_{\max u+1} u) = \mathcal{X}_{\max u+1} Tu = \emptyset.$$

Indeed, by the monotonicity, the contrast invariance, and Lemma 13.1.2, $u \leq C \Rightarrow Tu \leq C$.
In the same way, let $X \in \mathcal{L}$ and u a function such that $\mathcal{X}_0 u = X$. If X is bounded, then $u(\infty) < 0$, so that $Tu(\infty) = u(\infty) < 0$. Thus $\mathcal{T}(X) = \mathcal{X}_0 Tu$ is bounded. If $X^c = \{\mathbf{x} \mid u(\mathbf{x}) < 0\}$ is bounded, then $Tu(\infty) = u(\infty) \geq 0$. Thus $\mathcal{T}(X)^c = (\mathcal{X}_0 Tu)^c$ is bounded. Finally by the commutation with thresholds,

$$\infty \in X \Leftrightarrow u(\infty) \geq 0 \Leftrightarrow Tu(\infty) \geq 0 \Leftrightarrow \infty \in \mathcal{X}_0(Tu) = \mathcal{T}(X).$$

□

Exercise 13.10. Prove that the level set extension $\mathcal{T}$ is monotone. The argument is not given in the above proof. ■

Lemma 13.20. *Let $(X_n)_{n\geq 1}$ be a sequence in $\mathcal{L}$ such that $X_{n+1} \subset X_n^\circ$. There is a function $u \in \mathcal{F}$ such that $\mathcal{X}_{1-\frac{1}{n}}u = X_n$ for $n \geq 1$ and $\mathcal{X}_1 u = \bigcap_{n\geq 1} X_n$.*

Proof. Let us use the euclidian distance d of $\mathbb{R}^{N+1}$ restricted to S_N considered as a subset of $\mathbb{R}^{N+1}$. Set $u(\mathbf{x}) = 1$ if $\mathbf{x} \in \bigcap_n X_n$,

$$u(\mathbf{x}) = (1 - \frac{1}{n})\frac{d(\mathbf{x}, X_{n+1})}{d(\mathbf{x}, X_n^c) + d(\mathbf{x}, X_{n+1})} + (1 - \frac{1}{n+1})\frac{d(\mathbf{x}, X_n^c)}{d(\mathbf{x}, X_n^c) + d(\mathbf{x}, X_{n+1})}$$

for $\mathbf{x} \in X_n \setminus X_{n+1}$ and $n \geq 1$, $u(\mathbf{x}) = -\sup(-1, -d(\mathbf{x}, X_1))$ if $\mathbf{x} \notin X_1$. It is easily checked that u belongs in $\mathcal{F}$ and satisfies the announced properties. □

13.4 A first application: the extrema killer

This section is devoted to the study of operators that remove "peaks," or extreme values, from an image. Such peaks are often created by impulse noise, that is, local destruction of pixel values and their replacement by a random value. Old movies present this kind of noise and it also occurs by transmission failure in satellite imaging. The operators we study are called *area opening*, or *extrema killer* operators, and they have been shown to be very effective at removing this kind of noise. The action of these operators is illustrated in Figures 13.1 and 13.2.

The following definitions are standard, but we include them here for completeness.

Definition 13.21. *Consider a closed subset X of S_N. X is disconnected if it can be written as $X = (A \cap X) \cup (B \cap X)$, where A and B are disjoint open sets and both $A \cap X$ and $B \cap X$ are not empty. X is connected if it is not disconnected. The connected component of $\mathbf{x}$ in X, denoted by $cc(\mathbf{x}, X)$, is the maximal connected subset of X that contains $\mathbf{x}$.*

We wish to define a denoising operator on $\mathcal{L}$; since some sets therein contain ∞, we need an extension of the Lebesgue measure on $\mathbb{R}^N$ to S_N. This is immediately fixed by setting $\text{meas}(\{\infty\}) = +\infty$. The only property of this extended measure that we need to check is following:

Lemma 13.22. *if Y_n is a nonincreasing sequence of compact sets of S_N, then $meas(\cap_n Y_n) = \lim_n meas(Y_n)$.*

Proof. If the compact sets Y_n do not contain ∞ for n large enough, then they are bounded in $\mathbb{R}^N$ for n large and the result just follows from Lebesgue theorem. If instead the sets Y_n all contain ∞, then $\cap_n Y_n$ contains it too and all sets have infinite measure. □

Definition 13.23. *Let $a > 0$ a scale parameter and denote for every $X \in \mathcal{L}$ by X_i its connected components, so that $X = \bigcup_i X_i$. We call small component killer the operator on $\mathcal{L}$ which removes from X all connected components with area stricly less than a :*

$$\mathcal{T}_a X = \bigcup_{meas\,(X_i) \geq a} X_i. \tag{13.8}$$

Theoretically, X can have an uncountable number of components; take, for example, the Cantor set. However, X can have only a countable number of components with positive measure. The assumption $\text{meas}(\{\infty\}) = +\infty$ implies that all connected components of X containing ∞ stay in $\mathcal{T}_a X$. We are going to prove that the small component killer is upper semicontinuous and this uses some elementary topological lemmas.

Lemma 13.24. *Consider an arbitrary nonincreasing sequence of nonempty compact sets $(Y_n)_{n \in \mathbb{N}}$ of S_N and its limit $Y = \bigcap_{n \in \mathbb{N}} Y_n$. Then Y is not empty and compact. In addition, for any open set Z that contains Y, there is an index n_0 such that $Y_n \subset Z$ for all $n \geq n_0$.*

Proof. The first property is a classical property of compact sets. Assume by contradiction that the second property is not true. Then $Y_n \cap (S_N \setminus Z) \neq \emptyset$ infinitely often. This implies that $(Y_n \cap (S_N \setminus Z))_{n\in\mathbb{N}}$ is a nonincreasing sequence of nonempty compact sets. But this means that $Y \cap (S_N \setminus Z) \neq \emptyset$, which is a contradiction. □

Lemma 13.25. *Let $(Y_n)_{n\in\mathbb{N}}$ be a nonincreasing sequence of nonempty compact subsets of S_N and consider the intersection $Y = \bigcap_{n\in\mathbb{N}} Y_n$. If the Y_n are connected, then Y is connected.*

Proof. We know that Y is not empty and compact. Suppose, by contradiction, that Y is not connected. Then we can represent Y by $Y = (Y \cap Z_1) \cup (Y \cap Z_2)$, where Z_1 and Z_2 are disjoint open sets, $Y \cap Z_1 \neq \emptyset$, and $Y \cap Z_1 \neq \emptyset$. Since $Y \subset Z_1 \cup Z_2$, by Lemma 13.24 there exists an n_0 such that $Y_n \subset Z_1 \cup Z_2$ for all $n \geq n_0$, and for these n we have

$$Y_n = Y_n \cap (Z_1 \cup Z_2) = (Y_n \cap Z_1) \cup (Y_n \cap Z_2).$$

Furthermore, $Y_n \cap Z_1 \neq \emptyset$ and $Y_n \cap Z_1 \neq \emptyset$. This contradicts the fact that the Y_n are connected. □

Exercise 13.11. Show that $\mathcal{T}_a$ is idempotent: ${\mathcal{T}_a}^2 X = \mathcal{T}_a X$ and that it is a contraction mapping: $\mathcal{T}_a X \subset X$. ■

With the extrema killer we have a prime example of a theory that begins with a set operator $\mathcal{T}_a$ defined on $\mathcal{L}$.

Lemma 13.26. *The small component killer $\mathcal{T}_a$ is upper semicontinuous on $\mathcal{L}$.*

Proof. We first prove that $\mathcal{T}_a$ is monotone. Assume $X \subset Y$. Then for every $\mathbf{x} \in X$, $cc(\mathbf{x}, X) \subset cc(\mathbf{x}, Y)$. If $\text{meas}\,(cc(\mathbf{x}, X)) \geq a$, then $\text{meas}\,(cc(\mathbf{x}, Y)) \geq a$, and we conclude that $\mathcal{T}_a X \subset \mathcal{T}_a Y$. Now let $(X_n)_n$ be any nonincreasing sequence of nonempty compact sets and $X = \cap_n X_n$. We wish to show that $\mathcal{T}_a X = \bigcap_n \mathcal{T}_a X_n$. By monotonicity of $\mathcal{T}_a$,

$$\mathcal{T}_a X \subset \bigcap_n \mathcal{T}_a(X_n).$$

Let us show the converse inclusion. Let $\mathbf{x} \in \cap_n \mathcal{T}_a(X_n)$. Then $Y_n := cc(x, X_n)$ has measure larger than a for all n. In addition if $m < n$ then $Y_n \subset Y_m$. By Lemmas 13.24 and 13.25, $Y := \cap_n Y_n$ is a connected compact set that contains $\mathbf{x}$. In addition by Lemma 13.22, $\text{measure}(Y) = \lim_n \text{measure}(Y_n) \geq a$. Since $Y = \cap_n Y_n \subset \cap_n X_n = X$, we have $cc(\mathbf{x}, X) \supseteq Y$ and therefore $\mathbf{x} \in \mathcal{T}_a(X)$. □

We can now build a stack filter from $\mathcal{T}_a$.

Definition 13.27 (and proposition). *The stack filter T_a of $\mathcal{T}_a$ is called a maxima killer. $\mathcal{T}_a$ and T_a satisfy the commutation with thresholds. As a consequence, no connected component of a level set of $T_a u$ has measure less than a. Furthermore, T_a is standard monotone, translation and contrast invariant from $\mathcal{F}$ into $\mathcal{F}$.*

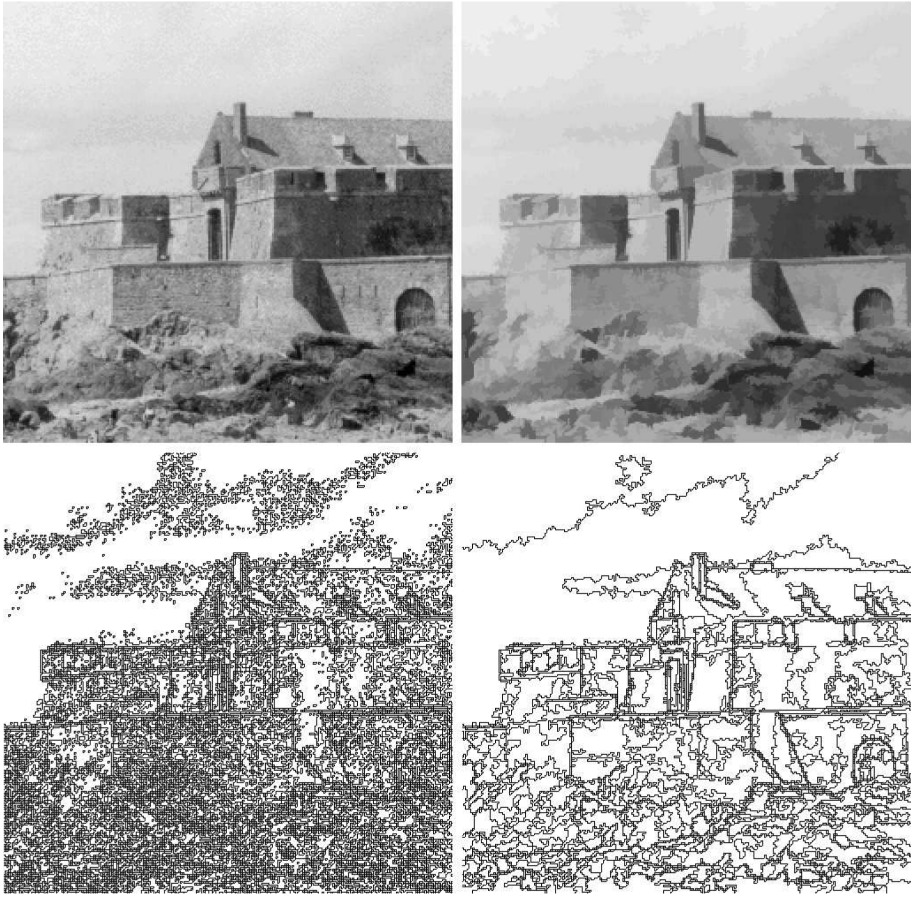

Figure 13.1: Extrema killer: maxima killer followed by minima killer. The extrema killer removes all connected components of upper and lower level sets with area less than some threshold, which here equals 20 pixels. Notice how texture disappears in the second image. All other features seem preserved. On the second row, we see for both the original and the processed image the level lines at 16 equally spaced levels. The level lines on the right hand side are a subset of the level lines of the left hand. All level lines surrounding extremal regions with area smaller than 20 have been removed and the other ones are untouched.

Proof. We just have to check that all assumptions of Theorem 13.16 are satisfied. $\mathcal{T}_a$ is obviously translation invariant, monotone and is upper semicontinuous by Lemma 13.26. It satisfies $\mathcal{T}_a(\emptyset) = \emptyset$, $\mathcal{T}_a(S_N) = S_N$. $\mathcal{T}_a(E)$ is compact if E is. Indeed, it is the union of a finite set of compact connected components. If E is bounded in $\mathbb{R}^N$, then so is $\mathcal{T}_a E \subset E$. $(\mathcal{T}_a E)^c$ is bounded in S_N if E^c is. Indeed, if E^c is bounded, then E has a connected component Y containing $B(0,R)^c$ for some $R > 0$. This connected component has infinite measure. Then $\mathcal{T}_a(E)$ still contains Y and $\mathcal{T}_a(E)^c$ is contained in $B(0,R)$. By construction, ∞ belongs to $\mathcal{T}_a X$ if and only if it belongs to X. Thus, $\mathcal{T}_a$ is standard monotone. □

A maxima killer T_a cuts off the maxima of continuous functions, but it does nothing for the minima. We can immediately define a minima killer T_a^- as the dual operator of T_a,

$$T_a^- u = -T_a(-u).$$

A good denoising process is to alternate T_a and T_a^-, as illustrated in Figures 13.1 and 13.2 . We note, however, that T_a and T_a^- do not necessarily commute, as is shown in Exercise 13.17.

13.5 Exercises

Exercise 13.12. Let $g : \mathbb{R} \to \mathbb{R}$ be a contrast change. Construct increasing contrast changes g_n and h_n such that $g_n(s) \to g(s)$, $h_n(s) \to g(s)$ for all s and $g_n \leq g \leq h_n$. Hint : define first an increasing continuous function $f(s)$ on $\mathbb{R}$ such that $f(-\infty) = 0$ and $f(+\infty) = \frac{1}{n}$. ■

Exercise 13.13. Let $u : \mathbb{R}^N \to \mathbb{R}$. Show that $\tau_{\mathbf{x}} \mathcal{X}_\lambda u = \mathcal{X}_\lambda \tau_{\mathbf{x}} u$, $\mathbf{x} \in \mathbb{R}^N$. ■

Exercise 13.14. Prove that a translation invariant operator $\mathcal{T}$ from $\mathcal{L}$ to $\mathcal{L}$ satisfies one of the three possibilities : $\mathcal{T}(\{\infty\}) = \{\infty\}$, $\mathcal{T}(\{\infty\}) = S_N$ or $\mathcal{T}(\{\infty\}) = \emptyset$. ■

Exercise 13.15. Let T be a translation invariant standard monotone operator on $\mathcal{F}$. Prove the following statements:

(i) $Tu = c$ for every constant function $u : S_N \to c$.

(ii) $u \geq c$ implies $Tu \geq c$, and $u \leq c$ implies $Tu \leq c$.

(iii) If in addition T commutes with the addition of constants, $\sup_{\mathbf{x}\in\mathbb{R}^N} |Tu(\mathbf{x}) - Tv(\mathbf{x})| \leq \sup_{\mathbf{x}\in\mathbb{R}^N} |u(\mathbf{x}) - v(\mathbf{x})|$.
(Hint: Write $-\sup|u(\mathbf{x}) - v(\mathbf{x})| \leq u(\mathbf{x}) - v(\mathbf{x}) \leq \sup|u(\mathbf{x}) - v(\mathbf{x})|$.)

■

Exercise 13.16.

1) In dimension 1, consider the set operator defined on $\mathcal{L}$ by $\mathcal{T}X = [\inf X, \infty]$ if $\inf(X \cap \mathbb{R}) \in \mathbb{R}$, $\mathcal{T}X = S_1$ if $\inf(X \cap \mathbb{R}) = -\infty$, $\mathcal{T}(\{\infty\}) = \{\infty\}$, $\mathcal{T}(\emptyset) = \emptyset$. Check that $\mathcal{T}$ satisfies all assumptions of Theorem 13.16 except one. Compute the stack filter associated with $\mathcal{T}$ and show that it satisfies all conclusions of the mentioned theorem except one : Tu does not belong to $\mathcal{F}$ and more specifically $Tu(\mathbf{x})$ is not continuous at ∞.

2) Consider the function operator on $\mathcal{F}$, $Tu(\mathbf{x}) = \sup_{\mathbf{x}\in S_N} u(\mathbf{x})$. Check that T is monotone, contrast invariant, and sends $\mathcal{F}$ to $\mathcal{F}$. Compute the level set extension $\mathcal{T}$ of T. ■

Exercise 13.17. Let $N = 1$ and take $u(x) = \sin x$ for $|x| \leq 8\pi$, $u(x) = 0$ otherwise. Compute $T_a u$ and $T_a^- u$ and show that they commute on u if $a \leq \pi$ and do not commute if $a > \pi$. Following the same idea, construct a function $u \in \mathcal{F}$ in dimension two such that $T_a T_a^- u \neq T_a^- T_a u$. ■

Exercise 13.18. Let X be a closed subset of a metric space endowed with a distance d and consider the distance function to X,

$$d(\mathbf{y}) = d(\mathbf{y}, X) = \inf_{\mathbf{x}\in X} d(\mathbf{x}, \mathbf{y}).$$

Show that d is 1-Lipschitz, that is, $|d(\mathbf{x}, X) - d(\mathbf{y}, X)| \leq d(\mathbf{x}, \mathbf{y})$. ■

Exercise 13.19. In the following questions, we explain the necessity of the assumptions $\mathcal{T}(\emptyset) = \emptyset$, $\mathcal{T}(S_N) = S_N$ for defining function monotone operators from $\mathcal{F}$ to $\mathcal{F}$.

1) Set $\mathcal{T}(X) = X_0$ for all $X \in \mathcal{L}$, where $X_0 \neq \emptyset$ is a fixed set. Check that the associated stack filter satisfies $Tu(\mathbf{x}) = +\infty$ if $\mathbf{x} \in X_0$, $Tu(\mathbf{x}) = -\infty$ otherwise.

2) Let $\mathcal{T}$ be a monotone set operator, without further assumption. Show that its associated stack filter T is, however, monotone and commutes with all contrast changes. (We extend each contrast change g by setting $g(\pm\infty) = \pm\infty$.) ■

Exercise 13.20. Take an operator $\mathcal{T}$ satisfying the same assumptions as in Theorem 13.16, but defined on $\mathcal{M}$ and apply the arguments of the proof of Theorem 13.16. Check that the stack filter associated with $\mathcal{T}$ is a contrast invariant, translation invariant monotone operator on the set of all bounded measurable functions, $L^\infty(\mathbb{R}^N)$. If in addition $\mathcal{T}$ is upper semicontinuous on $\mathcal{M}$, then the commutation with thresholds holds. ■

Exercise 13.21. The upper semicontinuity is necessary to ensure that a monotone set operator defines a function operator such that the commutation with thresholds $\mathcal{X}_\lambda(Tu) = \mathcal{T}(\mathcal{X}_\lambda(u))$ holds for every λ. Let us choose for example the following set operator $\mathcal{T}$,

$$\mathcal{T}(X) = X \text{ if } \mathrm{meas}(X) > a \text{ and } \mathcal{T}(X) = \emptyset \text{ otherwise }.$$

(We use the Lebesgue measure on $\mathbb{R}^N$, with the completion $\mathrm{meas}(\{\infty\}) = +\infty$)

1) Prove that $\mathcal{T}$ is standard monotone.

2) Let u be the function from S_1 into S_1 defined by $u(x) = max(-|x|, -2a)$ for some $a > 0$, with $u(\infty) = -2a$. Check that u belongs to $\mathcal{F}$. Then, applying the stack filter T of $\mathcal{T}$, check that

$$T(u)(x) = \sup\{\lambda, x \in \mathcal{T}(\mathcal{X}_\lambda u)\} = max(\min(-|x|, -a/2), -2a).$$

3) Deduce that $\mathcal{X}_{-a/2}T(u) = [-a/2, a/2]$, $\mathcal{X}_{-a/2}u = [-a/2, a/2]$ and therefore

$$\mathcal{T}(\mathcal{X}_{-a/2}u) = \emptyset \neq \mathcal{X}_{-a/2}T(u),$$

which means that T does not commute with thresholds. ■

Exercise 13.22. Like in the preceding exercise, we consider here contrast invariant operators defined on all measurable bounded functions of $\mathbb{R}^N$. The aim of the exercise is to show that such operators send images with finite range into images with finite range. More precisely, denote by $R(u) = u(\mathbb{R}^N)$ the range of u. Then we shall prove that for every u, $R(Tu) \subset \overline{Ru}$. In particular, if $R(u)$ is finite, then the range of Tu is a finite subset of Ru. If u is binary, Tu is, etc. This shows that contrast invariant operators preserve sharp contrasts. A binary image is transformed into a binary image. So contrast invariant operators create no blur, as opposed to linear operators, which always create new intermediate grey levels.

1) Consider

$$g(s) = s + \frac{1}{2}d(s, \overline{Ru})$$

where $d(s, X)$ denotes the distance from s to X, that is, $d(s, X) = \inf_{x \in X} |s - x|$. Show that g is a contrast change satisfying $g(s) = s$ for $s \in \overline{Ru}$ and $g(s) > s$ otherwise.

2) Check that $g(s) = s$ if and only if $s \in \overline{Ru}$. In particular, $g(u) = u$. Deduce from this and from the contrast invariance of T that for every $\mathbf{x} \in \mathbb{R}^N$, $Tu(\mathbf{x})$ is a fixed point of g. Conclude. ■

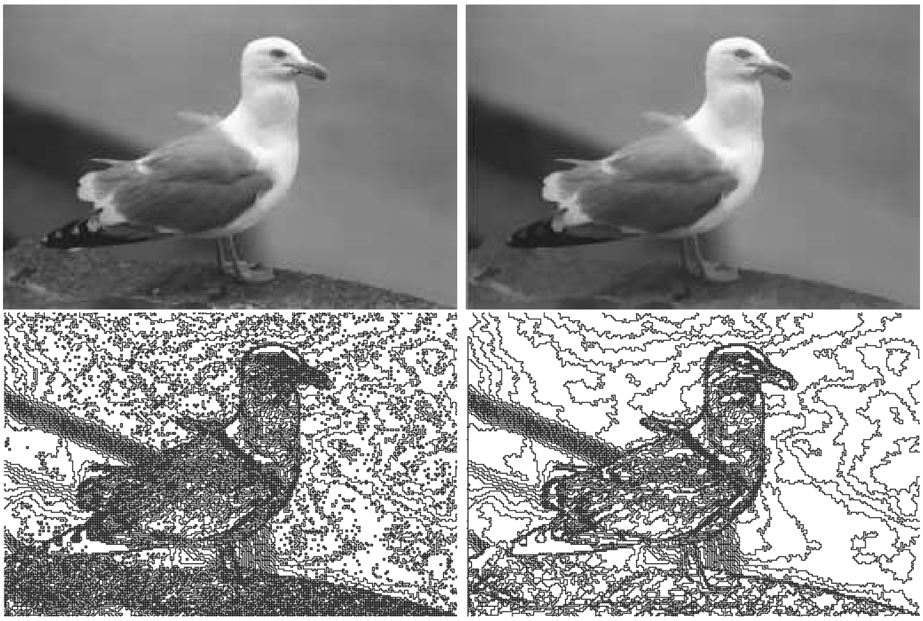

Figure 13.2: Extrema killer: maxima killer followed by minima killer. Above, left: original image. Above, right: image after extrema killer removed connected components of 20 pixels or less. Below: level lines (levels of multiples of 16) of the image before and after the application of the extrema killer.

13.6 Comments and references

Contrast invariance and stack filters. Image operators that commute with thresholds have been popular because, among other reasons, they are easily implemented in hardware (VLSI). This led to very simple patents being awarded in signal and image processing as late as 1987 [59]. These operators have been given four different names, although operators are equivalent: *stack filters* [37, 103, 259]; *threshold decomposition* [108]; *rank filters* [52, 129, 261]; and *order filters* [241]. The best known of these are the sup, inf, and median operators. The implementation of the last named has received much attention because of its remarkable denoising properties [82, 197, 266].

Maragos and Shafer [160, 161] and Maragos and Ziff [162] introduced the functional notation and established the link between stack filters and the Matheron formalism in "flat" mathematical morphology. The complete equivalence between contrast-invariant operators and stack filters, as developed in this chapter, does not seem to have appeared elsewhere; at least we do not know of other references. A related classification of rank filters with elegant and useful generalizations to the so-called *neighborhood filters* can be found in [129].

The extrema killer. The extrema killer is probably the most efficient denoising filter for images degraded by impulse noise, which is manifest by small spots. In spite of its simplicity, this filter has only recently seen much use. This is undoubtedly due to the nontrivial computations involved in searching for the connected components of upper and lower level sets. The first reference to the extrema killer that we know is [55]. The filter in its generality was defined by Vincent in [253]. This definition fits into the general theory of connected filters developed by Salembier and Serra [219]. Masnou defined a variant called the *grain filter* that is both contrast invariant and invariant under reverse contrast changes [166]. Monasse and Guichard developed a fast implementation of this filter based on the so-called *fast level set transform* [181].

We will develop in Chapter 19 a theory of scale space that is based on a family of image smoothing operators T_t, where t is a scale parameter. We note here that the family $(T_a)_{a\in\mathbb{R}^+}$ of extrema killers does not constitute a scale space because it does not satisfy one of the conditions, namely, what we call the local comparison principle. That this is so, is the content of Exercise 19.1.

Chapter 14

Sup-Inf Operators

The main contents of this chapter are two representation theorems: one for translation-invariant monotone set operators and one for functions operators that are monotone, contrast invariant, and translation invariant. If T is a function operator satisfying these three conditions, then it has a "sup-inf" representation of the form

$$Tu(\mathbf{x}) = \sup_{B \in \mathcal{B}} \inf_{\mathbf{y} \in B} u(\mathbf{x}+\mathbf{y}),$$

where $\mathcal{B}$ is a family of subsets of $\mathcal{M}(S_N)$, the set of all measurable subsets of S_N. This theorem is a nonlinear analogue of the Riesz theorem that states that a continuous linear translation-invariant operator from $L^2(\mathbb{R}^N)$ to $C^0(\mathbb{R}^N)$ can be represented as a convolution

$$Tu(\mathbf{x}) = \int_{\mathbb{R}^N} u(\mathbf{x}-\mathbf{y})k(\mathbf{y})\,\mathrm{d}\mathbf{y}.$$

In this case, the kernel $k \in L^2(\mathbb{R}^N)$ is called the impulse response. In the same way, $\mathcal{B}$ is an impulse response for the nonlinear operator.

14.1 Translation-invariant monotone set operators

Recall that a set of $\mathcal{M}$ can contain ∞. We have specified that $\mathbf{x}+\infty = \infty$ for every $\mathbf{x} \in S_N$. As a consequence, for any subset B of S_N, $\infty + B = \{\infty\}$. In this chapter we shall associate with any nonempty subset $\mathcal{B}$ of $\mathcal{M}$ a set operator defined by

$$\mathcal{T}X = \{\mathbf{x} \in S_N \mid \mathbf{x}+B \subset X \text{ for some } B \in \mathcal{B}\}. \tag{14.1}$$

Definition 14.1. *We say that a subset $\mathcal{B}$ of $\mathcal{M}$ is standard if it is not empty and if its associated operator satisfies*

(i) $\forall R > 0,\ \exists R' > 0,\ \mathcal{T}(B(0,R)) \subset B(0,R')$;

(ii) $\forall R > 0,\ \exists R' > 0,\ B(0,R')^c \subset \mathcal{T}(B(0,R)^c)$.

Exercise 14.1. Conditions (i) and (ii) on $\mathcal{B}$ are easily satisfied. Check that Condition (i) is equivalent to

$$\forall R > 0,\ \exists C > 0,\ (B \in \mathcal{B},\ \text{and diameter}(B) \le R) \Rightarrow B \subset B(0, C).$$

Check that this condition is achieved (e.g.) if all elements of $\mathcal{B}$ contain 0. Check that Condition (ii) is achieved if $\mathcal{B}$ contains at least one bounded element B. ■

Exercise 14.2. Show that if $\mathcal{B}$ contains $\emptyset$, then $\mathcal{B}$ is not standard. ■

Theorem 14.2 (Matheron). *Let $\mathcal{T}$ be a translation-invariant and standard monotone set operator. Consider the subset of $\mathcal{D}(\mathcal{T})$,*

$$\mathcal{B} = \{B \in \mathcal{D}(\mathcal{T}) \mid 0 \in \mathcal{T}B\} \tag{14.2}$$

Then $\mathcal{B}$ is standard and the operator $\mathcal{T}$ is associated with $\mathcal{B}$ by (14.1)*. Conversely, if $\mathcal{B}$ is any standard subset of $\mathcal{M}$, then* (14.1) *defines a translation-invariant standard monotone set operator on $\mathcal{M}$.*

Definition 14.3. *In Mathematical Morphology, a set $\mathcal{B}$ such that* (14.1) *holds is called a set of* structuring elements of $\mathcal{T}$ *and $\mathcal{B} = \{X \in \mathcal{D}(\mathcal{T}) \mid 0 \in \mathcal{T}X\}$ is called* the canonical set of structuring elements of $\mathcal{T}$.

Proof of Theorem 14.2.

Proof of (14.1).
Let $\mathcal{B} = \{X \in \mathcal{D}(\mathcal{T}) \mid 0 \in \mathcal{T}X\}$. Then for any $\mathbf{x} \in \mathbb{R}^N$,

$$\begin{aligned} \mathbf{x} \in \mathcal{T}X &\overset{(1)}{\Longleftrightarrow} 0 \in \mathcal{T}X - \mathbf{x} \overset{(2)}{\Longleftrightarrow} 0 \in \mathcal{T}(X - \mathbf{x}) \overset{(3)}{\Longleftrightarrow} X - \mathbf{x} \in \mathcal{B} \\ &\overset{(4)}{\Longleftrightarrow} X - \mathbf{x} = B \text{ for some } B \in \mathcal{B} \overset{(5)}{\Longleftrightarrow} \mathbf{x} + B \subset X \text{ for some } B \in \mathcal{B}. \end{aligned}$$

The equivalence (2) follows from the translation invariance of $\mathcal{T}X$; (3) is just the definition of $\mathcal{B}$; and (4) is a restatement of (3). The implication from left to right in (5) is obvious. The implication from right to left in (5) is the point where the monotonicity of $\mathcal{T}$ is used: Since $B \subset X - \mathbf{x}$, it follows from the monotonicity of $\mathcal{T}$ that $X - \mathbf{x} \in \mathcal{B}$.
Let now $\mathbf{x} = \infty$. Since $\mathcal{T}$ is standard, $\mathcal{B}$ is not empty (it contains S_N because $\mathcal{T}(S_N) = S_N$) and we have

$$\infty \in \mathcal{T}X \Leftrightarrow \infty \in X \Leftrightarrow \exists B \in \mathcal{B},\ \infty + B \subset X,$$

because $\infty + S_N = \{\infty\}$.

Proof that $\mathcal{B}$ is standard if $\mathcal{T}$ is standard monotone.
Since $\mathcal{T}(S_N) = S_N$, $\mathcal{B}$ contains S_N and is therefore not empty. The other properties are straightforward.

Proof that (14.1) **defines a standard monotone set operator if $\mathcal{B}$ is standard.**
Using (14.1), it is a straightforward calculation to check that $\mathcal{T}$ is monotone and translation invariant, and that $\mathcal{T}(S_N) = S_N$, $\mathcal{T}(\emptyset) = \emptyset$. The equivalence $\infty \in \mathcal{T}X$ if and only if $\infty \in X$ follows from the fact that $\mathcal{B}$ is not empty. $\mathcal{T}$ sends bounded sets onto bounded sets and complementary sets of bounded sets onto complementary sets of bounded sets by definition of a standard set $\mathcal{B}$. □

Exercise 14.3. Check that if $\mathcal{T}$ is standard monotone, then its canonical set of structuring elements satisfies (ii). ■

$\mathcal{B}_0 = \{X \mid 0 \in \mathcal{T}X\}$ is not the only set that can be used to represent $\mathcal{T}$. A monotone operator $\mathcal{T}$ can have many such sets and here is their characterization.

Proposition 14.4. *Let $\mathcal{T}$ be a translation invariant standard monotone set operator and let $\mathcal{B}_0$ its canonical set of structuring elements. Then $\mathcal{B}_1$ is another standard set of structuring elements for $\mathcal{T}$ if and only if it satisfies*

(i) $\mathcal{B}_1 \subset \mathcal{B}_0$,

(ii) for all $B_0 \in \mathcal{B}_0$, there is $B_1 \in \mathcal{B}_1$ such that $B_1 \subset B_0$.

Proof. Assume that $\mathcal{T}$ is obtained from some set $\mathcal{B}_1$ by (14.1) and let $\mathcal{B}_0$ be the canonical set of structuring elements of $\mathcal{T}$. Then for every $B_1 \in \mathcal{B}_1$, $\mathcal{T}B_1 = \{\mathbf{x} \mid \mathbf{x} + B \subset B_1 \text{ for some } B \in \mathcal{B}_1\}$. It follows that $0 \in \mathcal{T}B_1$ and therefore $B_1 \in \mathcal{B}_0$. Thus $\mathcal{B}_1 \subset \mathcal{B}_0$. In addition, if $B_0 \in \mathcal{B}_0$, then $0 \in \mathcal{T}B_0$, which means that $0 \in \{\mathbf{x} \mid \mathbf{x} + B_1 \subset B_0 \text{ for some } B_1 \in \mathcal{B}_1\}$, that is $B_1 \subset B_0$ for some $B_1 \in \mathcal{B}_1$.

Conversely, let $\mathcal{B}_1$ satisfy (i) and (ii) and let

$$\mathcal{T}_1 X = \{\mathbf{x} \mid \exists B_1 \in \mathcal{B}_1,\ \mathbf{x} + B_1 \subset X\}.$$

Using (i), one deduces that $\mathcal{T}_1 X \subset \mathcal{T}X$ for every X and using (ii) yields the converse inclusion. Thus $\mathcal{B}_1$ is a structuring set for $\mathcal{T}$. The fact that $\mathcal{B}_1$ is standard is an obvious check using (i) and (ii). □

14.2 The Sup-Inf form

Lemma 14.5. *Let $T : \mathcal{F} \to \mathcal{F}$ be a standard monotone function operator, $\mathcal{T}$ a standard monotone translation invariant set operator and $\mathcal{B}$ a set of structuring elements for $\mathcal{T}$. If T and $\mathcal{T}$ satisfy the commutation of thresholds $\mathcal{T}\mathcal{X}_\lambda u = \mathcal{X}_\lambda Tu$, then T has the "sup-inf" representation*

$$Tu(\mathbf{x}) = \sup_{B \in \mathcal{B}} \inf_{\mathbf{y} \in \mathbf{x}+B} u(\mathbf{y}). \tag{14.3}$$

Proof. For $u \in \mathcal{F}$, set $\tilde{T}u(\mathbf{x}) = \sup_{B\in\mathcal{B}} \inf_{\mathbf{y}\in\mathbf{x}+B} u(y)$. We shall derive the identity $T = \tilde{T}$ from the equivalence

$$\tilde{T}u(\mathbf{x}) \geq \lambda \Longleftrightarrow Tu(\mathbf{x}) \geq \lambda. \tag{14.4}$$

Assume first that $\mathbf{x} \in \mathbb{R}^N$. Then

$$Tu(\mathbf{x}) \geq \lambda \overset{(1)}{\Longleftrightarrow} Tu(\mathbf{x}) \geq \mu \text{ for all } \mu < \lambda \overset{(2)}{\Longleftrightarrow} \mathbf{x} \in \mathcal{X}_\mu Tu \text{ for all } \mu < \lambda$$

$$\overset{(3)}{\Longleftrightarrow} \mathbf{x} \in \mathcal{T}\mathcal{X}_\mu u \text{ for all } \mu < \lambda \overset{(4)}{\Longleftrightarrow} \exists B \in \mathcal{B},\ \mathbf{x} + B \subset \mathcal{X}_\mu u \text{ for all } \mu < \lambda$$

$$\overset{(5)}{\Longleftrightarrow} \text{There is a } B \in \mathcal{B} \text{ such that } \inf_{\mathbf{y}\in\mathbf{x}+B} u(\mathbf{y}) \geq \mu \text{ for all } \mu < \lambda$$

$$\overset{(6)}{\Longleftrightarrow} \sup_{B\in\mathcal{B}} \inf_{\mathbf{y}\in\mathbf{x}+B} u(\mathbf{y}) \geq \lambda \overset{(7)}{\Longleftrightarrow} \tilde{T}u(\mathbf{x}) \geq \lambda.$$

Equivalence (1) is just a statement about real numbers and (2) is the definition of a level set. It is at (3) that we replace $\mathcal{X}_\mu Tu$ with $\mathcal{T}\mathcal{X}_\mu u$. Equivalence (4) follows by the definition of $\mathcal{T}$ from $\mathcal{B}$ by (14.1). The equivalence (5) is the definition of the level set $\mathcal{X}_\mu u$. Equivalence (6) is another statement about real numbers, and (7) is the definition of $\tilde{T}$.
Assume now that $\mathbf{x} = \infty$. Since for all $B \in \mathcal{L}$, $\infty + B = \{\infty\}$, one obtains $\tilde{T}u(\infty) = u(\infty)$. By assumption $Tu(\infty) = u(\infty)$. This completes the proof of (14.3). □

From the preceding result, we can easily derive a general form for translation and contrast invariant standard monotone operators.

Theorem 14.6. *Let $T : \mathcal{F} \to \mathcal{F}$ be a translation and contrast invariant standard monotone operator. Then it has a "sup-inf" representation* (14.3) *with a standard set of structuring elements.*

Proof. By the level set extension (Theorem 13.19), T defines a unique standard monotone set operator $\mathcal{T} : \mathcal{L} \mapsto \mathcal{L}$. $\mathcal{T}$ is defined by the commutation of thresholds, $\mathcal{T}\mathcal{X}_\lambda u = \mathcal{X}_\lambda Tu$. By Lemma 14.5, the commutation with thresholds is enough to ensure that T has the sup-inf representation (14.3) for any set of structuring elements $\mathcal{B}$ of $\mathcal{T}$. □

Definition 14.7. *As a consequence of the preceding theorem, the canonical set of structuring elements of $\mathcal{T}$ will also be called* canonical set of structuring elements of T.

The next theorem closes the loop.

Theorem 14.8. *Given any standard subset $\mathcal{B}$ of $\mathcal{M}$, Equation* (14.3)*,*

$$Tu(\mathbf{x}) = \sup_{B\in\mathcal{B}} \inf_{\mathbf{y}\in\mathbf{x}+B} u(y),$$

defines a contrast and translation invariant standard monotone function operator from $\mathcal{F}$ into itself.

Proof. By Theorem 13.16, it is enough to prove that T is the stack filter of $\mathcal{T}$, the standard monotone set operator associated with $\mathcal{B}$. Let us call T' this stack filter and let us check that $Tu(\mathbf{x}) \geq \lambda \Leftrightarrow T'u(\mathbf{x}) \geq \lambda$.
we have $T'u = \sup\{\lambda,\ \mathbf{x} \in \mathcal{T}(\mathcal{X}_\lambda u)\}$. Thus by (14.1),

$$T'u(\mathbf{x}) \geq \lambda \Leftrightarrow \forall \mu < \lambda,\ \exists B,\ \mathbf{x} + B \subset \mathcal{X}_\mu u.$$

On the other hand,

$$\begin{aligned} Tu(\mathbf{x}) = \sup_{B\in\mathcal{B}} \inf_{\mathbf{y}\in\mathbf{x}+B} u \geq \lambda \quad &\Leftrightarrow \\ \forall\mu < \lambda,\ \exists B \in \mathcal{B}, \inf_{\mathbf{y}\in\mathbf{x}+B} u \geq \mu \quad &\Leftrightarrow \\ \forall\mu < \lambda, \exists B \in \mathcal{B},\ \mathbf{x} + B \subset \mathcal{X}_\mu u. & \end{aligned}$$

Thus, $T = T'$. □

We end this section by showing that sup-inf operators can also be represented as inf-sup operators,

$$Tu(\mathbf{x}) = \inf_{B \in \mathcal{B}} \sup_{\mathbf{y} \in \mathbf{x}+B} u(\mathbf{y}).$$

This is done, in the mathematical morphology terminology, by "duality". The dual operator of a function operator is defined by $\tilde{T}u = -T(-u)$. Notice that $\tilde{\tilde{T}} = T$.

Proposition 14.9. *If T is a standard monotone, translation invariant and contrast invariant operator, then so is $\tilde{T}$. As a consequence, T has a dual "inf-sup" form*

$$Tu = \inf_{B \in \tilde{\mathcal{B}}} \sup_{\mathbf{y} \in \mathbf{x}+B} u(\mathbf{y}),$$

where $\tilde{\mathcal{B}}$ is any set of structuring elements for $\tilde{T}$

Proof. Setting $\tilde{g}(s) = -g(-s)$, it is easily checked that $\tilde{g}$ is a contrast change if and only if g is. One has by the contrast invariance of T,

$$\tilde{T}(g(u)) = -T(-g(u)) = -T(\tilde{g}(-u)) = -\tilde{g}(T(-u)) = g(-T(-u)) = g(\tilde{T}u).$$

Thus, $\tilde{T}$ is contrast invariant. The standard monotonicity and translation invariance of $\tilde{T}$ are obvious. Finally, if we have $\tilde{T}u(\mathbf{x}) = \sup_{B \in \tilde{\mathcal{B}}} \inf_{\mathbf{y} \in \mathbf{x}+B} u(\mathbf{y})$, then

$$Tu = -\sup_{B \in \tilde{\mathcal{B}}} \inf_{\mathbf{y} \in \mathbf{x}+B} (-u(\mathbf{y})) = -\sup_{B \in \tilde{\mathcal{B}}} (-\sup_{\mathbf{y} \in \mathbf{x}+B} u(\mathbf{y})) = \inf_{B \in \tilde{\mathcal{B}}} \sup_{\mathbf{y} \in \mathbf{x}+B} u(\mathbf{y}).$$

□

Exercise 14.4. Check the standard monotonicity and translation invariance of $\tilde{T}$. ■

14.3 Locality and isotropy

For linear filters, locality can be defined by the fact that the convolution kernel is compactly supported. This property is important, as it guarantees that the smoothed image is obtained by a local average. Morphological filters may need a locality property for the same reason.

Definition 14.10. *We say that a translation invariant function operator T on $\mathcal{F}$ is local if there is some $M \geq 0$ such that*

$$(u = u' \text{ on } B(0, M)) \Rightarrow Tu(0) = Tu'(0).$$

The point 0 plays no special role in the definition. By translation invariance it is easily deduced from the definition that for $\mathbf{x} \in \mathbb{R}^N$, the values of $Tu(\mathbf{x})$ only depend upon the restriction of u to $B(\mathbf{x}, M)$.

Proposition 14.11. *Let $T : \mathcal{F} \to \mathcal{F}$ be a contrast and translation invariant standard monotone operator and $\mathcal{B}$ a set of structuring elements for T. If T is local, then $\mathcal{B}_M = \{B \in \mathcal{B} \mid B \subset \overline{B(0,M)}\}$ also is a set of structuring elements for T. Conversely, if all elements of $\mathcal{B}$ are contained in $\overline{B(0,M)}$, then T is local.*

Proof. We prove the statement with the sup-inf form for T given by Theorem 14.6,

$$Tu(\mathbf{x}) = \sup_{B\in\mathcal{B}} \inf_{\mathbf{y}\in B} u(\mathbf{x}+\mathbf{y}). \tag{14.5}$$

Consider the new function $u_\varepsilon(\mathbf{x}) = u(\mathbf{x}) - \frac{1}{\varepsilon}d(\mathbf{x}, B(0,M))$, where we take for d a distance function on $S_N \subset \mathbb{R}^{N+1}$, so that $u_\varepsilon \in \mathcal{F}$. Take any $B \in \mathcal{B}$ containing a point $\mathbf{z} \notin \overline{B(0,M)}$ and therefore not belonging to $\mathcal{B}_M$. Then $\inf_{\mathbf{y}\in B} u_\varepsilon(\mathbf{y}) \le u(\mathbf{z}) - \frac{1}{\varepsilon}d(\mathbf{z}, B(0,M)) < Tu(0)$ for ε small enough. So we can discard such B's in the computation of $Tu(0)$ by (14.5). Since by the locality assumption $Tu(0) = Tu_\varepsilon(0)$, we obtain

$$Tu(0) = Tu_\varepsilon(0) = \sup_{B\in\mathcal{B}_M} \inf_{\mathbf{y}\in B} u(\mathbf{y}).$$

By the translation invariance of all all considered operators, this proves the direct statement. The converse statement is straightforward. □

We end this paragraph with a definition and an easy characterization of isotropic operators in the sup-inf form. In the next proposition, we actually consider a more general setting, namely the invariance of T under some geometric group G of transformations of $\mathbb{R}^N$, for example the isometries. Since we use to extend the set and function operators to S_N, we must also extend such transforms by setting $g(\infty) = \infty$ for $g \in G$. Also, define the operator I_g on functions $u : S_N \to \mathbb{R}$ by $I_g u(\mathbf{x}) = u(g\mathbf{x})$.

Definition 14.12.

- *We say that $\mathcal{B}$ is invariant under a group G of transformations of S_N onto S_N if, for all $g \in G$, $B \in \mathcal{B}$ implies $gB \in \mathcal{B}$.*
- *If, for all $g \in G$, $TI_g = I_gT$ (resp. $\mathcal{T}g = g\mathcal{T}$), we say that T (resp. $\mathcal{T}$) is invariant under G.*
- *In particular, we say that T (resp. $\mathcal{T}$) is* isotropic *if it commutes with all linear isometries R of $\mathbb{R}^N$, and* affine invariant *if it commutes with all linear maps A with determinant 1.*

Proposition 14.13. *Let G be any group of linear maps : $g : \mathbb{R}^N \to \mathbb{R}^N$ extended to S_N by setting $g(\infty) = \infty$. If T (resp. $\mathcal{T}$) is translation invariant and invariant under G and $\mathcal{B}$ is a standard set of structuring elements for T (resp $\mathcal{T}$), then $G\mathcal{B} = \{gB \mid g \in G,\ B \in \mathcal{B}\}$ is another, G-invariant, standard set of structuring elements. Conversely, if $\mathcal{B}$ is a standard and G-invariant set of structuring elements for T (resp. $\mathcal{T}$), then this operator is G-invariant (and translation invariant.)*

Proof. All the verifications are straightforward. The only point to mention is that the considered groups are made of transforms sending bounded sets onto bounded sets and complementary sets of bounded sets onto complementary sets of bounded sets. □

Exercise 14.5. Prove carefully Proposition 14.13. ■

Some terminology.
It would be tedious to state theorems on operators on $\mathcal{F}$ with such a long list of requirements as *Standard Monotone, Translation and Contrast Invariant, Isotropic*. We shall call such operators *Morpho operators* because they retain the essential requirements of morphological smoothing filters . All the examples we consider in this book are actually Morpho operators. Not all are local, so we will specify it when needed. Operators can be still more invariant, in fact affine invariant, and we will specify it as well. Since all of these operators T have an inf-sup or a sup-inf form, we always take for $\mathcal{B}$ a standard structuring set reflecting the properties of T, that is, bounded in $B(0, M)$ when T is local and invariant by the same group as T. A last thing to specify is this: We have restricted our analysis to operators defined on $\mathcal{F}$. On the other hand, their inf-sup form permits to extend them to all measurable functions and we shall still denote the resulting operator by T. Tu can then assume the $-\infty$ and $+\infty$ values. All the same, it is an immediate check to see that this extension still is monotone and commutes with contrast changes:

Proposition 14.14. *Let T be a function operator in the* inf-sup *or* sup-inf *form associated with a standard set of structuring elements $\mathcal{B} \subset \mathcal{M}$. Then T is monotone and contrast invariant on the set of all bounded measurable functions of S_N.*

Exercise 14.6. Prove Proposition 14.14. ■

14.4 The who's who of monotone contrast invariant operators

The aim of this short section is to draw a synthetic picture of an equivalence chain built up in this chapter and in Chapter 13. We have constructed three kinds of objects,

- contrast and translation invariant standard monotone function operators $T : \mathcal{F} \to \mathcal{F}$;
- translation invariant standard monotone set operators $\mathcal{T}$ defined on $\mathcal{L}$;
- standard sets of structuring elements $\mathcal{B}$.

The results proven so far can be summarized in the following theorem.

Theorem 14.15. *Given any of the standard objects T, $\mathcal{T}$ and $\mathcal{B}$ mentioned above, one can pass to any other one by using one of the six formulae given* *below.*

$$\begin{aligned}
&\mathcal{B} \to T, && Tu(\mathbf{x}) = \sup_{B\in\mathcal{B}} \inf_{\mathbf{y}\in\mathbf{x}+B} u(\mathbf{y});\\
&\mathcal{B} \to \mathcal{T}, && \mathcal{T}X = \{\mathbf{x} \mid \exists B \in \mathcal{B},\ \mathbf{x} + B \subset X\};\\
&\mathcal{T} \to T, && Tu(\mathbf{x}) = \sup\{\lambda \mid \mathbf{x} \in \mathcal{T}\mathcal{X}_\lambda u\};\\
&T \to \mathcal{T}, && \mathcal{T}(\mathcal{X}_0 u) = \mathcal{X}_0(Tu);\\
&\mathcal{T} \to \mathcal{B}, && \mathcal{B} = \{B \in \mathcal{L} \mid 0 \in \mathcal{T}B\};\\
&T \to \mathcal{B}, && \text{by } T \to \mathcal{T} \text{ and } \mathcal{T} \to \mathcal{B}.
\end{aligned}$$

In addition, $\mathcal{B}$ can be bounded in some $B(0, M)$ if and only if T is local; T or $\mathcal{T}$ is G-invariant, for instance isotropic, if and only if it derives from some G-invariant (isotropic) $\mathcal{B}$. If an operator has the inf-sup or sup-inf form for some $\mathcal{B}$, it can be extended to all measurable functions on $\mathbb{R}^N$ into a monotone and contrast invariant operator.

Proof. Theorem 14.2 yields $\mathcal{T} \to \mathcal{B}$ and $\mathcal{B} \to \mathcal{T}$; Theorem 13.16 yields $\mathcal{T} \to T$; Theorem 14.6 yields $T \to \mathcal{T} \to \mathcal{B}$; Theorem 13.19 yields $T \to \mathcal{T}$. The final statements come from Propositions 14.11, 14.13 and 14.14. □

So we get a full equivalence between all objects, but we have left apart the commutation with thresholds property. When we define a set operator $\mathcal{T}$ from a function operator T by the level set extension, we know that $\mathcal{T} : \mathcal{L} \to \mathcal{L}$ is upper semicontinuous and that the commutation with thresholds $\mathcal{X}_\lambda(Tu) = \mathcal{T}(\mathcal{X}_\lambda u)$ holds. Conversely, if we define a function operator T as the stack filter of a standard monotone set $\mathcal{T}$, we do not necessarily have the commutation of thresholds; this is true only if $\mathcal{T}$ is upper semicontinuous on $\mathcal{L}$ (see Theorem 13.16) and this upper semicontinuity property is not always granted for interesting monotone operators, particularly when they are affine invariant. Fortunately enough, the commutation with thresholds is "almost" satisfied for any stack filter as we state in Proposition 14.18 in the next section.

14.4.1 Commutation with thresholds almost everywhere

In this section we always assume the considered sets to belong to $\mathcal{M}$ and the considered functions to be Lebesgue measurable. We say that a set X is contained in a set Y almost everywhere if

$$\text{measure}(X \setminus Y) = 0,$$

where measure denotes the usual Lebesgue measure in $\mathbb{R}^N$. We say that $X = Y$ almost everywhere if $X \subset Y$ and $Y \subset X$ almost everywhere. We say that two functions u and v are almost everywhere equal if $\text{measure}(\{\mathbf{x}, u(\mathbf{x}) \neq v(\mathbf{x})\}) = 0$.

Lemma 14.16. *Let $(X_\lambda)_{\lambda\in\mathbb{R}}$ be a nonincreasing family of sets of $\mathcal{M}$, that is $X_\lambda \subset X_\mu$ if $\lambda \geq \mu$. Then, for almost every λ in $\mathbb{R}$,*

$$X_\lambda = \bigcap_{\mu<\lambda} X_\mu, \qquad \textit{almost everywhere.} \tag{14.6}$$

Proof. Consider an integrable and strictly positive continuous function $h \in L^1(\mathbb{R}^N)$ (for instance, the gaussian.) Set $m(X) = \int_X h(\mathbf{x})d\mathbf{x}$. We notice that $m(X) = 0$ if and only if measure$(X) = 0$. The function $\lambda \to m(X_\lambda)$ is nonincreasing. Thus, it has a countable set of jumps. Since every countable set has zero Lebesgue measure, we deduce that for almost every λ,

$$\lim_{\mu \to \lambda} m(X_\mu) = m(X_\lambda).$$

As a consequence, for those λ's, $m(\bigcap_{\mu<\lambda} X_\mu \setminus X_\lambda) = 0$, which implies (14.6). □

Corollary 14.17. *Let $(X_\lambda)_{\lambda\in\mathbb{R}}$ be a family of measurable subsets of S_N such that $X_\lambda \subset X_\mu$ for $\lambda \geq \mu$, $X_\lambda = \emptyset$ for $\lambda \geq \lambda_0$, $X_\lambda = S_N$ for $\lambda \leq \mu_0$. Then the function u defined on S_N by the superposition principle*

$$u(\mathbf{x}) = \sup\{\lambda \mid \mathbf{x} \in X_\lambda\}$$

is bounded and satisfies for almost every λ, $X_\lambda = \mathcal{X}_\lambda u$ almost everywhere.

Proof. It is easily checked that $\mu_0 \leq u \leq \lambda_0$. We have

$$\mathcal{X}_\lambda u = \{\mathbf{x} \mid \sup\{\mu, \mathbf{x} \in X_\mu\} \geq \lambda\}$$

Now, if $\mathbf{x} \in X_\lambda$, we have $\sup\{\mu \mid \mathbf{x} \in X_\mu\} \geq \lambda$ which implies $\mathbf{x} \in \mathcal{X}_\lambda u$. Thus, $X_\lambda \subset \mathcal{X}_\lambda u$. Conversely, let λ be chosen so that $X_\lambda = \cap_{\mu<\lambda} X_\mu$ almost everywhere. This is by Lemma 14.16 true for almost every $\lambda \in \mathbb{R}$. Then if $\mathbf{x} \in \mathcal{X}_\lambda u$, we have by definition of u, $\mathbf{x} \in X_\mu$ for every $\mu < \lambda$. Thus $\mathbf{x} \in \bigcap_{\mu<\lambda} X_\mu$. We conclude that $X_\lambda u \subset \bigcap_{\mu<\lambda} X_\mu$ and therefore $\mathcal{X}_\lambda u \subset X_\lambda$ almost everywhere. □

Exercise 14.7. By using Corollary 14.17 show that if two measurable functions u and v are such that $\mathcal{X}_\lambda u = \mathcal{X}_\lambda v$ almost everywhere for almost every λ, then u and v are almost everywhere equal. ■

Proposition 14.18. *Let $\mathcal{T} : \mathcal{L} \to \mathcal{M}$ be a standard monotone set operator and T its stack filter. If $u \in \mathcal{F}$ then for almost every level $\lambda \in \mathbb{R}$,*

$$\mathcal{X}_\lambda(Tu) = \mathcal{T}(\mathcal{X}_\lambda(u)) \text{ almost everywhere.}$$

Proof. Since Tu is obtained from the sets $\mathcal{T}(\mathcal{X}_\lambda u)$ by superposition principle, this is an immediate consequence of Corollary 14.17. □

14.5 Exercises

Exercise 14.8. It is useful to have a test for $\mathcal{B}$ to determine whether or not the operator $\mathcal{T}$ can be expected to be upper semicontinuous on $\mathcal{L}$. Prove that the translation-invariant monotone operator in Theorem 14.2 defined by a given set $\mathcal{B}$ is upper semicontinuous on $\mathcal{L}$ if and only if the following condition holds: If $\bigcap_{n\in\mathbb{N}} \mathcal{T}X_n \neq \emptyset$, then there is a $B \in \mathcal{B}$ such that $\mathbf{x} + B \subset \bigcap_{n\in\mathbb{N}} X_n$, where $\mathbf{x} \in \bigcap_{n\in\mathbb{N}} \mathcal{T}X_n$ and $(X_n)_{n\in\mathbb{N}}$ is any nonincreasing sequence in $\mathcal{L}$. ■

Exercise 14.9. Suppose that $\mathcal{B} \subset \mathcal{L}$ contains exactly one set. Show that $\mathcal{T}$ is u.s.c. Generalize this to the case where $\mathcal{B}$ contains a finite number of sets. ∎

Exercise 14.10. Use Theorem 14.6 and Proposition 14.4 to show that the extrema killer T_a can be represented as a sup-inf function operator with the structuring elements

$$\mathcal{B}_a = \{B \mid B \text{ is compact, connected, meas}\,(B) = a, \text{ and } 0 \in B\} \cup \{\infty\}. \blacksquare$$

Check that $\mathcal{B}_a$ is standard.

Exercise 14.11. Let $\mathcal{B} = \{\{\mathbf{x}\} \mid \mathbf{x} \in D(0,1)\}$, $D(0,1) = \{\mathbf{x} \mid |\mathbf{x}| \leq 1\}$ and consider the associated set operator $\mathcal{T}$ and the associated function operator T, *defined on all measurable sets and functions of* $\mathbb{R}^N$ by formulas (14.1) and (14.3).

1) Check that $Tu(\mathbf{x}) = \sup_{\mathbf{y} \in \mathbf{x}+D} u(\mathbf{y})$.

2) Let $(q_n)_{n \in \mathbb{N}}$ be a countable dense set in $\mathbb{R}^N$ and consider u defined by $u(\mathbf{x}) = 1 - 1/n$ if $\mathbf{x} = q_n$ and $u(\mathbf{x}) = 0$ otherwise. Show that $\mathcal{T}\mathcal{X}_1 u \neq \mathcal{X}_1 T u$. The operator T in this exercise is one of the classic image operators called a *dilation.* Check that T commutes with thresholds when its domain of definition is restricted to $\mathcal{F}$ and the domain of $\mathcal{T}$ to $\mathcal{L}$. This example shows that this restriction is useful to get a simple theory. ∎

Exercise 14.12. Show the following property used in the proof of Lemma : if h is a positive continuous integrable function on $\mathbb{R}^N$ and if we set $m(X) = \int_X h(\mathbf{x})d\mathbf{x}$, then for every measurable set X, $m(X) = 0$ if and only if measure$(X) = 0$. ∎

14.6 Comments and references

The formalism presented in this chapter is due to Matheron [169] in the case of set operators and to Serra [228] and Maragos [157] in the case of function operators. Serra's formalism is actually more general than the one presented here; it will be developed in Chapter **??**, which is about "nonflat" morphology. Our presentation relating the sup-inf form of the operator directly to contrast invariance and establishing the full equivalence between sup-inf operators and contrast-invariant monotone operators is original. The fact, proven in Proposition 14.18 that commutation with thresholds occurs almost everywhere without further assumption was proven in [100].

The mysterious "set of structuring elements" has received a great deal of attention in the literature. Here are a few references: on finding the right set of structuring elements [218, 240]; on simplifying them [225]; on decomposing them into simpler ones as one does with linear filters [200, 270, 271]; on reducing the number [209].

Chapter 15

Erosions and Dilations

We are going to study in detail two of the simplest operators of mathematical morphology, the erosions and dilations. In fact, there will be essentially four operators: two set operators and the two related function operators. These operators will depend on a scale parameter t. We will also study the underlying PDEs $\partial u/\partial t = c|Du|$, where $c = 1$ for dilations and $c = -1$ for erosions.

15.1 Set and function erosions and dilations

We saw in Chapter 14 that every contrast-invariant monotone function operator has a sup-inf and an inf-sup representation in terms of some set of structuring elements. This is the point of view we take here, and furthermore, we assume that the set of structuring elements $\mathcal{B}$ has the simplest possible form, namely, $\mathcal{B} = \{B\}$. We actually introduce a parameter t scaling the size of B and therefore consider the two operators of the next definition.

Definition 15.1. *For $u \in \mathcal{F}$, define $D_{tB}u = D_t u$ by*

$$D_t u(\mathbf{x}) = \sup_{\mathbf{y} \in tB} u(\mathbf{x} - \mathbf{y}), \tag{15.1}$$

the "dilation of u by tB. In the same way, define $E_{tB}u = E_t u$, the "erosion of u by $-tB$", by

$$E_t u(\mathbf{x}) = \inf_{\mathbf{y} \in -tB} u(\mathbf{x} - \mathbf{y}). \tag{15.2}$$

These function operators have associated set operators.

Definition 15.2. *Let B be a non empty subset of $\mathbb{R}^N$ and let $t \geq 0$ be a scale parameter. The set operators $\mathcal{D}_{tB}$ and $\mathcal{E}_{tB}$ are defined on subsets $X \in \mathcal{M}(\mathbb{R}^N)$ by*

$$\mathcal{D}_{tB}X = \mathcal{D}_t X = X + tB = \{\mathbf{x} \mid \exists b \in B, \mathbf{x} - tb \in X\}, \tag{15.3}$$

$$\mathcal{E}_{tB} = \mathcal{E}_t X = \{\mathbf{x} \mid \mathbf{x} + tB \subset X\}, \tag{15.4}$$

and extended to $\mathcal{M}(S_N)$ by the standard extension (Definition 13.1.) $\mathcal{D}_t X$ is called the dilation of X by B at scale t. $\mathcal{E}_t X$ is called the erosion of X by B at scale t.

Exercise 15.1. (Duality formulas.) Show that $E_{tB}u = -D_{-tB}(-u)$ and $\mathcal{E}_{tB}X = (\mathcal{D}_{-tB}X^c)^c$. ■

Exercise 15.2. Show that if B is bounded, dilations and erosions are standard monotone operators. Compute their associated set of structuring elements (Proposition 14.2) and check that it is standard. ■

Theorem 15.3. *The function erosion by tB is the stack filter of the set erosion by tB ; the function dilation by tB is the stack filter of the set dilation by $t\overline{B}$ and the commutation with thresholds holds. In other terms for $u \in \mathcal{F}$ and all λ in $\mathbb{R}$, and calling $\overline{\mathcal{D}}_t$ the dilation by $t\overline{B}$,*

$$D_t u(\mathbf{x}) = \sup\{\lambda \mid \mathbf{x} \in \mathcal{D}_t \mathcal{X}_\lambda u\}, \quad \overline{\mathcal{D}}_t \mathcal{X}_\lambda u = \mathcal{X}_\lambda D_t u; \tag{15.5}$$

$$E_t u(\mathbf{x}) = \sup\{\lambda \mid \mathbf{x} \in \mathcal{E}_t \mathcal{X}_\lambda u\}, \quad \mathcal{E}_t \mathcal{X}_\lambda u = \mathcal{X}_\lambda E_t u. \tag{15.6}$$

Proof. We prove the statement for the dilations, the case of the erosions being just simpler. Consider some $X \in \mathcal{L}$ and $u(\mathbf{x}) \leq 0$ a function vanishing on X only. By the definition 13.18 of the level set extension $\tilde{\mathcal{D}}_t$ of D_t, $\tilde{\mathcal{D}}_t(X) = \mathcal{X}_0 D_t(u)$. Thus, using (15.3),

$$\mathbf{x} \in \tilde{\mathcal{D}}_t(X) \Leftrightarrow (D_t u)(\mathbf{x}) = 0 \Leftrightarrow \sup_{\mathbf{y} \in -tB} u(\mathbf{x} - \mathbf{y}) = 0 \Leftrightarrow$$

$$\exists \mathbf{y} \in t\overline{B},\ \mathbf{x} - \mathbf{y} \in X \Leftrightarrow \mathbf{x} \in X + t\overline{B} \Leftrightarrow \mathbf{x} \in \overline{\mathcal{D}}_t(X).$$

□

The operators $\mathcal{D}_t$ and $\mathcal{E}_t$ are in a certain sense the inverse of each other. This is clearly the case, for example, if $B = \{\mathbf{x}_0\}$. Then $\mathcal{D}_t$ is just the translation by $t\mathbf{x}_0$, and $\mathcal{E}_t = \mathcal{D}_t^{-1}$ is the translation by $-t\mathbf{x}_0$. If B is the open ball centered at zero with radius one, then $\mathcal{D}_t X$ is the set of all points whose distance from X is less than t, or the t-neighborhood of X. When B is symmetric with respect to zero, the operator $\mathcal{D}_t\mathcal{E}_t$ is called an *opening at scale* t and $\mathcal{E}_t\mathcal{D}_t$ is called a *closing at scale* t. These names have a topological origin. If B is the open ball centered at zero with radius one, then the opening at scale t of a set X is the union of all balls with radius t contained in X. The interior of X is the union of all open balls contained in X; it is also the largest open set contained in X. If we call the interior map $\mathcal{T}^\circ X = X^\circ$ the opening, then an opening at scale t appears as a quantified opening (see Exercise 15.6). The topological statement "the closure of the complement of X is the complement of the interior of X" has its counterpart for openings and closings at scale t, as shown in Exercise 15.6. The actions of erosions and dilations are illustrated in Figures 15.2, 15.2, and 15.2; actions of openings and closings are illustrated in Figures 15.2, 15.2, 15.2, 15.3, and 15.3.

15.2 Multiscale aspects

We say that the family of dilations $\{D_t \mid t > 0\}$ associated with a structuring element B is *recursive* if $D_t D_s = D_{t+s}$ for all $s, t > 0$, and similarly for the family $\{E_t \mid t > 0\}$. (A recursive family is also called a *semigroup.*) Being recursive is a very desirable property for any family of scaled operators used

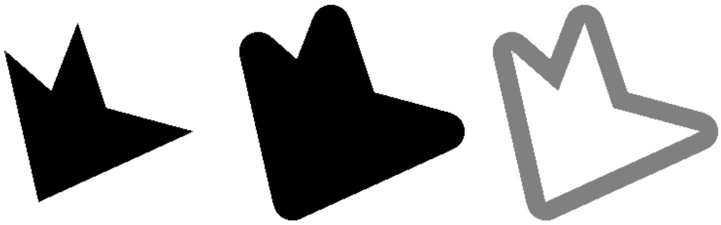

Figure 15.1: Dilation of a set. Left to right: A set; its dilation by a ball of radius 20; the difference set.

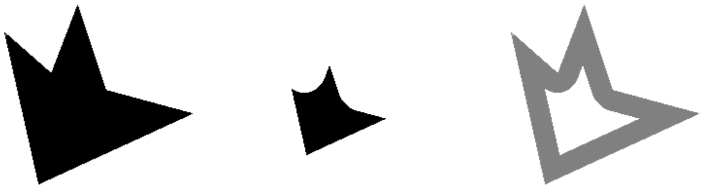

Figure 15.2: Erosion of a set. Left to right: A set; its erosion by a ball of radius 20; the difference set.

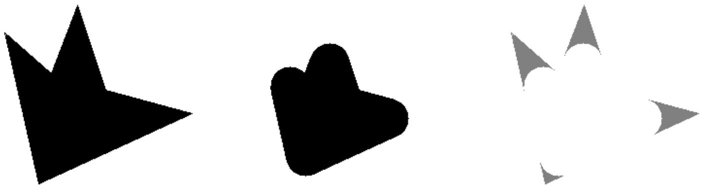

Figure 15.3: Opening of a set as curvature threshold from above. Left to right: A set X; its opening by a ball of radius 20; the difference set. This opening transforms X into the union of all balls of radius 20 contained in it. The resulting operation can be understood as a threshold from above of the curvature of the set boundary.

for image analysis. Having $D_t = (D_{t/n})^n$ is useful for practical computations. $\{D_t \mid t > 0\}$ and $\{E_t \mid t > 0\}$ will be recursive if and only if B is convex, but before proving this result we need the condition for B to be convex given in the next lemma. The proof of the next statement is an easy exercise.

Lemma 15.4. *B is convex if and only if $(s + t)B = sB + tB$ for all $s, t \geq 0$.*

Proposition 15.5. *The dilations $\mathcal{D}_t$ and the erosions $\mathcal{E}_t$ are recursive if and only the structuring element B is convex.*

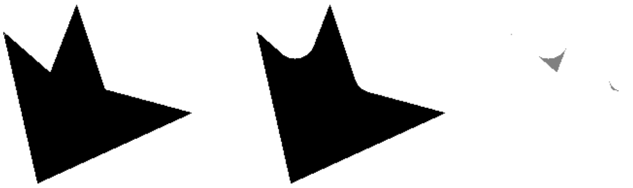

Figure 15.4: Closing of a set as a curvature threshold from below. Left to right: A set X; its closing by a ball of radius 20; the difference set. The closing of X is just the opening of X^c. It can be viewed as a threshold from below of the curvature of the set boundary.

Proof. Taking for simplicity B closed, we have

$$\mathcal{D}_t\mathcal{D}_sX = (X + sB) + tB = X + sB + tB$$

and

$$\mathcal{D}_{s+t}X = X + (s + t)B.$$

If $(t + s)B = tB + sB$, then clearly $\mathcal{D}_t\mathcal{D}_sX = \mathcal{D}_{s+t}X$. Conversely, if $\mathcal{D}_t\mathcal{D}_sX = \mathcal{D}_{s+t}X$, then by taking $X = \{0\}$ we see that $(t+s)B = tB+sB$. One can deduce the corresponding equivalence for erosions from the duality formula (Exercise 15.1). □

15.3 The PDEs associated with erosions and dilations

As indicated in the introduction to the chapter, scaled dilations and erosions are associated with the equations $\partial u/\partial t = \pm|Du|$. To explain this connection, we begin with a bounded convex set B that contains the origin, and we define the gauge $\|\cdot\|_B$ on $\mathbb{R}^N$ associated with B by $\|\mathbf{x}\|_B = \sup_{\mathbf{y}\in B}(\mathbf{x}\cdot\mathbf{y})$. If B is a ball centered at the origin with radius one, then $\|\cdot\|_B$ is the usual Euclidean norm, which we write simply as $|\cdot|$.

Proposition 15.6. *[Hopf–Lax formula [73, 140]]. Assume that B is a bounded convex set in $\mathbb{R}^N$ that contains the origin. Given $u_0 : \mathbb{R}^N \to \mathbb{R}$, define $u : \mathbb{R}^+ \times \mathbb{R}^N \to \mathbb{R}$ by $u(t,\mathbf{x}) = D_tu_0(\mathbf{x})$. Then u satisfies the equation*

$$\frac{\partial u}{\partial t} = \|Du\|_{-B}$$

at each point $(t,\mathbf{x})$ where u has continuous derivatives in t and $\mathbf{x}$. One has an analogue result with E_t and the equation $\partial u/\partial t = -\|Du\|_{-B}$.

Proof. We begin by proving the result for D_t at $t = 0$. Thus assume that u_0 is C^1 at $\mathbf{x}$. Then

$$u_0(\mathbf{x}-\mathbf{y}) - u_0(\mathbf{x}) = -Du_0(\mathbf{x})\cdot\mathbf{y} + o(|\mathbf{y}|),$$

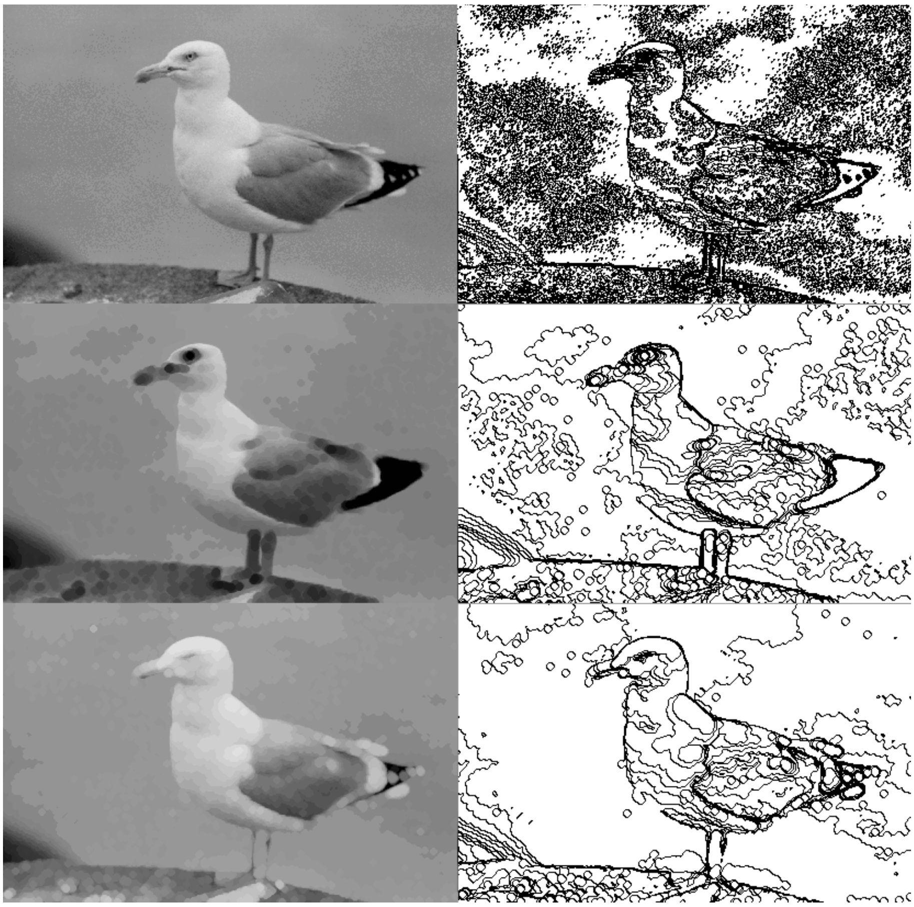

Figure 15.5: Erosion and dilation of a natural image. First row: a sea bird image and its level lines for all levels multiple of 12. Second row: an erosion with radius 4 has been applied. On the right, the resulting level lines where the circular shape of the structuring element (a disk with radius 4) appears around each local minimum of the original image. Erosion removes local maxima (in particular, all small white spots) but expands minima. Thus, all dark spots, like the eye of the bird, are expanded. Third row: the effect of a dilation with radius 4 and the resulting level lines. We see how local minima are removed (for example, the eye of the bird) and how white spots on the tail expand. Here, in turn, circular level lines appear around all local maxima of the original image.

and we have by applying D_h,

$$u(h,\mathbf{x}) - u(0,\mathbf{x}) = \sup_{\mathbf{y}\in hB} (-Du_0(\mathbf{x})\cdot\mathbf{y} + o(|\mathbf{y}|)).$$

Since B is bounded, the term $o(|\mathbf{y}|)$ is $o(|h|)$ uniformly for $\mathbf{y}\in hB$, and we get

$$u(h,\mathbf{x}) - u(0,\mathbf{x}) = h\sup_{\mathbf{z}\in B}((-Du_0(\mathbf{x})\cdot\mathbf{z}) + o(|h|).$$

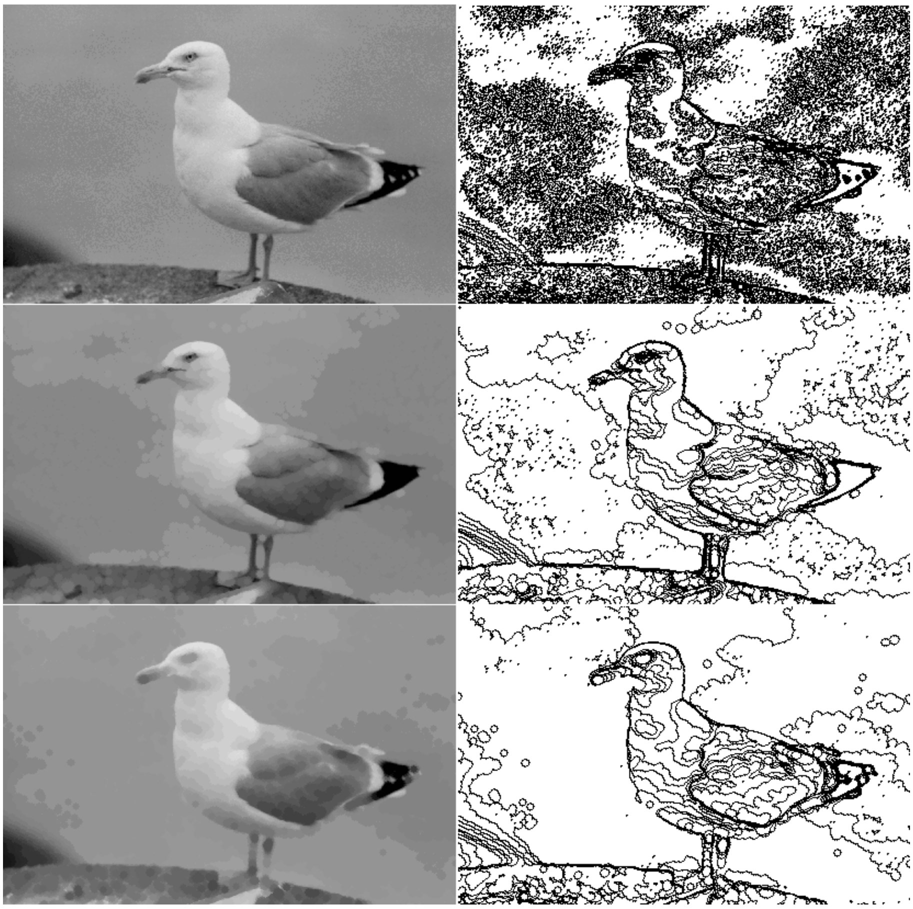

Figure 15.6: Openings and closings of a natural image. First row: the original image and its level lines for all levels multiple of 12. Second row: an opening with radius 4 has been applied. Third row: a closing with radius 4 has been applied. We can recognize the circular shape of the structuring element in the level lines displayed on the right.

We can divide both sides by h and pass to the limit as $|h| \to 0$ to obtain

$$\frac{\partial u}{\partial t}(0, \mathbf{x}) = \|Du_0(\mathbf{x})\|_{-B},$$

which is the result for $t = 0$. For an arbitrary $t > 0$, we have $D_{t+h} = D_t D_h = D_h D_t$, and we can write

$$u(t+h, \mathbf{x}) - u(t, \mathbf{x}) = D_h u(t, \cdot)(\mathbf{x}) - u(t, \mathbf{x}).$$

By repeating the argument made for $t = 0$ with u_0 replaced with $u(t, \cdot)$, we arrive at the general result. The proof for E_t is similar. □

Exercise 15.3. Prove the above result for E_t.

■

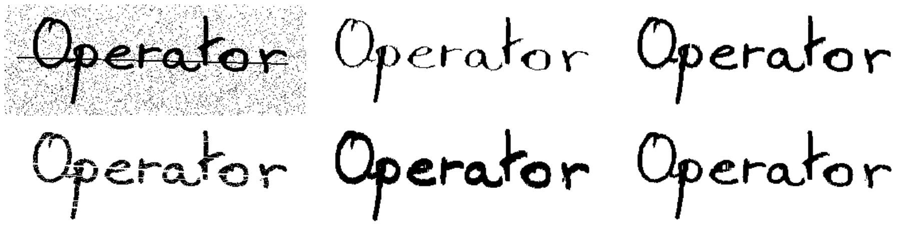

Figure 15.7: Denoising based on openings and closings. First row: scanned picture of the word "operator" with black dots and a black line added; a dilation with a 2×5 rectangle; an erosion with the same structuring element applied to the middle image. The resulting operator is a closing. Small black structures are removed by such a process. Second row: the word "operator" with a white line and white dots inside the letters; erosion with a rectangle 2×5; a dilation with the same structuring element applied to the middle image. The resulting operator is an opening. This time, small white structures are removed.

15.4 Exercises

Exercise 15.4. A straightforward adaptation on a grid $\mathbb{Z}\times\mathbb{Z}$ of the formulas $u(t,x) := \sup_{y\in B(x,t)} u_0(y)$ for dilation and $u(t,x) := \inf_{y\in B(x,t)} u_0(y)$ for erosion leads to the zero-order schemes

$$u^{n+1}(i,j) = \sup_{(k,l)\in B((i,j),t)\cap\mathbb{Z}^2} u^n(k,l)$$

and

$$u^{n+1}(i,j) = \inf_{(k,l)\in B((i,j),t)\cap\mathbb{Z}^2} u^n(k,l), \;\; u^0(i,j) = u_0(i,j).$$

Unfortunately, the zero-order schemes are strongly grid dependent. They do not make any difference between two balls which contain the same discrete pixels. In particular, such schemes only permit discrete motions of the shape boundaries. Thus, they are efficient only when t is large. Section 15.3 suggests that we can implement erosion and dilations on a finite image grid by more clever numerical schemes. One can try to discretize the associated PDE's $\partial u/\partial t = \pm|Du|$ by the Rouy-Tourin scheme:

$$u^{n+1}_{ij} = u^n_{ij} + \Delta t \left(\max(0, u^n_{i+1,j} - u^n_{ij}, u^n_{i-1,j} - u^n_{ij})^2 + \max(0, u^n_{i,j+1} - u^n_{ij}, u^n_{i,j-1} - u^n_{ij})^2\right)^{\frac{1}{2}}$$

for dilation and

$$u^{n+1}_{ij} = u^n_{ij} - \Delta t \left(\max(0, u^n_{ij} - u^n_{i+1,j}, u^n_{ij} - u^n_{i-1,j})^2 + \max(0, u^n_{ij} - u^n_{i,j+1}, u^n_{ij} - u^n_{i,j-1})^2\right)^{\frac{1}{2}}$$

for erosion. In both cases if $t = n\Delta t$ then $u^n(i,j)$ is a discrete version of $u(t,(i,j))$.

1) Explain why the schemes are consistent with their underlying partial differential equation. Check that with this clever scheme local maxima of u^n do not go up by dilation and local minima do not go down by erosion. Show that for example the following scheme would be a catastrophe at extrema (you'll have to try it anyway):

$$u^{n+1}_{ij} = u^n_{ij} + \Delta t \left(\max(|u^n_{i+1,j} - u^n_{ij}|, |u^n_{i-1,j} - u^n_{ij}|)^2 + \max(|u^n_{i,j+1} - u^n_{ij}|, |u^n_{i,j-1} - u^n_{ij}|)^2\right)^{\frac{1}{2}}.$$

2) Implement the schemes and compare their performance with the discrete zero order schemes for several shapes and images.

3) Compute on some well-chosen images the "top hat transforms" $u - O_t u$ and $F_t u - u$. The first transform aims at extracting all structures from an image which are

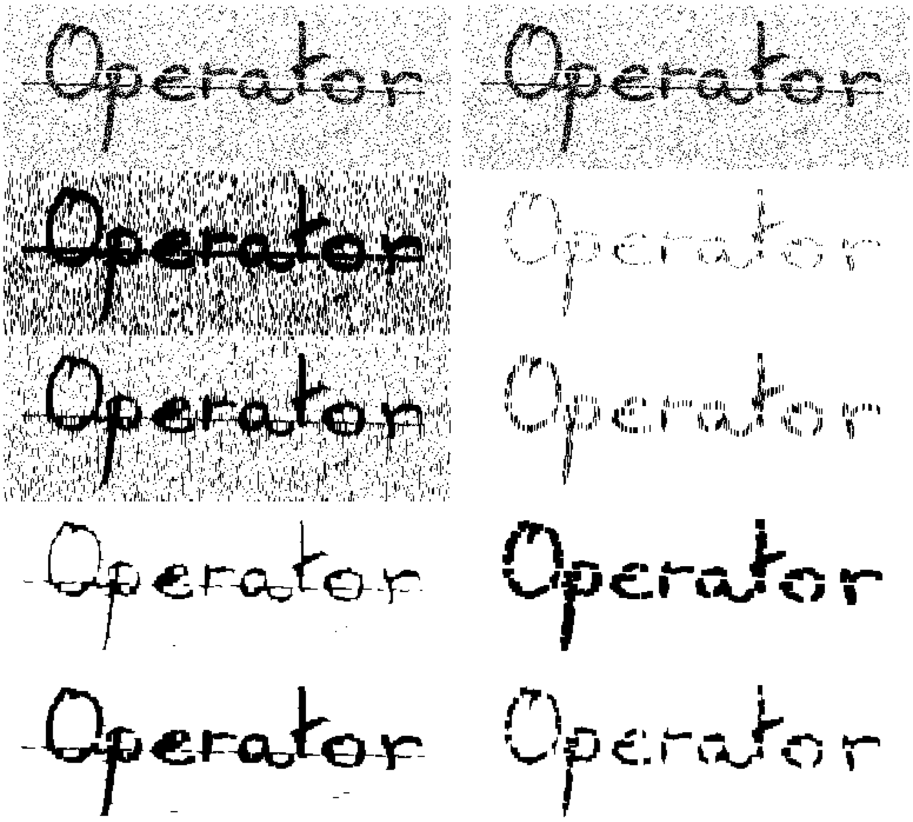

Figure 15.8: Examples of denoising based on opening or closing, as in Figure 15.7. Perturbations made with both black and white lines or dots have been added to the "operator" image. First column, top to bottom: original perturbed image; erosion with a 1×3 rectangle; then dilation with the same structuring element. (In other words, opening with this rectangle.) Then a dilation is applied with a rectangle 3×1, and finally an erosion with the same rectangle. Second column: The same process is applied, but with erosions and dilations exchanging their roles. It does not work so well because closing expands white perturbations and opening expands black perturbations. These operators do not commute. See Figure **??**, where an application of the median filter is more successful.

thinner than t and have brightness above the average. The second transform does the same job for dark structures. These transforms can be successfully applied on aerial images for extracting roads or rivers, and in many biological applications. ■

Exercise 15.5. Show that $E_t(u) = -D_t(-u)$ if B is symmetric with respect to zero. ■

Exercise 15.6.

(i) Let $B = \{\mathbf{x} \mid |\mathbf{x}| < 1\}$. Show that $\mathcal{D}_t\mathcal{E}_tX$ is the union of all open balls with radius t contained in X.

(ii) Let B be any structuring element that is symmetric with respect to zero. Write $X^c = \mathbb{R}^N \setminus X$. Show that $\mathcal{D}_tX^c = (\mathcal{E}_tX)^c$. Use this to show that $\mathcal{E}_t\mathcal{D}_tX^c =$

$(\mathcal{D}_t\mathcal{E}_t X)^c$. ■

Exercise 15.7. Prove that the dilation and erosion set operators associated with B are standard monotone if and only if B is bounded. If B is bounded and isotropic, prove that the associated erosion and dilation function operators are local *Morpho* operators. ■

15.5 Comments and references

Erosions and dilations. Matheron introduced dilations and erosions as useful tools for set and shape analysis in his fundamental book [169]. A full account of the properties of dilations, erosions, openings, and closings, both as set operators and function operators, can be found in Serra's books [228, 230]. We also suggest the introductory paper by Haralick, Sternberg, and Zhuang [102] and an earlier paper by Nakagawa and Rosenfeld [192]. An axiomatic algebraic approach to erosions, dilations, openings, and closings has been developed by Heijmans and Ronse [106, 212]. We did not develop this algebraic point of view here. The obvious relations between the dilations and erosions of a set and the distance function have been exploited numerically in [110], [132], and [234]. The skeleton of a shape can be defined as the set of points where the distance function to the shape is singular. A numerical procedure for computing the skeleton this way is proposed in [133].

The PDEs. The connection between the PDEs $\partial u/\partial t = \pm|Du|$ and multiscale dilations and erosions comes from the work of Lax, where it is used to give stable and efficient numerical schemes for solving the equations [140]. Rouy and Tourin [213] have shown that the distance function to a shape is a viscosity solution of $1 - |Du| = 0$ with the null boundary condition (Dirichlet condition) on the boundary of the shape. To define efficient numerical schemes for computing the distance function, they actually implement the evolution equation $\partial u/\partial t = 1 - |Du|$ starting from zero and with the null boundary condition on the boundary of the shape. The fact that the multiscale dilations and erosions can be computed using the PDEs $\partial u/\partial t = \pm|Du|$ has been rediscovered or revived, thirty years after Lax's work, by several authors: Alvarez et. al. [7], van den Boomgaard and Smeulders [249], Maragos [158, 159]. See also [248] for a numerical review. For an implementation using curve evolution, see [220]. Curiously, the link between erosions, dilations, and their PDEs seems to have remained unknown or unexploited until 1992. The erosion and dilation PDEs can be used for shape thinning, which is a popular way to compute the skeleton. Pasquignon developed an erosion PDE with adaptive stopping time that allows one to compute directly a skeleton that does not look like barbed wire [201].

Chapter 16

Median Filters and Mathematical Morphology

This entire chapter is devoted to median filters. They are among the most characteristic and numerically efficient contrast-invariant monotone operators. The denoising effects of median filters are illustrated in Figures 16.1 and 16.2; the smoothing effect of a median filter is illustrated in Figure 16.4. They also are extremely useful in 3D-image or movie denoising.

As usual, there will be two associated operators, a set operator and a function operator. All of the median operators (or filters) will be defined in terms of a nonnegative measurable weight function $k : \mathbb{R}^N \to [0, +\infty)$ that is normalized:

$$\int_{\mathbb{R}^N} k(\mathbf{y})\,\mathrm{d}\mathbf{y} = 1.$$

The k-measure of a measurable subset $B \subset \mathbb{R}^N$ is denoted by $|B|_k$ and defined by

$$|B|_k = \int_B k(\mathbf{y})\,\mathrm{d}\mathbf{y} = \int_{\mathbb{R}^N} k(\mathbf{y})\mathbf{1}_B(\mathbf{y})\,\mathrm{d}\mathbf{y}.$$

Clearly, $0 \le |B|_k \le 1$. The simplest example for k is given by the function $k = c_N^{-1}(r)\mathbf{1}_{B(0,r)}$, where $B(0,r)$ denotes the ball of radius r centered at the origin and $c_N(r)$ is the Lebesgue measure of $B(0,r)$. Another classical example to think of is the Gaussian.

16.1 Set and function medians

We first define the set operators, whose form is simpler. We define them on $\mathcal{M}(\mathbb{R}^N)$, the set of measurable subsets of $\mathbb{R}^N$ and then apply the standard extension to $\mathcal{M}(S_N)$ given in Definition 13.1.

Definition 16.1. *Let $X \in \mathcal{M}(\mathbb{R}^N)$ and let k be a weight function. The median set of X weighted by k is defined by*

$$\mathcal{M}\mathrm{ed}_k X = \{\mathbf{x} \mid |X - \mathbf{x}|_k \ge \tfrac{1}{2}\} \qquad (16.1)$$

and its standard extension to $\mathcal{M}(S_N)$ *by*

$$\mathcal{M}\mathrm{ed}_k X = \{\mathbf{x} \mid |X - \mathbf{x}|_k \geq \tfrac{1}{2}\} \cup (X \cap \{\infty\}). \tag{16.2}$$

The extension amounts to add ∞ to $\mathcal{M}\mathrm{ed}_k X$ if ∞ belongs to X. Note that we have already encountered the median operator in Section 7.1. Koenderink and van Doorn defined the dynamic shape of X at scale t to be the set of $\mathbf{x}$ such that $G_t * \mathbf{1}_X(\mathbf{x}) \geq 1/2$. The dynamic shape is, in our terms, a Gaussian-weighted median filter.

To gain some intuition about median filters, we suggest considering the weight k defined on $\mathbb{R}^2$ by $k = (1/\pi r^2)\mathbf{1}_{B(0,r)}$. Then $\mathbf{x} \in \mathbb{R}^2$ belongs to $\mathcal{M}\mathrm{ed}_k X$ if and only if the Lebesgue measure of $X \cap B(\mathbf{x}, r)$ is greater than or equal to half the measure of $B(0, r)$. Thus, $\mathbf{x} \in \mathcal{M}\mathrm{ed}_k X$ if points of X are in the majority around $\mathbf{x}$.

Lemma 16.2. $\mathcal{M}\mathrm{ed}_k$ *is a standard monotone operator on* $\mathcal{M}$.

Proof. Obviously $\mathcal{M}\mathrm{ed}_k(\emptyset) = \emptyset$ and $\mathcal{M}\mathrm{ed}_k(S_N) = S_N$. By definition, $\infty \in \mathcal{M}\mathrm{ed}_k X \Leftrightarrow \infty \in X$. If X is bounded, it is a direct application of Lebesgue theorem that

$$|X - \mathbf{x}|_k = \int k(\mathbf{y})\mathbf{1}_{X-\mathbf{x}}(\mathbf{y})d\mathbf{y} \to 0 \text{ as } \mathbf{x} \to \infty.$$

Thus $|X - \mathbf{x}|_k < \frac{1}{2}$ for $\mathbf{x}$ large enough and $\mathcal{M}\mathrm{ed}_k X$ is therefore bounded. In the same way, if X^c is bounded $|X - \mathbf{x}|_k \to 1$ as $\mathbf{x} \to \infty$ and therefore $(\mathcal{M}\mathrm{ed}_X)^c$ is bounded. □

Lemma 16.3. *We can represent* $\mathcal{M}\mathrm{ed}_k$ *by*

$$\mathcal{M}\mathrm{ed}_k X = \{\mathbf{x} \mid \mathbf{x} + B \subset X, \textit{ for some } B \in \mathcal{B}\}, \tag{16.3}$$

where $\mathcal{B} = \{B \mid |B|_k \geq \frac{1}{2}\}$ *or* $\mathcal{B} = \{B \mid |B|_k = \frac{1}{2}\}$.

Proof. By Lemma 16.2, $\mathcal{M}\mathrm{ed}_k$ is standard monotone and it is obviously translation invariant. So we can apply Theorem 14.2. The canonical set of structuring elements of $\mathcal{M}\mathrm{ed}_k$ is

$$\mathcal{B} = \{B \mid 0 \in \mathcal{M}\mathrm{ed}_k B\} = \{B \mid |B|_k \geq \frac{1}{2}\}.$$

The second set $\mathcal{B}$ mentioned in the lemma, which we call now for convenience $\mathcal{B}'$, is a subset of $\mathcal{B}$ such that for every $B \in \mathcal{B}$, there is some $B' \in \mathcal{B}'$ such that $B' \subset B$. Thus by Proposition 14.4, $\mathcal{M}\mathrm{ed}_k$ can be defined from $\mathcal{B}'$. □

The next lemma will help defining the function operator Med_k associated with the set operator $\mathcal{M}\mathrm{ed}_k$.

Lemma 16.4. *The set operator* $\mathcal{M}\mathrm{ed}_k$ *is translation invariant and upper semicontinuous on* $\mathcal{M}$.

Proof. The first property is straightforward. Consider a nonincreasing sequence $(X_n)_{n\in\mathbb{N}}$ in $\mathcal{M}$ and let us show that

$$\mathcal{M}\mathrm{ed}_k \bigcap_{n\in\mathbb{N}} X_n = \bigcap_{n\in\mathbb{N}} \mathcal{M}\mathrm{ed}_k X_n.$$

Since $\mathcal{M}\mathrm{ed}_k$ is monotone, it is always true that $\mathcal{M}\mathrm{ed}_k \bigcap_{n\in\mathbb{N}} X_n \subset \bigcap_{n\in\mathbb{N}} \mathcal{M}\mathrm{ed}_k X_n$. To prove the other inclusion, assume that $\mathbf{x} \in \bigcap_{n\in\mathbb{N}} \mathcal{M}\mathrm{ed}_k X_n$. If $\mathbf{x} \in \mathbb{R}^N$, by the definition of $\mathcal{M}\mathrm{ed}_k$, $|X_n - \mathbf{x}| \geq 1/2$ for all $n \in \mathbb{N}$. Since $X_n - \mathbf{x} \downarrow \bigcap_{n\in\mathbb{N}}(X_n - \mathbf{x})$, we deduce from Lebesgue Theorem that $|X_n - \mathbf{x}|_k \downarrow |\bigcap_{n\in\mathbb{N}}(X_n - \mathbf{x})|_k$. This means that $|\bigcap_{n\in\mathbb{N}}(X_n - \mathbf{x})|_k \geq 1/2$, and hence that $\mathbf{x} \in \mathcal{M}\mathrm{ed}_k \bigcap_{n\in\mathbb{N}}(X_n - \mathbf{x})$. If $\mathbf{x} = \infty$, it belongs to $\mathcal{M}\mathrm{ed}_k X_n$ for all n and therefore to X_n for all n. Thus, it belongs to $\bigcap_{n\in\mathbb{N}} X_n$ and therefore to $\mathcal{M}\mathrm{ed}_k(\bigcap_{n\in\mathbb{N}} X_n)$. □

Definition 16.5 (and proposition). *Define the function operator* Med_k *from* $\mathcal{M}\mathrm{ed}_k$ *as a stack filter,*

$$\mathrm{Med}_k u(\mathbf{x}) = \sup\{\lambda \mid \mathbf{x} \in \mathcal{M}\mathrm{ed}_k \mathcal{X}_\lambda u\}.$$

Then Med_k *is standard monotone, contrast invariant and translation invariant from* $\mathcal{F}$ *to* $\mathcal{F}$. Med_k *and* $\mathcal{M}\mathrm{ed}_k$ *commute with thresholds,*

$$\mathcal{X}_\lambda \mathrm{Med}_k u = \mathcal{M}\mathrm{ed}_k \mathcal{X}_\lambda u. \tag{16.4}$$

If k is radial, Med_k *therefore is Morpho.*

Proof. By Lemma 16.4, $\mathcal{M}\mathrm{ed}_k$ is upper semicontinuous and by Lemma 16.2 it is standard monotone and translation invariant. So we can apply Theorem 13.16, which yields all announced properties for Med_k. □

We get a sup-inf formula for the median as a direct application of Theorem 14.6.

Proposition 16.6. *The median operator* Med_k *has the sup-inf representation*

$$\mathrm{Med}_k u(\mathbf{x}) = \sup_{B\in\mathcal{B}} \inf_{\mathbf{y}\in\mathbf{x}+B} u(\mathbf{y}), \tag{16.5}$$

where $\mathcal{B} = \{B \mid B \in \mathcal{M}, |B|_k = 1/2\}$.

A median value is a kind of average, but with quite different results, as is illustrated in Exercise 16.4.

16.2 Self-dual median filters

The median operator Med_k, as defined, is not invariant under "reverse contrast," that is, it does not satisfy $-\mathrm{Med}_k u = \mathrm{Med}_k(-u)$ for all $u \in \mathcal{F}$. This is clear from the example in the next exercise. Self-duality is a conservative requirement which is true for all linear filters. It means that the white and black balance is respected by the operator. We have seen that dilations favor whites and erosions favor black colors: These operators are not self-dual.

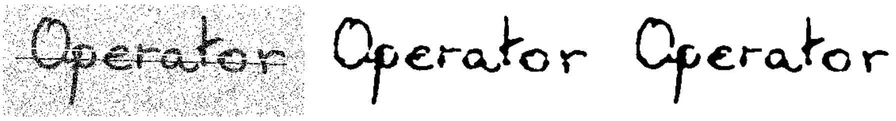

Figure 16.1: Example of denoising with a median filter. Left to right: scanned picture of the word "operator" with perturbations and noise made with black or white lines and dots; the image after one application of a median filter with a circular neighborhood of radius 2; the image after a second application of the same filter. Compare with the denoising using openings and closings (Figure 15.8).

Exercise 16.1. Consider the one-dimensional median filter with $k = \frac{1}{2}\mathbf{1}_{[-2,-1]\cup[1,2]}$. Let $u(x) = -1$ if $x \le -1$, $u(x) = 1$ if $x \ge 1$, $u(x) = x$ elsewhere. Check that $\mathrm{Med}_k u(0) \ne -\mathrm{Med}(-u)(0)$. ■

As we did with erosions and dilations, one can define a dual version of the median Med_k^- by

$$\mathrm{Med}_k^- u = -\mathrm{Med}_k(-u), \text{ so that} \tag{16.6}$$

$$\mathrm{Med}_k^- u(\mathbf{x}) = \inf_{|B|_k \ge \frac{1}{2}} \sup_{\mathbf{y}\in\mathbf{x}+B} u(\mathbf{y}). \tag{16.7}$$

A quite general condition on k is sufficient to guarantee that Med_k and Med_k^- agree on continuous functions.

Definition 16.7. *We say that k is not separable if $|B|_k \ge 1/2$ and $|B'|_k \ge 1/2$ imply that $\overline{B} \cap \overline{B'} \ne \emptyset$.*

Proposition 16.8.

(*i*) *For every measurable function u, $\mathrm{Med}_k u \ge \mathrm{Med}_k^- u$.*

(*ii*) *Assume that k is not separable. Then for every $u \in \mathcal{F}$, $\mathrm{Med}_k u = \mathrm{Med}_k^- u$ and Med_k is self-dual.*

Proof. Both operators are translation invariant, so without loss of generality we may assume that $\mathbf{x} = 0$. To prove (*i*), let $\lambda = \mathrm{Med}_k u(0) = \sup_{|B|_k \ge 1/2} \inf_{y\in B} u(\mathbf{y})$. Take $\varepsilon > 0$ and consider the level set $\mathcal{X}_{\lambda+\varepsilon} u$. Then $\inf_{\mathbf{y}\in\mathcal{X}_{\lambda+\varepsilon}} u(\mathbf{y}) \ge \lambda+\varepsilon$. Thus $|\mathcal{X}_{\lambda+\varepsilon} u|_k < 1/2$, since $\inf_{\mathbf{y}\in B} \le \lambda$ for any set B such that $|B| \ge 1/2$. Thence $|(\mathcal{X}_{\lambda+\varepsilon} u)^c|_k \ge 1/2$. By the definition of level sets, $\sup_{\mathbf{y}\in(\mathcal{X}_{\lambda+\varepsilon} u)^c} u(\mathbf{y}) \le \lambda + \varepsilon$. These two last relations imply that

$$\inf_{|B|_k \ge \frac{1}{2}} \sup_{\mathbf{y}\in B} u(\mathbf{y}) \le \lambda + \varepsilon.$$

Since $\varepsilon > 0$ was arbitrary, this proves (*i*).

The assumption that k is not separable implies that for all B and B' having k-measure greater than or equal to $1/2$, we have $\inf_{\mathbf{y}\in\overline{B}} u(\mathbf{y}) \le \sup_{\mathbf{y}\in\overline{B'}} u(\mathbf{y})$. Since $u \in \mathcal{F}$ is continuous, $\inf_{\mathbf{y}\in B} u(\mathbf{y}) \le \sup_{\mathbf{y}\in B'} u(\mathbf{y})$. Since B and B' were arbitrary except for the conditions $|B|_k \ge 1/2$ and $|B'|_k \ge 1/2$, the last inequality implies that

$$\sup_{|B|_k \ge \frac{1}{2}} \inf_{\mathbf{y}\in B} u(\mathbf{y}) \le \inf_{|B'|_k \ge \frac{1}{2}} \sup_{\mathbf{y}\in B'} u(\mathbf{y}).$$

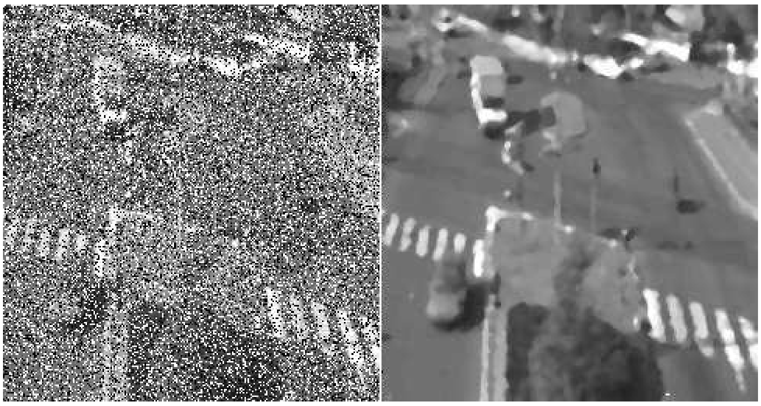

Figure 16.2: Denoising based on a median filter. Left: an image altered on 40% of its pixels with salt and pepper noise. Right: the same image after three iterations of a median filter with a 3×3 square mask.

From this last inequality and (i), we conclude that $\mathrm{Med}_k u = \mathrm{Med}_k^- u$. □

16.2.1 Chessboard dilemma and fattening effect

In Figure 16.2.1, the median filter has been applied iteratively to a function u whose grid values are equal to 255 at the white pixels and to 0 at the black pixels. The function is continuous, being (e.g.) interpolated by standard bilinear interpolation. The iso-level set $I_{127.5}u := \{\mathbf{x} \mid u(\mathbf{x}) = 127.5\}$ consists of the line segments separating the squares and has therefore zero measure. As we know, the median filter tends to smooth, to round off the level lines of the image. Yet we have with a chessboard a fundamental ambiguity : are these iso-level lines surrounding the black squares, or are they surrounding the white squares? In other terms, do we see in a chessboard a set of white squares on black background, or conversely?

Since our operator is self-dual it doesn't favor any of the considered interpretations: it rounds off simultaneously the lines surrounding the black squares and the level lines surrounding the white squares (second image of Figure 16.2.1). This results in the "fattening" of the level lines separating white and black, which have the mid-level 127.5. Hence the appearance in the second image of a grey zone separating the smoothed out black and white squares. If we take a level set $\mathcal{X}_\varepsilon Tu$ of this image with $\varepsilon < 0$ (third image), the fattened set joins the level set and we observe black squares on white background. Symmetrically if $\varepsilon > 0$ the level set shows white squares on black background.

16.3 Discrete median filters and the "usual" median value

We define a discrete median filter by considering, instead of a function, a uniform discrete measure $k = \sum_{i=1,\dots,N} \delta_{\mathbf{x}_i}$, where $\delta_{\mathbf{x}_i}$ denotes the Dirac mass at $\mathbf{x}_i$.

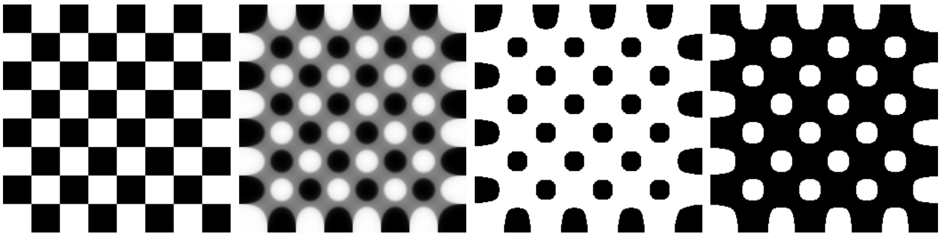

Figure 16.3: The chessboard dilemma. Left: a chessboard image. The next image is obtained by a self-dual median filter. Notice the expansion of the median grey level, 127.5, who was invisible in the original image and grows in the second one. This effect is called "fattening effect". The third and fourth image show the evolution of the level sets at levels 127 and 128 respectively. This experiment illustrates a dilemma as to whether we consider the chessboard as black squares on white background, or conversely.

We could normalize k, but this is not necessary, as will become clear. Translates of the points $\mathbf{x}_i$ create the discrete neighborhood that is used to compute the median value of a function u at a point $\mathbf{x}$. We denote the set of subsets of $\{1, \dots, N\}$ by $\mathcal{P}(N)$ and the number of elements in $P \in \mathcal{P}(N)$ by $\operatorname{card}(P)$. Since $\operatorname{card}(P) = |P|_k$, we will suppress the k-notation is favor of the more transparent "card(P)," but one should remember that the k-measure is still there. An immediate generalization of the definition of the median filters to the case where k is such a discrete measure yields

$$\begin{aligned}
\mathrm{Med} u(\mathbf{x}) &= \sup_{\substack{P \in \mathcal{P}(N) \\ \operatorname{card}(P) \geq N/2}} \inf_{i \in P} u(\mathbf{x} - \mathbf{x}_i), \\
\mathrm{Med}^- u(\mathbf{x}) &= \inf_{\substack{P \in \mathcal{P}(N) \\ \operatorname{card}(P) \geq N/2}} \sup_{i \in P} u(\mathbf{x} - \mathbf{x}_i).
\end{aligned}$$

When k was continuous, we could replace "$|B|_k \geq 1/2$" with "$|B|_k = 1/2$," but this is not directly possible in the discrete case, since $N/2$ is not an integer if N is odd. To fix this, we define the function M by $M(N) = N/2$ if N is even and $M(N) = (N/2) + (1/2)$ if N is odd. Now we have

$$\begin{aligned}
\mathrm{Med} u(\mathbf{x}) &= \sup_{\substack{P \in \mathcal{P}(N) \\ \operatorname{card}(P) = M(N)}} \inf_{i \in P} u(\mathbf{x} - \mathbf{x}_i), \\
\mathrm{Med}^- u(\mathbf{x}) &= \inf_{\substack{P \in \mathcal{P}(N) \\ \operatorname{card}(P) = M(N)}} \sup_{i \in P} u(\mathbf{x} - \mathbf{x}_i).
\end{aligned}$$

The fact that we can replace "$\operatorname{card}(P) \geq N/2$" with "$\operatorname{card}(P) = M(N)$" has been argued elsewhere for the continuous case; for the discrete case, it is a matter of simple combinatorics. Given any $\mathbf{x}$, let $y_i = u(\mathbf{x} - \mathbf{x}_i)$. After a suitable permutation of the i's, we can order the y_i as follows: $y_1 \leq, \cdots, \leq y_M \leq, \cdots \leq$

y_N. Then for N even,

$$\{\inf_{i\in P} y_i \mid \mathrm{card}(P) \geq N/2\} = \{\inf_{i\in P} y_i \mid \mathrm{card}(P) = M\} = \{y_1, \dots, y_{M+1}\},$$
$$\{\sup_{i\in P} y_i \mid \mathrm{card}(P) \geq N/2\} = \{\sup_{i\in P} y_i \mid \mathrm{card}(P) = M\} = \{y_M, \dots, y_N\},$$

and $\mathrm{Med}u(\mathbf{x}) = y_{M+1} \geq y_M = \mathrm{Med}^- u(\mathbf{x})$. If N is odd, we have

$$\{\inf_{i\in P} y_i \mid \mathrm{card}(P) \geq N/2\} = \{\inf_{i\in P} y_i \mid \mathrm{card}(P) = M\} = \{y_1, \dots, y_M\},$$
$$\{\sup_{i\in P} y_i \mid \mathrm{card}(P) \geq N/2\} = \{\sup_{i\in P} y_i \mid \mathrm{card}(P) = M\} = \{y_M, \dots, y_N\},$$

and $\mathrm{Med}u(\mathbf{x}) = \mathrm{Med}^- u(\mathbf{x}) = y_M$. This shows that $\mathrm{Med} = \mathrm{Med}^-$ if and only if N is odd. What we see here is the discrete version of Proposition 16.8. When N is odd, the measure is not separable, since two sets P and P' with $\mathrm{card}(P) \geq N/2$ and $\mathrm{card}(P') \geq N/2$ always have a nonempty intersection. In general, a median filter with an odd number of pixels is preferred, since $\mathrm{Med} = \mathrm{Med}^-$ in this case.

This discussion shows that the definition of the discrete median filter Med corresponds to the usual statistical definition of the median of a set of data: If the given data consists of the numbers $y_1 \leq y_2 \leq \cdots \leq y_N$ and $N = 2n+1$, them by definition, the median is y_{n+1}. In case $N = 2n$, the median is $(y_n + y_{n+1})/2$. In both cases, half of the terms are greater than or equal to the median and half of the terms are less than or equal to the median. The usual median minimizes the functional $\sum_{i=1}^{N} |y_i - y|$. Exercise 16.9 shows how Med and Med^- relate to this functional.

Finally, we wish to show that the discrete median filter Med can be a cyclic operator on discrete images. As a simple example, consider the chessboard image, where $u(i,j) = 255$ if $i+j$ is even and $u(i,j) = 0$ otherwise. When we apply the median filter that takes the median of the four values surrounding a pixel and the pixel value, it is clear that the filter "reverses" the chessboard pattern. Indeed, any white pixel (value 255) is surrounded by four black pixels (value zero), so the median filter transforms the white pixel into a black pixel. In the same way, a black pixel is transformed into a white pixel and this can go for ever.

16.4 Exercises

Exercise 16.2. Check that $\mathcal{M}\mathrm{ed}_k$ as defined in Definition 16.1 is monotone and translation invariant. ■

Exercise 16.3. Koenderink and van Doorn defined the dynamic shape of X at scale t to be the set of $\mathbf{x}$ such that $G_t * \mathbf{1}_X(\mathbf{x}) \geq 1/2$. Check that this is a Gaussian-weighted median filter. ■

Exercise 16.4. Consider the weighted median filter defined on S_1 with $k = (1/2)\mathbf{1}_{[-1,1]}$. Compute $\mathrm{Med}_k u$ for $u(x) = \frac{1}{1+x^2}$. Compare the result with the local average $M_1 u(x) = \frac{1}{2}\int_{-1}^{1} u(x+y)dy$. What happens on intervals where u is monotone? ■

Exercise 16.5. Saying that k is not separable is a fairly weak assumption. It corresponds roughly to saying that the support of k cannot be split into two disjoint connected components each having k-measure $1/2$. Show that if k is continuous and if its support is connected, then it is not separable. ■

Exercise 16.6. Prove the following inequalities for any measurable function :

$$\sup_{|B|_k \geq \frac{1}{2}} \inf_{\mathbf{y} \in \mathbf{x}+B} u(\mathbf{y}) \geq \sup_{|B|_k > \frac{1}{2}} \inf_{\mathbf{y} \in \mathbf{x}+B} u(\mathbf{y}) \geq \inf_{|B|_k \geq \frac{1}{2}} \sup_{\mathbf{y} \in \mathbf{x}+B} u(\mathbf{y}),$$

$$\sup_{|B|_k \geq \frac{1}{2}} \inf_{\mathbf{y} \in \mathbf{x}+B} u(\mathbf{y}) \geq \inf_{|B|_k > \frac{1}{2}} \sup_{\mathbf{y} \in \mathbf{x}+B} u(\mathbf{y}) \geq \inf_{|B|_k \geq \frac{1}{2}} \sup_{\mathbf{y} \in \mathbf{x}+B} u(\mathbf{y}).$$

■

Exercise 16.7. Median filter on measurable sets and functions. The aim of the exercise is to study the properties of the median filter extended to the set $\mathcal{M}$ of all measurable sets of S_N and all bounded measurable functions ($u \in L^\infty(S_N)$). The definition of $\mathcal{M}\mathrm{ed}_k$ on $\mathcal{M}$ is identical to the current definition.

1) Using the result of Exercise 13.20, show that one can define Med_k from $\mathcal{M}\mathrm{ed}_k$ as a stack filter and that it is monotone, translation and contrast invariant. In addition, $\mathcal{M}\mathrm{ed}_k$ and Med_k still satisfy the commutation with thresholds, $\mathcal{X}_\lambda \mathrm{Med}_k u = \mathcal{M}\mathrm{ed}_k \mathcal{X}_\lambda u$.

2) Prove that $\mathcal{M}\mathrm{ed}_k$ maps measurable sets into closed sets. Deduce that if u is a measurable function, then $\mathrm{Med}_k u$ is upper semicontinuous and $\mathrm{Med}_k^- u$ is lower semicontinuous.

3) Assume that k is not separable. Check that the proof of Proposition 16.8 still applies to the more general Med_k and Med_k^-, applied to all measurable functions. Deduce that if k is not separable, then $\mathcal{M}\mathrm{ed}_k u$ is continuous whenever u is a measurable function.

■

Exercise 16.8. The discrete median filters can more generally be defined in terms of a nonuniform measure k that places different weights k_i on the points $\mathbf{x}_i$, so $|\{x_i\}|_k = k_i$. Check that $\mathrm{Med}_k^- u \leq \mathrm{Med}_k u$. Prove that $\mathrm{Med}_k^- u = \mathrm{Med}_k u$ if and only if there is no subset of the numbers $k_1, \dots, k_N$ whose sum is $K/2$. In particular, if the k_i are integers and K is odd, then $\mathrm{Med}_k^- u = \mathrm{Med}_k u$. ■

Exercise 16.9. Variational interpretations of the median and the average values.

Let $\mathrm{arginf}_m g(m)$ denote the value of m, if it exists, at which g attains its infimum. Consider N real numbers $\{\mathbf{x}_i \mid i = 1, 2, \dots, N\}$ and denote by $\mathrm{Med}((x_i)_i)$ and $\mathrm{Med}^-((x_i)_i)$ their usual lower and upper median values (we already know that both are equal if N is odd but can be different if N is even).

(i) Show that

$$\frac{1}{N}\sum_{i=1}^{N} x_i = \mathrm{arginf}_m \sum_{i=1}^{N} (x_i - m)^2.$$

(ii) Show that

$$\mathrm{Med}^-((x_i)_i) \leq \mathrm{arginf}_m \sum_{i=1}^{N} |x_i - m| \leq \mathrm{Med}((x_i)_i).$$

(iii) Let $k = \mathbf{1}_B$, where B is set with Lebesgue measure equal to one. Let $\mathrm{Med}_B u$ denote the *median value of u in B*, defined by $\mathrm{Med}_B u = \mathrm{Med}_k u(0)$. Consider a bounded measurable function u defined on B. Show that

$$\int_B u(\mathbf{x})\,\mathrm{d}\mathbf{x} = \mathrm{arginf}_m \int_B (u(\mathbf{x}) - m)^2\,\mathrm{d}\mathbf{x}$$

and that

$$\mathrm{Med}_B^- u \leq \mathrm{arginf}_m \int_B |u(\mathbf{x}) - m|\,\mathrm{d}\mathbf{x} = \frac{\mathrm{Med}_B^- u + \mathrm{Med}_B u}{2} \leq \mathrm{Med}_B u.$$

(iv) Deduce from the above that the mean value is the best constant approximation in the L^2 norm and that the median is the best constant approximation in the L^1 norm. ■

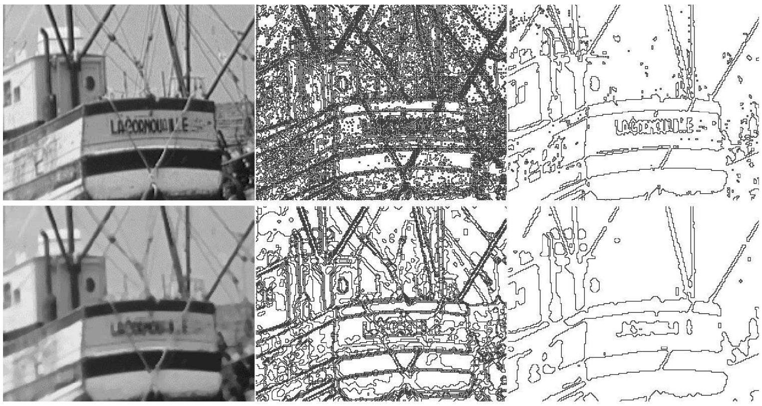

Figure 16.4: Smoothing effect of a median filter on level lines. Above, left to right: original image; all of its level lines (boundaries of level sets) with levels multiple of 12; level lines at level 100. Below,left to right: result of two iterations of a median filter with a disk with radius 2; corresponding level lines (levels multiple of 12); level lines at level 100.

16.5 Comments and references

The remarkable denoising properties and numerical efficiency of median filters for the removal of all kinds of impulse noise in digital images, movies, and video signals are well known and acclaimed [66, 117, 193, 202, 207]. The last reference cited as well as the next three all propose simple and efficient implementations of the median filter [19, 64, 111]. An introduction to weighted median filters can be found in [35, 266], and information about some generalizations (conditional median filters, for example) can be found in [16, 142, 239]. The min, max, and median filters are particular instances of rank order filters; see [60] for a general presentation of these filters. There are few studies on iterated median filters. The use of iterated median filters as a scale space is, however, proposed in [23]. The extension of median filtering to multichannel (color) images is problematic, although there have been some interesting attempts [50, 208].

Part III

Local Asymptotic Analysis of Operators

Chapter 17

Curves and Curvatures

This chapter contains the fundamentals of differential geometry that are used in the book. Our main aim is to define the orientation and curvatures of a curve or a surface as the main contrast invariant differential operators we shall deal with in image and curve smoothing.

17.1 Tangent, normal, and curvature

We summarize in this section the concepts and results about smooth curves that are needed in this chapter and elsewhere in the book. The curves we considered will always be plane curves.

Definition 17.1. *We call simple arc or Jordan arc the image* Γ *of a continuous one-to-one function* $\mathbf{x} : [0,1] \to \mathbb{R}^2$, $\mathbf{x}(t) = (x(t), y(t))$. *We say that* Γ *is a simple closed curve or Jordan curve if the mapping restricted to* $(0,1)$ *is one-to-one and if* $\mathbf{x}(0) = \mathbf{x}(1)$. *If* $\mathbf{x}$ *is continuously differentiable on* $[0,1]$, *we define the arc length of the segment of the curve between* $\mathbf{x}(t_0)$ *and* $\mathbf{x}(t)$ *by*

$$L(\mathbf{x}, t_0, t) = \int_{t_0}^{t} |\mathbf{x}'(\tau)| \, \mathrm{d}\tau = \int_{t_0}^{t} \sqrt{\mathbf{x}'(\tau) \cdot \mathbf{x}'(\tau)} \, \mathrm{d}\tau. \tag{17.1}$$

In particular, set

$$L(t) = L(\mathbf{x}, 0, t) = \int_{0}^{t} |\mathbf{x}'(\tau)| \, \mathrm{d}\tau = \int_{0}^{t} \sqrt{\mathbf{x}'(\tau) \cdot \mathbf{x}'(\tau)} \, \mathrm{d}\tau.$$

The curves we deal with will always be smooth. Now, we want the definition of "smoothness" to describe an intrinsic property of Γ rather than a property of some parameterization $\mathbf{x}(s)$ of Γ. If a function $\mathbf{x}$ representing Γ is C^1, then the function L in equation (17.1) has a derivative with respect to s,

$$L'(t) = |\mathbf{x}'(t)|$$

that is continuous. Nevertheless, the curve itself may not conform to our idea of being smooth, which at a minimum requires a tangent at every point $\mathbf{y} \in \Gamma$. For example, the motion of a point on the boundary of a unit disk as it rolls along the x-axis is described by $\mathbf{x}(t) = (t - \sin t, 1 - \cos t)$, which is a C^∞ function.

Nevertheless, the curve has cusps at all multiples of 2π. The problem is that $\mathbf{x}'(2k\pi) = 0$.

Definition 17.2. *We say that a curve Γ admits an arc-length parameterization $s \in \mathbb{R} \mapsto \mathbf{x}(s)$ if the function $\mathbf{x}$ is C^1 and $L'(s) = |\mathbf{x}'(s)| = 1$ for all s. In case Γ is closed, we identify $[0, l(\Gamma)]$ algebraically with the circle group by adding elements of $[0, l(\Gamma)]$ modulo $l(\Gamma)$. We say that Γ is C^m, $m \in \mathbb{N}$, $m \geq 1$, if the arc-length parameterization $\mathbf{x}$ is a C^m function.*

Exercise 17.1. The aim of the exercise is to give a formula transforming a C^1 parameterization $t \in [0,1] \to \mathbf{x}(t)$ such that $|\mathbf{x}'(t)| \neq 0$ for all t into an arc-length parameterization. Notice that $L : [0,1] \to [0, L(1)]$ is increasing. Set, for $s \in [0, L(1)]$, $\tilde{\mathbf{x}}(s) = \mathbf{x}(L^{-1}(s))$ and check that $\tilde{\mathbf{x}}$ is an arc-length parameterization of the curve defined by $\mathbf{x}$. ■

An arc-length parameterization is also called a *Euclidean parameterization.* If a Jordan curve has an arc-length parameterization $\mathbf{x}$, then the domain of definition of $\mathbf{x}$ on the real line must be an interval $[a, b]$, where $b - a$ is the length of Γ, which we denote by $l(\Gamma)$. In this case, we will always take $[0, l(\Gamma)]$ as the domain of definition of $\mathbf{x}$.

Definition 17.3. *Assume that Γ is C^2 and let $s \mapsto \mathbf{x}(s)$ be an arc-length parameterization. The tangent vector $\boldsymbol{\tau}$ is defined as $\boldsymbol{\tau}(s) = \mathbf{x}'(s)$. The curvature vector of the curve Γ is defined by $\boldsymbol{\kappa}(s) = \mathbf{x}''(s)$. The normal vector $\mathbf{n}(s)$ is defined by $\mathbf{n}(s) = \boldsymbol{\tau}^{\perp}$, where $(x, y)^{\perp} = (-y, x)$.*

One can easily describe all Euclidean parameterizations of a Jordan curve.

Proposition 17.4. *Suppose that Γ is a C^1 Jordan curve with arc-length parameterization $\mathbf{x} : [0, l(\Gamma)] \to \Gamma$. Then any other arc-length parameterization $\mathbf{y} : [0, l(\Gamma)] \to \Gamma$ is of the form $\mathbf{y}(s) = \mathbf{x}(s + \sigma)$ or $\mathbf{y}(s) = \mathbf{x}(-s + \sigma)$ for some $\sigma \in [0, l(\Gamma)]$.*

Proof. Denote by C the interval $[0, l(\Gamma)]$, defined as an additive subgroup of $\mathbb{R}$ modulo $l(\Gamma)$. Let $\mathbf{x}$, $\mathbf{y} : C \mapsto \Gamma$ be two length preserving parameterizations of Γ. Then $f = \mathbf{x} \circ \mathbf{y}^{-1}$ is a length preserving bijection of C. Using the parameterization of C, this implies $f(s) = \pm s + \sigma$ for some $\sigma \in [0, l(\Gamma)]$ and the proof is easily concluded. (See exercise 17.7 for some more details.) □

Proposition 17.5. *Let Γ be a C^2 Jordan curve, and let $\mathbf{x}$ and $\mathbf{y}$ by any two arc-length parameterizations of Γ.*

(i) If $\mathbf{x}(s) = \mathbf{y}(t)$, then $\mathbf{x}'(s) = \pm\mathbf{y}'(t)$.

(ii) The vector $\boldsymbol{\kappa}$ is independent of the choice of arc-length parameterizations and it is orthogonal to $\boldsymbol{\tau} = \mathbf{x}'$.

Proof. By Proposition 17.4, $\mathbf{y}(s) = \mathbf{x}(\pm s + \sigma)$ and (i) follows by differentiation. This is also geometrically obvious: $\mathbf{x}'(s)$ and $\mathbf{y}'(t)$ are unit vectors tangent to Γ at the same point. Thus, they either point in the same direction or they point in opposite directions.

Using any of the above representations and differentiating twice shows that $\mathbf{x}'' = \mathbf{y}''$. Since $\mathbf{x}' \cdot \mathbf{x}' = 1$, differentiating this expression shows that $\mathbf{x}'' \cdot \mathbf{x}' = 0$. Thus, $\mathbf{x}''$ and $\mathbf{x}'$ are orthogonal and $\mathbf{x}''$ and $\mathbf{x}'^{\perp}$ are collinear. □

It will be convenient to have a flexible notation for the curvature in the different contexts we will use it. This is the object of the next definition.

Definition 17.6 (and notation). *Given a C^2 curve Γ, which is parameterized by length as $s \mapsto \mathbf{x}(s)$ and $\mathbf{x} = \mathbf{x}(s)$ a point of Γ, we denote in three equivalent ways the curvature of Γ at $\mathbf{x} = \mathbf{x}(s)$,*

$$\boldsymbol{\kappa}(\mathbf{x}) = \boldsymbol{\kappa}(\mathbf{x}(s)) = \boldsymbol{\kappa}(s) = \mathbf{x}''(s).$$

In the first notation, $\boldsymbol{\kappa}$ is the curvature of the curve Γ at a point $\mathbf{x}$ implicitly supposed to belong Γ. In the second notation a particular parameterization of Γ, $\mathbf{x}(s)$, is being used. In the third one, $\mathbf{x}$ is omitted.

The above notations create no ambiguity or contradiction, since by Proposition 17.5 the curvature is independent of the Euclidean parameterization. Of course, a smooth Jordan curve is locally a graph. More specifically:

Proposition 17.7. *A C^1 Jordan arc Γ can be represented around each one of its points $\mathbf{x}_0$ as the graph of a C^1 scalar function $y = f(x)$ such that $\mathbf{x}_0 = (0, f(0)) = (0, 0)$, $f'(0) = 0$, and*

$$\boldsymbol{\kappa}(\mathbf{x}_0) = (0, f''(0)). \tag{17.2}$$

Conversely, the graph of any C^1 function f is a C^1 Jordan arc. If f is C^2 the curvature of its associated Jordan curve satisfies (17.2) *at each point where $f'(0) = 0$.*

Proof. Assume we are given a C^1 Jordan arc Γ and an arc-length parameterization $\mathbf{c}$ in a neighborhood of $\mathbf{x}_0 = \mathbf{c}(s_0) \in \Gamma$. We assume, without loss of generality, that $s_0 = 0$. Then we can establish a local coordinate system with origin $\mathbf{x}_0$ and based on the two unit vectors $\mathbf{c}'(0)$ and $\mathbf{c}'(0)^{\perp}$ where the x-axis is positive in the direction of $\mathbf{c}'(0)$. If we write $\mathbf{c}(s) = (x(s), y(s))$ in this coordinate system, then

$$\begin{aligned} x(s) &= \mathbf{c}(s) \cdot \mathbf{c}'(0), \\ y(s) &= \mathbf{c}(s) \cdot \mathbf{c}'(0)^{\perp}. \end{aligned}$$

Since $\mathrm{d}x/\mathrm{d}s(s) = \mathbf{c}'(s) \cdot \mathbf{c}'(0)$, $\mathrm{d}x/\mathrm{d}s(0) = 1$. Then the inverse function theorem implies the existence of a C^1 function g and a $\delta > 0$ such that $s = g(x)$ for $|x| < \delta$. This means that, for $|x| < \delta$, Γ is represented locally by the graph of the C^1 function $f(x) = y(g(x)) = \mathbf{c}(g(x)) \cdot \mathbf{c}'(0)^{\perp}$. To be slightly more precise, denote the graph of f for $|x| < \delta$ by Γ_f. Since g is one-to-one, Γ_f is a homeomorphic image of the open interval $(-\delta, \delta)$ and $\Gamma_f \subset \Gamma$. If Γ is C^2, then f is C^2 and $f''(0) = \mathbf{c}''(0) \cdot \mathbf{c}'(0)^{\perp}$. Thus, on the local coordinate system, the coordinates of $c''(0) = \boldsymbol{\kappa}(x_0)$ are $(0, f''(0))$.

Conversely, given a C^1 function f, we can consider the graph Γ_f of f in a neighborhood of the origin. Then Γ_f is represented by $\mathbf{c}$, where $\mathbf{c}(x) = (x, f(x))$.

We may assume that $f(0)=0$ and $f'(0)=0$ (by a translation and rotation if necessary). The arc-length along Γ is measured by

$$s(x) = \int_0^x \sqrt{1+[f'(t)]^2}\,\mathrm{d}t,$$

and $s'(x) = \sqrt{1+[f'(x)]^2}$, so that $s'(0)=1$. This time there is a C^1 function h such that $h(s)=x$ and $h'(s) = (1+[f'(h(s))]^2)^{-1/2}$. Then Γ is represented by $\tilde{\mathbf{c}}(s) = (h(s), f(h(s)))$. Short computations show that $|\tilde{\mathbf{c}}'(s)| = 1$. If in addition f is C^2, then Γ is C^2 and it is an easy check that $\tilde{\mathbf{c}}''(0)\cdot\tilde{\mathbf{c}}'(0)^{\perp} = f''(0)$. □

Exercise 17.2. Make the above “short computations” and the “easy check”. ■

17.2 The structure of the set of level lines

We saw in Chapter 11 how an image can be represented by its level sets. The next step, with a view toward shape analysis, is the representation of an image in terms of its level lines. We rely heavily on the implicit function theorem to develop this representation. We begin with a two-dimensional version. The statement here is just a slight variation on the implicit function theorem quoted in section 1.

Theorem 17.8. *Let $u \in \mathcal{F}$ be a C^1 function such that $Du(\mathbf{x}_0) \neq 0$ at some $\mathbf{x}_0 = (x_0, y_0)$. Let $\boldsymbol{i}$ denote the unit vector in the direction (u_x, u_y), let $\boldsymbol{j}$ denote the unit vector in the orthogonal direction $(-u_y, u_x)$, and write $\mathbf{x} = \mathbf{x}_0 + x\boldsymbol{i} + y\boldsymbol{j}$. Then there is a disk $D(\mathbf{x}_0, r)$ and a unique C^1 function φ, $\varphi : [-r, r] \to \mathbb{R}$, such that if $\mathbf{x} \in D(\mathbf{x}_0, r)$, then*

$$u(x,y) = u(\mathbf{x}_0) \iff x = \varphi(y).$$

The following corollary is a global version of this local result.

Corollary 17.9. *Assume that $u \in \mathcal{F}$ is C^1 and let $u^{-1}(\lambda) = \{\mathbf{x} \mid u(\mathbf{x}) = \lambda\}$ for $\lambda \in \mathbb{R}$. If $\lambda \neq u(\infty)$ and $Du(\mathbf{x}) \neq 0$ for all $\mathbf{x} \in u^{-1}(\lambda)$, then $u^{-1}(\lambda)$ is a finite union of disjoint Jordan curves.*

Proof. From Theorem 17.8 we know that for each point $\mathbf{x} \in u^{-1}(\lambda)$ there is an open disk $D(\mathbf{x}, r(\mathbf{x}))$ such that $\overline{D}(\mathbf{x}, r(\mathbf{x})) \cap u^{-1}(\lambda)$ is a C^1 Jordan arc $\mathbf{x}(s)$ and we can take the endpoints of the arc on $\partial D(\mathbf{x}, r(\mathbf{x}))$. Since $\lambda \neq u(\infty)$, $u^{-1}(\lambda)$ is compact. Thus there is a finite number of points $\mathbf{x}_i$, $i = 1, \dots, m$, such that $u^{-1}(\lambda) \subset \bigcup_{i=1}^m D(\mathbf{x}_i, r(\mathbf{x}_i))$. This implies that $u^{-1}(\lambda)$ is a finite union of Jordan arcs which we can parameterize by length. The rest of the proof is very intuitive and is left to the reader. I consists of iteratively gluing the Jordan arcs until they close up into one or several Jordan curves. □

The next theorem is one of the few results that we are going to quote rather than prove, as we have done with the implicit function theorem.

Theorem 17.10 (Sard's theorem). *Let $u \in \mathcal{F} \cap C^1$. Then for almost every λ in the range of u, the set $u^{-1}(\lambda)$ is nonsingular, which means that for all $\mathbf{x} \in u^{-1}(\lambda)$, $Du(\mathbf{x}) \neq 0$.*

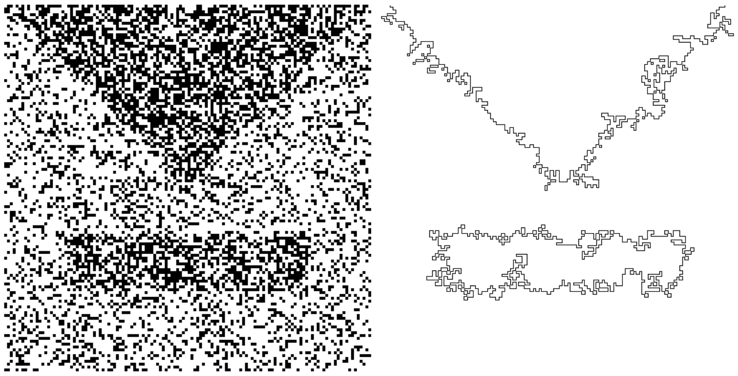

Figure 17.1: Level lines as representatives of the shapes present in an image. Left: noisy binary image with two apparent shapes; right: the two longest level lines.

As a direct consequence of Sard's Theorem and Corollary 17.9, we obtain:

Corollary 17.11. *Let $u \in \mathcal{F} \cap C^1$. Then for almost every λ in the range of u, the set $u^{-1}(\lambda)$ is the union of a finite set of disjoint simple closed C^1 curves.*

The sole purpose of the next proposition is to convince the reader that the level lines of a function provide a faithful representation of the function.

Proposition 17.12. *Let $u \in \mathcal{F} \cap C^1$. Then u can be reconstructed from the following data: the family of all of its level lines at nonsingular levels, the level of each level line being also kept.*

Proof. Let G be the closure of the union of the ranges of all level lines of u at nonsingular levels. If $\mathbf{x} \in G$, then there are points $\mathbf{x}_n$ belonging to level lines of some levels λ_n such that $\mathbf{x}_n \to \mathbf{x}$. As a consequence, $\lambda_n = u(\mathbf{x}_n) \to u(\mathbf{x})$. So we get back the value of $u(\mathbf{x})$.
Let now $\mathbf{x}$ belong to the open set G^c. Let us first prove that $Du(\mathbf{x}) = 0$. Assume by contradiction that $Du(\mathbf{x}) \neq 0$. By using the first order Taylor expansion of u around $\mathbf{x}$, one sees that for all $r > 0$ the connected range $u(B(\mathbf{x}, r))$ must contain some interval $(u(\mathbf{x}) - \alpha(r), u(\mathbf{x}) + \alpha(r))$ with $\alpha(r) \to 0$ as $r \to 0$. By Sard's theorem some of the values in this interval are nonsingular. Thus we can find nonsingular levels $\lambda_n \to u(\mathbf{x})$ and points $\mathbf{x}_n \to \mathbf{x}$ such that $u(\mathbf{x}_n) = \lambda_n$. This implies that $\mathbf{x} \in G$ and yields a contradiction.
Thus $Du(\mathbf{x}) = 0$ on G^c and u is therefore constant on each connected component A of G^c. The value of u is then uniquely determined by the value of u on the boundary of A. This value is known, since ∂A is contained in G. □

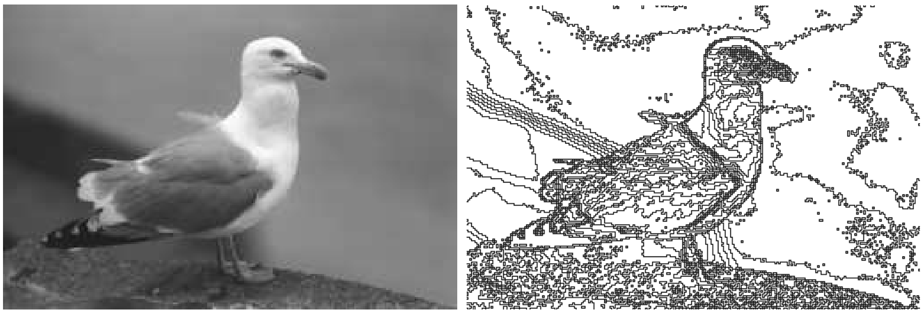

Figure 17.2: Level lines as a complete representation of the shapes present in an image. All level lines of the image of a sea bird for levels that are multiples of 12 are displayed. Notice that we do not need a previous smoothing to visualize the shape structures in an image: It is sufficient to quantize the displayed levels.

17.3 Curvature of the level lines

The intrinsic local coordinates

We continue to work in $\mathbb{R}^2$. Consider a real-valued function u that is twice continuously differentiable in a neighborhood of $\mathbf{x}_0 \in \mathbb{R}^2$. To simplify the notation, we will often write Du rather than $Du(\mathbf{x}_0)$, and so on.

Definition 17.13. *If $Du = (u_x, u_y) \neq 0$, then we establish a local coordinate system by letting $\boldsymbol{i} = Du/|Du|$ and $\boldsymbol{j} = Du^{\perp}/|Du|$, where $Du^{\perp} = (-u_y, u_x)$. Thus, for a point $\mathbf{x}$ near $\mathbf{x}_0$, we write $\mathbf{x} = \mathbf{x}_0 + x\boldsymbol{i} + y\boldsymbol{j}$ and the local coordinates of $\mathbf{x}$ are (x, y). (See Figure 17.3.) Without risk of ambiguity we shall write $u(x, y)$ for $u(\mathbf{x}) = u(\mathbf{x}_0 + x\boldsymbol{i} + y\boldsymbol{j})$.*

Since u is C^2, we can use Taylor's formula to express u in this coordinate system in a neighborhood of $\mathbf{x}_0$.

$$u(\mathbf{x}) = u(x, y) = u(\mathbf{x}_0) + px + ax^2 + by^2 + cxy + O(|\mathbf{x}|^3), \tag{17.3}$$

where $p = u_x(0, 0) = |Du(\mathbf{x}_0)| > 0$ and

$$\begin{aligned}
a &= \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(0,0) = \frac{1}{2}D^2u\Big(\frac{Du}{|Du|}, \frac{Du}{|Du|}\Big)(\mathbf{x}_0),\\
b &= \frac{1}{2}\frac{\partial^2 u}{\partial y^2}(0,0) = \frac{1}{2}D^2u\Big(\frac{Du^{\perp}}{|Du|}, \frac{Du^{\perp}}{|Du|}\Big)(\mathbf{x}_0),\\
c &= \frac{\partial^2 u}{\partial x \partial y}(0,0) = \quad D^2u\Big(\frac{Du^{\perp}}{|Du|}, \frac{Du}{|Du|}\Big)(\mathbf{x}_0).
\end{aligned} \tag{17.4}$$

Exercise 17.3. Check the three above formulas. ■

The implicit function theorem 17.8 ensures that in a neighborhood of $\mathbf{x}_0$ the set $\{\mathbf{x} \mid u(\mathbf{x}) = u(\mathbf{x}_0)\}$ is a C^2 graph whose equation can be written in the local coordinates $x = \varphi(y)$, where φ is a C^2 function in an interval I containing $y = 0$. In this interval, we have $u(\varphi(y), y) = u(\mathbf{x}_0)$. Differentiating this shows that $u_x\varphi' + u_y = 0$ for $y \in I$. Since $|Du(\mathbf{x}_0)| = u_x(0, 0)$ and $u_y(0, 0) = 0$

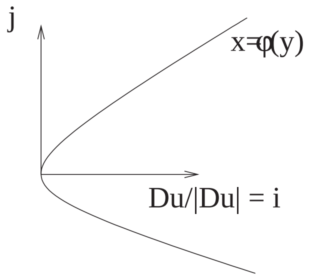

Figure 17.3: Intrinsic coordinates. Note that $\varphi''(0) > 0$, so $b < 0$.

in our coordinate system, we obtain $\varphi'(0) = 0$. A second differentiation of $u_x(\varphi(y), y)\varphi' + u_y(\varphi(y), y) = 0$ yields

$$(u_{xx}\varphi' + u_{xy})\varphi' + u_x\varphi'' + u_{yx}\varphi' + u_{yy} = 0.$$

Since $\varphi'(0) = 0$, we obtain $\varphi''(0) = -u_{yy}(0,0)/u_x(0,0)$. Using the notation of (17.4), one obtains

$$\varphi(y) = -\frac{b}{p}y^2 + o(y^2). \tag{17.5}$$

Equation (17.5) is the representation of the level line $\{\mathbf{x} \mid u(\mathbf{x}) = u(\mathbf{x}_0)\}$ in the intrinsic coordinates at $\mathbf{x}_0$. Let us set $|2b/p| = 1/R$. If the curve is a circle, R is the radius of this circle. More generally R is called *radius of the osculatory circle* to the curve. See exercise 17.11.

We are now going to do another simple computation to determine the curvature vector of the Jordan arc $\mathbf{c}$ defined by $\mathbf{c}(y) = \mathbf{x}_0 + \varphi(y)\boldsymbol{i} + y\boldsymbol{j}$ near $y = 0$. Recall that we denote the curvature of a curve $\mathbf{c}$ by $\boldsymbol{\kappa}(\mathbf{c})$ and the value of this function at a point $\mathbf{c}(y)$ by $\boldsymbol{\kappa}(\mathbf{c})(y)$.) Since in the local coordinates $\mathbf{c}'(y) = (\varphi'(y), 1)$ and $\mathbf{c}''(y) = (\varphi''(y), 0)$, at $y = 0$, we have $\mathbf{c}'(0) = (0,1)$ and $\mathbf{c}''(0) = (\varphi''(0), 0)$, so that $\mathbf{c}''(0) \cdot \mathbf{c}'(0) = 0$. Using this and the expression of the curvature in local graph coordinates (17.2) yields

$$\boldsymbol{\kappa}(\mathbf{c})(0) = (\varphi''(0), 0) = \varphi''(0)\frac{Du}{|Du|}(\mathbf{x}_0).$$

We now use (17.5) and (17.4) to write the last expression as

$$\boldsymbol{\kappa}(\mathbf{c})(0) = -\frac{1}{|Du|}D^2u\Big(\frac{Du^\perp}{|Du|}, \frac{Du^\perp}{|Du|}\Big)\frac{Du}{|Du|}(\mathbf{x}_0) \tag{17.6}$$

This tells us that the vectors $\boldsymbol{\kappa}(\mathbf{c})(0)$ and $Du(0)$ are collinear. Equation (17.6) also leads to the following definition and lemma introducing a scalar curvature.

Definition 17.14. *Let u be a real-valued function that is C^2 in a neighborhood of a point $\mathbf{x} \in \mathbb{R}^2$ and assume that $Du(\mathbf{x}) \neq 0$. The curvature of u at $\mathbf{x}$, denoted by* $\operatorname{curv}(u)(\mathbf{x})$*, is the real number defined in the local coordinates at $\mathbf{x}$ by*

$$\frac{1}{|Du|^3}D^2u(Du^\perp, Du^\perp)(\mathbf{x}) = \frac{u_{xx}u_y^2 - 2u_{xy}u_xu_y + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{3/2}}(0,0). \tag{17.7}$$

Exercise 17.4. Check the above identity. ■

Lemma 17.15. *Assume that* $u : \mathbb{R}^2 \to \mathbb{R}$ *is* C^2 *in a neighborhood of a point* $\mathbf{x}_0$ *and assume that* $Du(\mathbf{x}_0) \neq 0$*. Let* $N = N(\mathbf{x}_0)$ *be a neighborhood of* $\mathbf{x}_0$ *in which the iso-level set of* u*,* $\{\mathbf{x} \mid u(\mathbf{x}) = u(\mathbf{x}_0)\}$*, is a simple* C^2 *arc, which we still denote by* $\mathbf{x} = \mathbf{x}(s)$*. Then at every point* $\mathbf{x}$ *of this arc,*

$$\boldsymbol{\kappa}(\mathbf{x}) = -\mathrm{curv}(u)(\mathbf{x})\frac{Du}{|Du|}(\mathbf{x}). \tag{17.8}$$

Proof. This is an immediate consequence of (17.6) and (17.7). We need only remark that, given the hypotheses of the lemma, there is a neighborhood N of $\mathbf{x}_0$ such that $Du(\mathbf{x}) \neq 0$ for $\mathbf{x} \in N$ and such that $\{\mathbf{x} \mid u(\mathbf{x}) = u(\mathbf{x}_0)\}$ is a simple C^2 arc for $\mathbf{x} \in N$. Then the argument we made to derive (17.6) holds for any point $\mathbf{x} \in N \cap \{\mathbf{x} \mid u(\mathbf{x}) = u(\mathbf{x}_0)\}$. □

The next exercise proposes as a sanity check a verification that the curvature thus defined is contrast invariant and rotation invariant.

Exercise 17.5. Use equation (17.7) to show that

$$\mathrm{curv}(u) = \frac{\partial}{\partial x}\Big(\frac{Du}{|Du|}\Big) + \frac{\partial}{\partial y}\Big(\frac{Du}{|Du|}\Big) = \mathrm{div}\Big(\frac{Du}{|Du|}\Big). \tag{17.9}$$

Use this last relation to show that $\mathrm{curv}(g(u)) = \mathrm{curv}(u)$ if g is any C^2 function $g : \mathbb{R} \to \mathbb{R}$ such that $g'(x) > 0$ for all $x \in \mathbb{R}$. What happens if $g'(x) < 0$ for all $x \in \mathbb{R}$? Show that $\mathrm{curv}(U) = \mathrm{curv}(u)$, where $U(s,t) = u(x,y)$ and $x = s\cos\theta - t\sin\theta$, $y = s\sin\theta + t\cos\theta$. Check that $\mathrm{curv}(-u) = -\mathrm{curv}(u)$ and give a geometric interpretation to this relation. ■

Before leaving this section, we wish to emphasize geometric aspects of the functions we have introduced. Perhaps the most important fact is that the curvature of a C^2 Jordan arc Γ is an intrinsic property of Γ; it does not depend on the parameterization. If $\mathbf{x}$ is a point on Γ, then the curvature vector $\boldsymbol{\kappa}(\mathbf{x})$ points toward the center of the osculating circle. Furthermore, $1/|\boldsymbol{\kappa}(\mathbf{x})|$ is the radius of this circle, so when $|\boldsymbol{\kappa}(\mathbf{x})|$ is large, the osculating circle is small, and the curve is "turning a sharp corner."

If $Du(\mathbf{x}) \neq 0$, then the vector $Du(\mathbf{x})$ points in the direction of greatest increase, or steepest ascent, of u at $\mathbf{x}$: Following the gradient leads uphill. The function $\mathrm{curv}(u)$ does not have such a clear geometric interpretation, and it is perhaps best thought of in terms of equation (17.8): $\mathrm{curv}(u)(\mathbf{x})$ is the coefficient of $-Du(\mathbf{x})/|Du(\mathbf{x})|$ that yields the curvature vector $\boldsymbol{\kappa}(\mathbf{x})$ of the level curve through the point $\mathbf{x}$. We cannot over emphasize the importance of the two operators curv and Curv for the theories that follow. In addition to (17.8), a further relation between these operators is shown in Proposition 18.8, and it is this result that connects function smoothing with curve smoothing.

17.4 The principal curvatures of a level surface

We saw in Exercise 17.5 that $\mathrm{curv}(u)$ was contrast invariant. This idea will be generalized to $\mathbb{R}^N$ by introducing other differential operators that are contrast invariant. These operators will be functions of the *principal curvatures* of the level surfaces of u. For $\mathbf{z} \in \mathbb{R}^N$, $\mathbf{z}^\perp$ will denote the hyperplane $\{\mathbf{y} \mid \mathbf{z} \cdot \mathbf{y} = 0\}$

that is orthogonal to $\mathbf{z}$. (There should be no confusion with this notation and the same notation for $\mathbf{z} \in \mathbb{R}^2$. In $\mathbb{R}^2$, $\mathbf{z}^\perp$ is a vector orthogonal to $\mathbf{z}$, and the corresponding "hyperplane" is the line $\{t\mathbf{z}^\perp \mid t \in \mathbb{R}\}$.)

Proposition 17.16. *Assume that $u : \mathbb{R}^N \to \mathbb{R}$ is C^2 in a neighborhood of a point $\mathbf{x}_0$ and assume that $Du(\mathbf{x}_0) \neq 0$. Let $g : \mathbb{R} \to \mathbb{R}$ be a C^2 contrast change such that $g'(s) > 0$ for all $s \in \mathbb{R}$. Then $Dg(u(\mathbf{x}_0)) = g'(u(\mathbf{x}_0))Du(\mathbf{x}_0)$, and $\tilde{D}^2 g(u(\mathbf{x}_0)) = g'(u(\mathbf{x}_0))\tilde{D}^2 u(\mathbf{x}_0)$, where $\tilde{D}^2 u(\mathbf{x}_0)$ denotes the restriction of the quadratic form $D^2 u(\mathbf{x}_0)$ to the hyperplane $Du(\mathbf{x}_0)^\perp$. This means, in particular, that $(1/|Du(\mathbf{x}_0)|)\tilde{D}^2 u(\mathbf{x}_0)$ is invariant under such a contrast change.*

Proof. To simplify the notation, we will suppress the argument $\mathbf{x}_0$; thus, we write Du for $Du(\mathbf{x}_0)$, and so on. We use the notation $\mathbf{y} \otimes \mathbf{y}$, $\mathbf{y} \in \mathbb{R}^N$, to denote the linear mapping $\mathbf{y} \otimes \mathbf{y} : \mathbb{R}^N \to \mathbb{R}^N$ defined by $(\mathbf{y} \otimes \mathbf{y})(\mathbf{x}) = (\mathbf{x} \cdot \mathbf{y})\mathbf{y}$. The range of $\mathbf{y} \otimes \mathbf{y}$ is the one-dimensional space $\mathbb{R}\mathbf{y}$.

An application of the chain rule shows that $Dg(u) = g'(u)Du$. This implies that $Du^\perp = Dg(u)^\perp$. (Recall that $g'(s) > 0$ for all $s \in \mathbb{R}$.) A second differentiation shows that

$$D^2 g(u) = g''(u)Du \otimes Du + g'(u)D^2 u.$$

If $\mathbf{y} \in Du^\perp$, then $(Du \otimes Du)(\mathbf{y}) = 0$ and $D^2 g(u)(\mathbf{y}, \mathbf{y}) = g'(u)D^2 u(\mathbf{y}, \mathbf{y})$. This means that $D^2 g(u) = g'(u)D^2 u$ on $Du^\perp = Dg(u)^\perp$, which proves the result. □

Exercise 17.6. Taking euclidian coordinates, give the matrix of $\mathbf{y} \otimes \mathbf{y}$. Check the above differentiations. ■

We are now going to define locally the level surface of a smooth function u, and for this we quote one more version of the implicit function theorem, in arbitrary dimension N.

Theorem 17.17 (Implicit function theorem). *Assume that $u : \mathbb{R}^N \to \mathbb{R}$ is C^m in the neighborhood of $\mathbf{x}_0$ and assume that $Du(\mathbf{x}_0) \neq 0$. Write $\mathbf{x} = \mathbf{x}_0 + \mathbf{y} + z\boldsymbol{i}$, where $\boldsymbol{i} = Du(\mathbf{x}_0)/|Du(\mathbf{x}_0)|$ and $\mathbf{y} \in Du(\mathbf{x}_0)^\perp$. Then there exists a ball $B(\mathbf{x}_0, \rho)$ and a unique real-valued C^m function φ defined on $B(\mathbf{x}_0, \rho) \cap \{\mathbf{x} \mid \mathbf{x} = \mathbf{x}_0 + \mathbf{y}, \boldsymbol{i} \cdot \mathbf{y} = 0\}$ such that for every $\mathbf{x} \in B(\mathbf{x}_0, \rho)$*

$$u(\mathbf{x}) = u(\mathbf{x}_0) \iff \varphi(\mathbf{y}) = z.$$

In other words, the equation $\varphi(\mathbf{y}) = z$ describes the set $\{\mathbf{x} \mid u(\mathbf{x}) = u(\mathbf{x}_0)\}$ near $\mathbf{x}_0$ as the graph of a C^m function φ. Thus, locally we have a surface passing through $\mathbf{x}_0$ that we call the *level surface of u around* $\mathbf{x}_0$.

We are going to use Proposition 17.16 and Theorem 17.17, first, to give a simple intrinsic representation for the level surface of a function u around a point $\mathbf{x}_0$ and, second, to relate the eigenvalues of the quadratic form introduced in Proposition 17.16 to the curvatures of lines drawn on the level surface of u.

Proposition 17.18. *Assume that $u : \mathbb{R}^N \to \mathbb{R}$ is C^2 in a neighborhood of $\mathbf{x}_0 \in \mathbb{R}^N$ and that $p = Du(\mathbf{x}_0) \neq 0$. Denote the eigenvalues of the restriction of the quadratic form $D^2 u(\mathbf{x}_0)$ to the hyperplane $Du(\mathbf{x}_0)^\perp$ by $\mu_1, \dots, \mu_{N-1}$. Let $\boldsymbol{i}_N = Du(\mathbf{x}_0)/|Du(\mathbf{x}_0)|$ and select $\boldsymbol{i}_1, \dots, \boldsymbol{i}_{N-1}$ so they form an orthonormal* *basis of eigenvectors of the restriction of* $D^2u(\mathbf{x}_0)$ *to* $Du(\mathbf{x}_0)^\perp$. *Write* $\mathbf{x} = \mathbf{x}_0 + \mathbf{z}$, *where* $\mathbf{z} = x_1\boldsymbol{i}_1 + \cdots + x_N\boldsymbol{i}_N = \mathbf{y} + x_N\boldsymbol{i}_N$. *Then if* $|\mathbf{z}|$ *is sufficiently small, the function* $\varphi(\mathbf{y}) = x_N$ *that solves the equation* $u(\mathbf{y}, \varphi(\mathbf{y})) = u(\mathbf{x}_0)$ *can be expressed locally as*

$$x_N = \frac{-1}{2p}\sum_{i=1}^{N-1}\mu_i x_i^2 + o(|\mathbf{y}|^2).$$

Proof. Assume, without loss of generality, that $\mathbf{x}_0 = 0$ and that $u(0) = 0$. Using the notation of Theorem 17.17, for $\mathbf{x} \in B(0,\rho)$, $u(\mathbf{y}, x_N) = 0$ if and only if $\varphi(\mathbf{y}) = x_N$, and φ is C^2 in $B(0,\rho)$. Furthermore, by differentiating the expression $u(\mathbf{y}, \varphi(\mathbf{y})) = 0$, we see that $u_{x_i} + u_{x_N}\varphi_{x_i} = 0$, $i = 1, \dots, N-1$ for $|\mathbf{x}| < \rho$. In particular, $u_{x_i}(0) + u_{x_N}(0)\varphi_{x_i}(0) = 0$. In the local coordinate system we have chosen, $|Du(0)| = |u_{x_N}(0)|$, and since $Du(0) \neq 0$, we conclude that $u_{x_i}(0) = 0$ for $i = 1, \dots, N-1$ and hence that $\varphi_{x_i}(0) = 0$ for $i = 1, \dots, N-1$. This means that the local expansion of φ has the form

$$\varphi(\mathbf{y}) = \frac{1}{2}D^2\varphi(0)(\mathbf{y}, \mathbf{y}) + o(|\mathbf{y}|^2).$$

Now differentiate the relation $u_{x_i} + u_{x_N}\varphi_{x_i} = 0$ again to obtain

$$u_{x_ix_j} + u_{x_ix_N}\varphi_{x_j} + (u_{x_Nx_j} + u_{x_Nx_N}\varphi_{x_j})\varphi_{x_i} + u_{x_N}\varphi_{x_ix_j} = 0.$$

Since we have just shown that $\varphi_{x_i}(0) = 0$ for $i = 1, \dots, N-1$, we see from this last expression that $\tilde{D}^2u(0) + p\tilde{D}^2\varphi(0) = 0$, where $p = u_{x_N}(0)$ and $\tilde{D}^2u(0)$, $\tilde{D}^2\varphi(0)$ are the restrictions of the quadratic forms $D^2u(0)$ and $D^2\varphi(0)$ to the hyperplane $Du(0)^\perp$. Thus we have

$$x_N = \frac{-1}{2p}D^2u(0)(\mathbf{y}, \mathbf{y}) + o(|\mathbf{y}|^2). \tag{17.10}$$

Recall that $\mathbf{y} \in Du(0)^\perp$ and that $\mathbf{y} = x_1\boldsymbol{i}_1 + \cdots + x_{N-1}\boldsymbol{i}_{N-1}$, where the $\boldsymbol{i}_i$ are an orthonormal basis of eigenvectors of $D^2(0)$ restricted to $Du(0)^\perp$. Thus,

$$x_N = \frac{-1}{2p}\sum_{i=1}^{N-1}\mu_i x_i^2 + o(|\mathbf{y}|^2),$$

which is what we wished to prove. □

This formula reads

$$x_2 = \frac{-1}{2p}\mu_1 x_1^2 + o(|x_1|^2)$$

if $N = 2$, which is just equation (17.5) with different notation. Thus, $\mu_1 = |Du|\text{curv}(u)$, confirming that $\mu_1 = \partial^2 u/\partial x_1^2$. We are now going to use our two-dimensional analysis to give a further interpretation of the eigenvalues μ_i for $N > 2$. We begin by considering the curve $\Gamma_{\boldsymbol{\nu}}$ defined by the two equations $\mathbf{x} = \mathbf{x}_0 + t\boldsymbol{\nu} + x_N\boldsymbol{i}_N$ and $\varphi(t\boldsymbol{\nu}) = x_N$, where $\boldsymbol{\nu}$ is a unit vector in $Du(\mathbf{x}_0)^\perp$. Their solution in the local coordinates is $\varphi(t\boldsymbol{\nu}) = x_N$, whenever $t \in \mathbb{R}$ is small. Thus, Γ_ν is a curve passing by $\mathbf{x}_0$, drawn on the level surface of u and projecting into a straight line of $Du^\perp$. By (17.10) its equation is

$$x_N = \varphi(t\boldsymbol{\nu}) = \frac{-1}{2|Du(\mathbf{x}_0)|}D^2u(\mathbf{x}_0)(\boldsymbol{\nu}, \boldsymbol{\nu})t^2 + o(t^2),$$

and its normal at $\mathbf{x}_0$ is $\frac{Du(\mathbf{x}_0)}{|Du(\mathbf{x}_0)|}$. Thus the curvature vector of Γ_ν at $\mathbf{x}_0$ is

$$\boldsymbol{\kappa}(\Gamma_{\boldsymbol{\nu}})(\mathbf{x}_0) = \frac{-1}{|Du(\mathbf{x}_0)|} D^2u(\mathbf{x}_0)(\boldsymbol{\nu},\boldsymbol{\nu}) \frac{Du(\mathbf{x}_0)}{|Du(\mathbf{x}_0)|}.$$

By defining $\kappa_{\boldsymbol{\nu}} = |Du(\mathbf{x}_0)|^{-1} D^2u(\mathbf{x}_0)(\boldsymbol{\nu},\boldsymbol{\nu})$, we have

$$\boldsymbol{\kappa}(\Gamma_{\boldsymbol{\nu}})(\mathbf{x}_0) = -\kappa_{\boldsymbol{\nu}} \frac{Du(\mathbf{x}_0)}{|Du(\mathbf{x}_0)|},$$

which has the same form as equation (17.8). So the modulus of $\kappa_{\boldsymbol{\nu}}$ is equal to the modulus of the curvature of Γ_ν at $\mathbf{x}_0$. This leads us to call principal curvatures of the level surface of u at $\mathbf{x}_0$ the numbers $\kappa_{\boldsymbol{\nu}}$ obtained by letting $\boldsymbol{\nu} = \boldsymbol{i}_j$, $j = 1, \dots, N-1$, where the unit vectors $\boldsymbol{i}_j$ are an orthonormal system of eigenvectors of $D^2u(\mathbf{x}_0)$ restricted to $Du(\mathbf{x}_0)^\perp$.

Definition 17.19. *Let $u : \mathbb{R}^2 \to \mathbb{R}$ be C^2 at $\mathbf{x}_0$, with $Du(\mathbf{x}_0) \neq 0$. The principal curvatures of u at $\mathbf{x}_0$ are the real numbers*

$$\kappa_j = \frac{\mu_j}{|Du(\mathbf{x}_0)|},$$

where μ_j are the eigenvalues of $D^2u(\mathbf{x}_0)$ restricted to $Du(\mathbf{x}_0)^\perp$.

It follows from Proposition 17.16 that the principal curvatures are invariant under a C^2 contrast change g such that $g'(s) > 0$ for all $s \in \mathbb{R}$.

Definition 17.20. *The mean curvature of a C^2 function $u : \mathbb{R}^N \to \mathbb{R}$ at $\mathbf{x}_0 \in \mathbb{R}^N$ is the sum of the principal curvatures at $\mathbf{x}_0$. It is denoted by $\mathrm{curv}(u)(\mathbf{x}_0)$.*

Note that this definition agrees with Definition 17.2 when $N = 2$. The next result provides another representation for $\mathrm{curv}(u)$.

Proposition 17.21. *The mean curvature of u is given by*

$$\mathrm{curv}(u) = \mathrm{div}\Big(\frac{Du}{|Du|}\Big).$$

Proof. Represent the matrix D^2u in the coordinate system $\boldsymbol{i}_j$, $j = 1, \dots, N-1$, and $\boldsymbol{i}_N = Du(\mathbf{x}_0)/|Du(\mathbf{x}_0)|$, where the $\boldsymbol{i}_j$, $j = 1, \dots, N-1$, form a complete set of eigenvectors of the linear mapping $D^2u(\mathbf{x}_0)$ restricted to $Du^\perp(\mathbf{x}_0)$. Then in this coordinate system, $D^2u(\mathbf{x}_0)$ has the following form (illustrated for $N = 5$):

$$D^2u(\mathbf{x}_0) = \begin{bmatrix} u_{11} & 0 & 0 & 0 & u_{15} \\ 0 & u_{22} & 0 & 0 & u_{25} \\ 0 & 0 & u_{33} & 0 & u_{35} \\ 0 & 0 & 0 & u_{44} & u_{45} \\ u_{51} & u_{52} & u_{53} & u_{54} & u_{55} \end{bmatrix},$$

where $u_{jk} = u_{x_jx_k}(\mathbf{x}_0)$, and $u_{jj} = \kappa_j$ is the eigenvalue associated with $\boldsymbol{i}_j$. Thus, by definition, we see that

$$\mathrm{curv}(u) = \frac{\Delta u}{|Du|} - \frac{1}{|Du|} D^2u\Big(\frac{Du}{|Du|}, \frac{Du}{|Du|}\Big).$$

We also have

$$\begin{aligned}
\operatorname{div}\Big(\frac{Du}{|Du|}\Big) &= \sum_{j=1}^{N} \frac{\partial}{\partial x_j}\Big(\frac{u_{x_j}}{|Du|}\Big) \\
&= \frac{1}{|Du|}\sum_{j=1}^{N} u_{x_j x_j} - \frac{1}{|Du|^3}\sum_{j,k=1}^{N} u_{x_j x_k} u_{x_j} u_{x_k} \\
&= \frac{\Delta u}{|Du|} - \frac{1}{|Du|} D^2 u\Big(\frac{Du}{|Du|}, \frac{Du}{|Du|}\Big). \qquad \square
\end{aligned}$$

□

With this representation, it is clear that the mean curvature has the same invariance properties as the curvature of a C^2 function defined on $\mathbb{R}^2$. (See Exercise 17.5.)

17.5 Exercises

Exercise 17.7. Let Γ by a Jordan arc parameterized by $\mathbf{x} : [0,1] \to \Gamma$ and by $\mathbf{y} : [0,1] \to \Gamma$. Show that $\mathbf{x} = \mathbf{y} \circ f$ or $\mathbf{x} = \mathbf{y} \circ (1-f)$, where f is a continuous, strictly increasing function that maps $[0,1]$ onto $[0,1]$. Hint: $\mathbf{x}$ and $\mathbf{y}$ are one-to-one, and since $[0,1]$ is compact, they are homeomorphisms. Thus, $\mathbf{y}^{-1}(\mathbf{x}) = f$ is a one-to-one continuous mapping of $[0,1]$ onto itself. As an application, give a proof of Proposition 17.4. ■

Exercise 17.8. State and prove an adaptation of Propositions 17.4 and 17.5 to a Jordan arc. ■

The curvature vector has been defined in terms of the arc length. Curves, however, are often naturally defined in terms of other parameters. The next two exercises develop the differential relations between an arc-length parameterization and another parameterization.

Exercise 17.9. Assume that Γ is a C^2 Jordan arc or curve. Let $s \mapsto \mathbf{x}(s)$ be an arc-length parameterization and let $t \mapsto \mathbf{y}(t)$ be any other parameterization with the property that $\mathbf{y}'(t) \neq 0$. Since $\mathbf{x}$ and $\mathbf{y}$ are one-to-one, we can consider the function $\mathbf{y}^{-1}(\mathbf{x}) = \varphi$. Then $\mathbf{x}(s) = \mathbf{y}(\varphi(s))$, where $\varphi(s) = t$. The inverse function φ^{-1} is given by

$$s = \varphi^{-1}(t) = \int_{t_0}^{t} \sqrt{\mathbf{y}'(r)\cdot\mathbf{y}'(r)}\,\mathrm{d}r,$$

so we know immediately that φ^{-1} is absolutely continuous with continuous derivative equal to $\sqrt{\mathbf{y}'(t)\cdot\mathbf{y}'(t)}$. Thus, we also know that $\varphi'(s) = |\mathbf{y}'(\varphi(s))|^{-1}$. Note that we made a choice above by taking $\sqrt{\mathbf{y}'(r)\cdot\mathbf{y}'(r)}$ to be positive. This is equivalent to assuming that $\mathbf{x}'(s)$ and $\mathbf{y}'(\varphi(s))$ point in the same direction or that $\varphi'(s) > 0$.

(i) Show that $\boldsymbol{\kappa}(s) = \mathbf{x}''(s) = \mathbf{y}''(\varphi(s))[\varphi'(s)]^2 + \mathbf{y}'(\varphi(s))\varphi''(s)$ and deduce that

$$\varphi''(s) = -\frac{\mathbf{y}''(\varphi(s))\varphi'(s)\cdot\mathbf{y}'(\varphi(s))}{|\mathbf{y}'(\varphi(s))|^3} = -\frac{\mathbf{y}''(\varphi(s))\cdot\mathbf{y}'(\varphi(s))}{|\mathbf{y}'(\varphi(s))|^4}.$$

(ii) Use the results of (i) to show that

$$\boldsymbol{\kappa}(s) = \mathbf{x}''(s) = \frac{1}{|\mathbf{y}'(t)|^2}\left[\mathbf{y}''(t) - \left(\mathbf{y}''(t)\cdot\frac{\mathbf{y}'(t)}{|\mathbf{y}'(t)|}\right)\frac{\mathbf{y}'(t)}{|\mathbf{y}'(t)|}\right], \qquad (17.11)$$

where $\varphi(s) = t$. Show that we get the same expression for the right-hand side of (17.11) with the assumption that $\varphi'(s) < 0$. This shows that the curvature vector $\boldsymbol{\kappa}$ does not depend on the choice of parameter.

(iii) Consider the scalar function $\kappa(\mathbf{y})$ defined by $\kappa(\mathbf{y})(s) = \boldsymbol{\kappa}(s)\cdot\mathbf{x}'(s)^{\perp}$. Use equation (17.11) to show that

$$\kappa(\mathbf{y})(t) = \frac{\mathbf{y}''(t)\cdot[\mathbf{y}'(t)]^{\perp}}{|\mathbf{y}'(t)|^3}$$

Note that $\kappa(\mathbf{y})$ is determined up to a sign that depends on the sign of $\varphi'(s)$; however, $|\kappa(\mathbf{y})| = |\boldsymbol{\kappa}|$ is uniquely determined. ■

Exercise 17.10. Assume that Γ is a Jordan arc or curve that is represented by a C^1 function $t \mapsto \mathbf{x}(t)$ with the property that $\mathbf{x}'(t) \neq 0$. Prove that Γ is C^1.

Exercise 17.11.

(i) Consider the arc-length parameterization of the circle with radius r centered at the origin given by $\mathbf{x}(s) = (r\cos(s/r), r\sin(s/r))$. Show that the length of the curvature vector is $1/r$.

(ii) Compute the scalar curvature of the graph of $y = (a/2)x^2$ at $x = 0$. ■

Exercise 17.12. Complete the proof of Corollary 17.9.

Exercise 17.13. The kinds of techniques used in this exercise are important for work in later chapters. The exercise demonstrates that it is possible to bracket a C^2 function locally with two functions that are radial and either increasing or decreasing. We say that a function f is radial and increasing if there exists an increasing function $g : \mathbb{R}^+ \to \mathbb{R}$ such that $f(\mathbf{x}) = g(|\mathbf{x}_c - \mathbf{x}|^2)$, $\mathbf{x}_c \in \mathbb{R}^2$. We say that f is radial and decreasing if g is decreasing. Let $u : \mathbb{R}^2 \to \mathbb{R}$ be C^2 and assume that $Du(\mathbf{x}_0) \neq 0$. We wish to show that for every $\varepsilon > 0$ there exist two C^2 radial functions f_ε^- and f_ε^+ (increasing or decreasing, depending on the situation) that satisfy the following four conditions:

$$f_\varepsilon^-(\mathbf{x}_0) = u(\mathbf{x}_0) = f_\varepsilon^+(\mathbf{x}_0), \tag{17.12}$$

$$Df_\varepsilon^-(\mathbf{x}_0) = Du(\mathbf{x}_0) = Df_\varepsilon^+(\mathbf{x}_0), \tag{17.13}$$

$$\mathrm{curv}(f_\varepsilon^-)(\mathbf{x}_0) + \frac{2\varepsilon}{p} = \mathrm{curv}(u)(\mathbf{x}_0) = \mathrm{curv}f_\varepsilon^+(\mathbf{x}_0) - \frac{2\varepsilon}{p}, \tag{17.14}$$

$$f_\varepsilon^-(\mathbf{x}) + o(|\mathbf{x}_0 - \mathbf{x}|^2) \le u(\mathbf{x}) \le f_\varepsilon^+(\mathbf{x}) + o(|\mathbf{x}_0 - \mathbf{x}|^2). \tag{17.15}$$

1. Without loss of generality, take $\mathbf{x}_0 = (0,0)$, $u(0,0) = 0$, and $Du(\mathbf{x}_0) = (p, 0)$, $p > 0$. Then we have the Taylor expansion

$$u(\mathbf{x}) = px + ax^2 + by^2 + cxy + o(x^2 + y^2),$$

where a, b, and c are given in (17.4). Show that for every $\varepsilon > 0$,

$$px+\Big(-\frac{c^2}{\varepsilon}+a\Big)x^2+(b-\varepsilon)y^2+o(x^2+y^2) \le u(x,y) \le px+\Big(\frac{c^2}{\varepsilon}+a\Big)x^2+(b+\varepsilon)y^2+o(x^2+y^2).$$

2. Let f be a radial function defined by $f(x,y) = g((x-x_c)^2 + y^2)$, where $g : \mathbb{R}^+ \to \mathbb{R}$ is C^2 and either increasing or decreasing. Show by expanding f at $(0,0)$ that

$$f(x,y) = g(x_c^2) - 2x_c g'(x_c^2)x + (2x_c^2 g''(x_c^2) + g'(x_c^2))x^2 + g'(x_c^2)y^2 + o(x^2 + y^2).$$

3. The idea is to construct f_ε^+ and f_ε^- by matching the coefficients of the expansion of f with the coefficients of the functions $px + (\pm(c^2/\varepsilon) + a)x^2 + (b \pm \varepsilon)y^2$. There are three cases to consider: $b < 0$, $b = 0$, and $b > 0$. Show that in each case it is possible to find values of x_c and functions g so the functions f_ε^+ and f_ε^- satisfy the four condition. Note that both x_c and g depend on ε. Discuss the geometry for each case. ■

Exercise 17.14. By computing explicitly the terms $\partial g(u)/\partial x_i$, verify that $Dg(u) = g'(u)Du$. Similarly, verify that $D^2(g(u)) = g''(u)Du \otimes Du + g'(u)D^2u$ by computing the second-order terms $\partial^2 g(u)/\partial x_i \partial x_j$. ■

17.6 Comments and references

Calculus and differential geometry. The differential calculus of curves and surfaces used in this chapter can be found in many books, and no doubt most readers are familiar with this material. Nevertheless, a few references to specific results may be useful. As a general reference on calculus, and as a specific reference for the implicit function theorem, we suggest the text by Courant and John [57]. (The implicit function theorem can be found on page 221 of volume II.) Elementary results about classical differential geometry can be found in [237]. A statement and proof of Sard's theorem can be found in [138].

Level lines. An introduction to the use of level lines in computer vision can be found in [45]. A complete discussion of the definition of level lines for BV functions can be found in [11]. One can decompose an image into into its level lines at quantized levels and conversely reconstruct the image from this topographic map. A fast algorithm, the Fast Level Set Transform (FLST) performing these algorithms is described in [148]. Its principle is very simple: a) perform the bilinear interpolation, b) rule out all singular levels where saddle point occur c) quantize the other levels, in which the level lines are finite unions of parametric Jordan curves. The image is then parsed into a set of parametric Jordan curves. This set is easily ordered in a tree structure, since two Jordan level curves do not meet. Thus either one surrounds the other one or conversely. The level lines tree is a shape parser for the image, many level lines surrounding perceptual shapes or parts of perceptual shapes.

Curvature. It is a well-known mathematical technique to define a set implicitly as the zero set of its distance function. In case the set is a curve, one can compute its curvature at a point $\mathbf{x}$ by computing the curvature $\mathrm{curv}(u)(\mathbf{x})$, where u is a signed distance function of the curve. This yields an intrinsic formula for the curvature that is not dependent on a parameterization of the curve. The same technique has been applied in recent years as a useful numerical tool. This started with Barles report on flame propagation [24] and was extended by Sethian [231] and by Osher and Sethian [199] in a series of papers on the numerical simulation of the motion of a surface by its mean curvature.

Chapter 18

The Main Curvature Equations

The purpose of this chapter is to introduce the curvature motion PDE's for Jordan curves and images. Our main task is to establish a formal link between curve evolution and image evolution. This link will be established through the PDE formulation. The basic differential geometry used in this chapter was thoroughly developed in Chapter 17, which must therefore be read first.

18.1 The definition of a shape and how it is recognized

Relevant information in images has been reduced to the image level sets in Chapter 11. By Corollary 17.9, if the image is C^1, the boundary of its level sets is a finite set Jordan curves at almost every level. Thus, shape analysis can be led back to the study of these curves which we shall call "elementary shapes".

Definition 18.1. *We call elementary shape any C^1 planar Jordan curve.*

The many experiments where we display level lines of digital images make clear enough why a smoothing is necessary to restore their structure. These experiments also show that we can in no way assimilate these level lines with our common notion of shape as the silhouette of a physical object in full view. Indeed, in images of a natural environment, most observed objects are partially hidden (occluded) by other objects and often deformed by perspective. When we observe a level line we cannot be sure that it belongs to a single object; it may be composed of pieces of the boundaries of several objects that are occluding each other. Shape recognition technology has therefore focused on local methods, that is, methods that work even if a shape is not in full view or if the visible part is distorted. As a consequence, image analysis adopts the following principle: *Shape recognition must be based on local features of the shape's boundary, in this case local features of the Jordan curve, and not on its global features. If the boundary has some degree of smoothness, then these local features are based on the derivatives of the curve, namely the tangent vector, the curvature, and* *so on.* Many local recognition methods involve the "salient" points of a shape, which are the points where the curvature is zero (inflection points) and points where the curvature has a maximum or minimum (the "corners" of the shape).

18.2 Multiscale features and scale space

Computational shape recognition methods often make the following two basic assumptions, neither of which is true in practice for the rough shape data:

- The shape is a smooth Jordan curve.
- The boundary has a small number of inflexion points and curvature extrema. This number can be made as small as desired by smoothing.

The fact that these conditions can be obtained by properly smoothing a C^1 Jordan curve was proven in 1986-87 by Gage and Hamilton [90] and Grayson [98]. They showed that it is possible to transform a C^1 Jordan curve into a C^∞ Jordan curve by using the so-called *intrinsic heat equation.*

For convenience, and unless it would cause ambiguity, we will not make a distinction between a Jordan curve Γ as a subset of the plane and a function $s \mapsto \mathbf{x}(s)$ such that $\Gamma = \{\mathbf{x}(s)\}$. As we have already done, we will speak of the Jordan curve $\mathbf{x}$. Since we will be speaking of families of Jordan curves dependent on a parameter $t > 0$, we will most often denote these families by $\mathbf{x}(t, s)$, where the second variable is a parameterization of the Jordan curve. Thus, $\mathbf{x}(t, s)$ has three meanings: a family of Jordan curves, a family of functions that represent these curves, and a particular point on one of these curves. The notation s will be usually reserved to an arc-length parameter.

Definition 18.2. *Let $\mathbf{x}(t)$, $t > 0$, be a family of C^2 Jordan curves. We say that $\mathbf{x}(t)$ satisfies the intrinsic heat equation if*

$$\frac{\partial \mathbf{x}}{\partial t} = \boldsymbol{\kappa}(\mathbf{x}(t)). \tag{18.1}$$

Theorem 18.3 (Grayson). *Let $\mathbf{x}_0$ be a C^1 Jordan curve. By using the intrinsic heat equation, it is possible to evolve $\mathbf{x}_0$ into a family of Jordan curves $\mathbf{x}(t, s)$ such that $\mathbf{x}(0, s) = \mathbf{x}_0(s)$ and such that for every $t > 0$, $\mathbf{x}(t, s)$ is C^∞ (actually analytical) and satisfies the equation* (18.1)*. Furthermore, for every $t > 0$, $\mathbf{x}(t, s)$ has only a finite number of inflection points and curvature extrema, and the number of these points does not increase with t. For every initial curve, there is a scale t_0 such that the curve $\mathbf{x}(t, s)$ is convex for $t \geq t_0$ and there is a scale t_1 such that the curve $\mathbf{x}(t, s)$ is a single point for $t \geq t_1$.*

It is time to say what we mean by "curve scale space", or "shape scale space." We will refer to any process that smooths a Jordan curve and that depends on a real parameter t. Thus a shape scale space associates with an initial Jordan curve $\mathbf{x}(0, s) = \mathbf{x}_0(s)$ a family of smooth curves $\mathbf{x}(t, s)$. For example, the intrinsic heat equation eliminates spurious details of the initial shape and retains simpler, more reliable versions of the shape, and these smoothed shapes have finite codes. A scale space is *causal* in the terminology of vision theory if it does not introduce new features. Grayson's theorem therefore defines a causal scale space.

18.3 From image motion to curve motion

The intrinsic heat equation is only one example from a large family of nonlinear equations that move curves with a curvature-dependent speed, that is, $\partial \mathbf{x}/\partial t$ is a function of the curvature of the curve $\mathbf{x}$. There are two conditions on the curvature dependence. The velocity vector must always point towards the concavity of the curve. Its norm must be a nondecreasing function of the magnitude of the curvature $|\boldsymbol{\kappa}(\mathbf{x})|$. The first condition ensures that the equation is a smoothing which reduces asperities. The second condition intuitively preserves the inclusion between curves. This can be appreciated by considering two circles C and C' such that C' surround C and C' and C are tangent at some point $\mathbf{x}$. Then the first condition implies that both circles shrink, but the second condition implies that the smaller circle C shrinks faster, so that the inclusion between circles is preserved by the evolution.

Definition 18.4. *We say that a C^2 function $u : \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}$ satisfies a curvature equation if for some real-valued function $g(\kappa, t)$, which is nondecreasing in κ and satisfies $g(0,t) = 0$,*

$$\frac{\partial u}{\partial t}(t, \mathbf{x}) = g(\mathrm{curv}(u)(t, \mathbf{x}), t)|Du|(t, \mathbf{x}). \tag{18.2}$$

Definition 18.5. *Let $\mathbf{x}(t)$ be a family of C^2 Jordan curves. We say that the functions $\mathbf{x}(t)$ satisfy a curvature equation if for some real-valued function $g(\kappa, t)$ nondecreasing in κ with $g(0,t) = 0$, they satisfy*

$$\frac{\partial \mathbf{x}}{\partial t} = g(|\boldsymbol{\kappa}(\mathbf{x})|, t)\mathbf{n}(t), \tag{18.3}$$

where $\mathbf{n}$ is a unit vector in the direction of $\boldsymbol{\kappa}(\mathbf{x})$.

In the preceding definition, the equation makes sense if $\boldsymbol{\kappa}(\mathbf{x}) = 0$ since then the second member is zero. As we shall see, these equations are the only candidates to be curve or image scale spaces, and one of the objectives of this book is to identify which forms for g are particularly relevant for image analysis. The above definitions are quite restrictive because they require the curves or images to be C^2. A more generally applicable definition of solutions for these equations will be given in Chapter **??** with the introduction of viscosity solutions. Our immediate objective is to establish the link between the motion of an image and the motion of its level lines. This will establish the relation between equations (18.2) and (18.3).

18.3.1 A link between image and curve evolution

Lemma 18.6. (Definition of the "normal flow"). *Suppose that $(t, \mathbf{x}) \mapsto u(t, \mathbf{x})$ is C^2 in a neighborhood $T \times U$ of the point $(t_0, \mathbf{x}_0) \in \mathbb{R} \times \mathbb{R}^2$, and assume that $Du(t_0, \mathbf{x}_0) \neq 0$. Then there exists an open interval J centered at t_0, an open disk V centered at $\mathbf{x}_0$, and a unique C^1 function $\mathbf{x} : J \times V \to \mathbb{R}^2$ that satisfy the following properties:*

(i) $u(t, \mathbf{x}(t, \mathbf{y})) = u(t_0, \mathbf{y})$ *and* $\mathbf{x}(t_0, \mathbf{y}) = \mathbf{y}$ *for all* $(t, \mathbf{y}) \in J \times V$.

(ii) The vectors $(\partial \mathbf{x}/\partial t)(t,\mathbf{y})$ and $Du(t,\mathbf{x}(t,\mathbf{y}))$ are collinear.

In addition, the function $\mathbf{x}$ satisfies the following differential equation:

$$\frac{\partial \mathbf{x}}{\partial t}(t,\mathbf{y}) = -\Big(\frac{Du}{|Du|^2}\frac{\partial u}{\partial t}\Big)(t,\mathbf{x}(t,\mathbf{y})). \tag{18.4}$$

The trajectory $t \mapsto \mathbf{x}(t,\mathbf{y})$ is called the normal flow starting from $(t_0,\mathbf{y})$.

Proof. Differentiating the relation $u(t,\mathbf{x}(t)) = 0$ with respect to t yields $\frac{\partial u}{\partial t} + Du.\frac{\partial \mathbf{x}}{\partial t} = 0$. By multiplying this equation by the vector Du we see that $\frac{\partial \mathbf{x}}{\partial t}$ is collinear to Du if and only if (18.4) holds. Now, this relation defines $\mathbf{x}(t)$ as the solution of an ordinary differential equation, with initial condition $\mathbf{x}(t_0) = \mathbf{y}$. Since u is C^2, the second member of (18.4) appears to be a Lipschitz function of $(t,\mathbf{x})$ provided $Du(t,\mathbf{x}) \neq 0$, which is ensured for $(t,\mathbf{x})$ close enough to $(t_0,\mathbf{x}_0)$. Thus, by Cauchy-Lipschitz Theorem, there exists an open interval J such that the O.D.E. (18.4) has a unique solution $\mathbf{x}(t,\mathbf{y})$ for all $\mathbf{y}$ in a neighborhood of $\mathbf{x}_0$ and $t \in J$. □

Proposition 18.7. *Assume that the function $(t,\mathbf{x}) \mapsto u(t,\mathbf{x})$ is C^2 in a neighborhood of $(t_0,\mathbf{x}_0)$ and that $Du(t_0,\mathbf{x}_0) \neq 0$. Then u satisfies the curvature motion equation*

$$\frac{\partial u}{\partial t}(t,\mathbf{x}) = \mathrm{curv}(u)(t,\mathbf{x})|Du|(t,\mathbf{x}) \tag{18.5}$$

in a neighborhood of $(t_0,\mathbf{x}_0)$ if and only if the normal flow $\mathbf{x}(t,\mathbf{y})$ of u in this neighborhood satisfies the intrinsic heat equation

$$\frac{\partial \mathbf{x}}{\partial t}(t,\mathbf{y}) = \boldsymbol{\kappa}(\mathbf{x}(t,\mathbf{y})), \tag{18.6}$$

where $\boldsymbol{\kappa}(\mathbf{x}(t,\mathbf{y}))$ denotes the curvature vector of the level line of $u(t)$ passing by $\mathbf{x}(t,\mathbf{y})$.

Proof. Assume first that $\mathbf{x}(t,\mathbf{y})$ satisfies (18.6). Applying (17.8) for all t in a neighborhood of t_0 to each image $u(t) : \mathbf{x} \to u(t,\mathbf{x})$ yields

$$\boldsymbol{\kappa}(\mathbf{x}(t,\mathbf{y})) = -\mathrm{curv}(u)\frac{Du}{|Du|}(t,\mathbf{x}(t,\mathbf{y})).$$

Substituting (18.6) in this last relation we obtain

$$\frac{\partial \mathbf{x}}{\partial t}(t,\mathbf{y}) = -\mathrm{curv}(u)\frac{Du}{|Du|}(t,\mathbf{x}(t,\mathbf{y}))$$

and by the normal flow equation (18.4),

$$(\frac{\partial u}{\partial t}\frac{Du}{|Du|^2})(t,\mathbf{x}(t,\mathbf{y})) = \mathrm{curv}(u)\frac{Du}{|Du|}(t,\mathbf{x}(t,\mathbf{y})).$$

Multiplying this equation by $Du(t,\mathbf{x}(t,\mathbf{y}))$ yields the curvature motion equation (18.5).

The converse statement follows exactly the same lines backwards. □

Exercise 18.1. Write the proof of the converse statement of Proposition 18.7. ■

The preceding proof is immediately adaptable to all curvature equations :

Proposition 18.8. *Assume that the function $(t,\mathbf{x}) \mapsto u(t,\mathbf{x})$ is C^2 in a neighborhood of $(t_0,\mathbf{x}_0)$ and that $Du(t_0,\mathbf{x}_0) \neq 0$. Let $g : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}$ be continuous and nondecreasing with respect to κ and such that $g(-\kappa,t) = -g(\kappa,t)$. Then u satisfies the curvature motion equation*

$$\frac{\partial u}{\partial t}(t,\mathbf{x}) = g(\mathrm{curv}(u)(t,\mathbf{x}),t)|Du|(t,\mathbf{x}) \tag{18.7}$$

in a neighborhood of $(t_0,\mathbf{x}_0)$ if and only if the normal flow $t \mapsto \mathbf{x}(t,\cdot)$ satisfies the curvature equation

$$\frac{\partial \mathbf{x}}{\partial t}(t,\mathbf{y}) = g(|\boldsymbol{\kappa}(\mathbf{x}(t,\mathbf{y}))|)\frac{\boldsymbol{\kappa}(\mathbf{x}(t,\mathbf{y}))}{|\boldsymbol{\kappa}(\mathbf{x}(t,\mathbf{y}))|}. \tag{18.8}$$

18.3.2 Introduction to the affine curve and function equations

There are two curvature equations that are affine invariant and are therefore particularly well suited for use in shape recognition. In their definition, for $x \in \mathbb{R}$, $x^{1/3}$ stands for $\mathrm{sign}(x)|x|^{1/3}$.

Definition 18.9. *The image evolution equation*

$$\frac{\partial u}{\partial t}(t,\mathbf{x}) = (\mathrm{curv}(u)(t,\mathbf{x}))^{1/3}|Du(t,\mathbf{x})| \tag{18.9}$$

is called affine morphological scale space (AMSS). The curve evolution equation

$$\frac{\partial \mathbf{x}}{\partial t}(t,s) = |\boldsymbol{\kappa}(\mathbf{x}(t,s))|^{1/3}\mathbf{n}(t,s) \quad \Big(= \frac{\boldsymbol{\kappa}(\mathbf{x}(t,s))}{|\boldsymbol{\kappa}(\mathbf{x}(t,s))|^{2/3}}\Big) \tag{18.10}$$

is called affine scale space (ASS).

It is clear that AMSS and ASS are equivalent in the sense of Proposition 18.8. As one would expect from the names of these equations, they both have some sort of affine invariance. This is the subject of the next definition, Exercises 18.3 and 18.4 and the next section.

Definition 18.10. *We say that a curvature equation (E) (image evolution equation) is affine invariant, if for every linear map A with positive determinant, there is a positive constant $c = c(A)$ such that $(t,\mathbf{x}) \mapsto u(t,\mathbf{x})$ is a solution of (E) if and only if $(ct, A\mathbf{x}) \mapsto u(ct, A\mathbf{x})$ is a solution of (E).*

18.3.3 The affine scale space as an intrinsic heat equation

Suppose that for each scale t, $\sigma \mapsto \mathbf{x}(t,\sigma)$ is a Jordan arc (or curve) parameterized by σ, which is not in general an arc length. As in Chapter 17, we will denote the curvature of $\mathbf{x}$ by $\boldsymbol{\kappa}$. We wish to demonstrate a formal equivalence between the affine scale space,

$$\frac{\partial \mathbf{x}}{\partial t} = |\boldsymbol{\kappa}|^{1/3}\mathbf{n}(\mathbf{x}), \tag{18.11}$$

and an "intrinsic heat equation"

$$\frac{\partial \mathbf{x}}{\partial t} = \frac{\partial^2 \mathbf{x}}{\partial \sigma^2}, \tag{18.12}$$

where σ is a special parameterization called *affine length*. We define an *affine length parameter* of a Jordan curve (or arc) to be any parameterization $\sigma \mapsto \mathbf{x}(\sigma)$ such that

$$[\mathbf{x}_\sigma, \mathbf{x}_{\sigma\sigma}] = 1, \tag{18.13}$$

where $[\mathbf{x}, \mathbf{y}] = \mathbf{x}^\perp \cdot \mathbf{y}$. If s is an arc-length parameterization, then we have (Definition 17.3)

$$\boldsymbol{\tau} = \mathbf{x}_s \quad \mathbf{n} = |\boldsymbol{\kappa}|^{-1}\mathbf{x}_{ss} \quad \Big(= \frac{\boldsymbol{\kappa}(\mathbf{x})}{|\boldsymbol{\kappa}(\mathbf{x})|}\Big). \tag{18.14}$$

We also have

$$\mathbf{x}_\sigma = \mathbf{x}_s \frac{\partial s}{\partial \sigma} \quad \text{and} \quad \mathbf{x}_{\sigma\sigma} = \mathbf{x}_{ss}\Big(\frac{\partial s}{\partial \sigma}\Big)^2 + \mathbf{x}_s \frac{\partial^2 s}{\partial \sigma^2}. \tag{18.15}$$

Thus,

$$[\mathbf{x}_\sigma, \mathbf{x}_{\sigma\sigma}] = [\mathbf{x}_s, \mathbf{x}_{ss}]\Big(\frac{\partial s}{\partial \sigma}\Big)^3,$$

and if (18.13) holds, then

$$[\mathbf{x}_s, \mathbf{x}_{ss}]\Big(\frac{\partial s}{\partial \sigma}\Big)^3 = 1.$$

Since by (18.14) $[\mathbf{x}_s, \mathbf{x}_{ss}] = \operatorname{sign}([\mathbf{x}_s, \mathbf{x}_{ss}])|\boldsymbol{\kappa}|$, we conclude that

$$\frac{\partial s}{\partial \sigma} = (\operatorname{sign}([\mathbf{x}_s, \mathbf{x}_{ss}])|\boldsymbol{\kappa}|)^{-1/3}. \tag{18.16}$$

Substituting this result in the expression for $\mathbf{x}_{\sigma\sigma}$ shown in (18.15) and writing $\mathbf{x}_s = \boldsymbol{\tau}$, we see that

$$\mathbf{x}_{\sigma\sigma} = |\boldsymbol{\kappa}|^{1/3}\mathbf{n} + \Big(\frac{\partial^2 s}{\partial \sigma^2}\Big)\boldsymbol{\tau}.$$

This tells us that equation (18.12) is equivalent to the following equation:

$$\frac{\partial \mathbf{x}}{\partial t} = |\boldsymbol{\kappa}|^{1/3}\mathbf{n} + \Big(\frac{\partial^2 s}{\partial \sigma^2}\Big)\boldsymbol{\tau}. \tag{18.17}$$

Now it turns out that the graphs of the functions $\mathbf{x}$ that you get from one time to another do not depend on the term involving $\boldsymbol{\tau}$; you could drop this term and get the same graphs. More precisely, Epstein and Gage [72] have shown that the tangential component of an equation like (18.17) does not matter as far as the geometric evolution of the curve is concerned. In fact, the tangential term just moves points along the curve itself, and the total curve evolution is determined by the normal term. As a consequence, equation (18.11) is equivalent to equation (18.12) in any neighborhood that avoids an inflection point, that is, in any neighborhood where $\mathbf{n}(\mathbf{x}) \neq 0$. At an inflection point, $\boldsymbol{\kappa} = 0$, and the two equations give the same result.

18.4 Curvature motion in N dimensions

We consider an evolution $(t, \mathbf{x}) \mapsto u(t, \mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^N$ and $u(0, \cdot) = u_0$ is an initial N–dimensional image. Let $\kappa_i(u)(t, \mathbf{x})$, $i = 1, \ldots, N-1$, denote the i^{th} principal curvature at the point $(t, \mathbf{x})$. By definition 17.20 the mean curvature is $\operatorname{curv}(u) = \sum_{i=1}^{N-1} \kappa_i$. We will now define three curvature motion flow equations in N dimensions.

Mean curvature motion. This equation is a direct translation of equation (18.5) in N dimensions:

$$\frac{\partial u}{\partial t} = |Du| \operatorname{curv}(u).$$

This says that the motion of a level hypersurface of u in the normal direction is proportional to its mean curvature.

Gaussian curvature motion for convex functions. We say that a function is convex if all of its principal curvatures have the same sign. An example of such a function is the signed distance function to a regular convex shape. The equation is

$$\frac{\partial u}{\partial t} = |Du| \prod_{i=1}^{N-1} \kappa_i.$$

The motion of a level hypersurface is proportional to the product of its principal curvatures, which is the Gaussian curvature. As we will see in Chapter **??**, this must be modified before it can be applied to a nonconvex function.

Affine-invariant curvature motion. The equation is

$$\frac{\partial u}{\partial t} = |Du| \Big| \prod_{i=1}^{N-1} \kappa_i \Big|^{1/(N+1)} H\Big(\sum_{i=1}^{N-1} \operatorname{sign}(\kappa_i) \Big),$$

where $H(N-1) = 1$, $H(-N+1) = -1$, and $H(n) = 0$ otherwise. The motion is similar to Gaussian curvature motion, but the affine invariance requires that the Gaussian curvature be raised to the power $1/(N+1)$. There is no motion at a point where the principal curvatures have mixed signs. This means that only concave or convex parts of level surfaces get move by such an equation.

18.5 Exercises

Exercise 18.2. Check that all of the curvature equations (18.2) are contrast invariant. That is, assuming that h is a real-valued C^2 increasing function defined on $\mathbb{R}$ and u is C^2, show that the function v defined by $v(t, \mathbf{x}) = h(u(t, \mathbf{x}))$ satisfies one of these equations if and only if u satisfies the same equation. ■

Exercise 18.3. Assume that $(t, \mathbf{x}) \mapsto u(t, \mathbf{x})$ is a C^2 function and that A is a 2×2 matrix with positive determinant, which we denote by $|A|$. Define the function v by $v(t, \mathbf{x}) = u(ct, A\mathbf{x})$, where $c = |A|^{-2/3}$.

(i) Prove that for each point $\mathbf{x}$ such that $Du(\mathbf{x}) \neq 0$ one has the relation

$$\operatorname{curv}(v)(\mathbf{x})|Dv(\mathbf{x})|^3 = |A|^2 \operatorname{curv}(u)(A\mathbf{x})|Du(A\mathbf{x})|^3.$$

(ii) Use (i) to deduce that the AMSS equation (18.9) is affine invariant, that is, $(t,\mathbf{x}) \mapsto u(t,\mathbf{x})$ is a solution of AMSS if and only $(t,\mathbf{x}) \mapsto v(t,\mathbf{x})$ does. ■

Exercise 18.4. This exercise is to show that the affine scale space (equation (18.10)) is affine invariant. It relies directly on results from Exercise 17.9. Let $\sigma \mapsto \mathbf{c}(\sigma)$ be a C^2 curve, and assume that $|\mathbf{c}'(\sigma)| > 0$. Then we know from Exercise 17.9 that

$$\boldsymbol{\kappa}(\mathbf{c})(\sigma) = \frac{1}{|\mathbf{c}'(\sigma)|^2}\left[\mathbf{c}''(\sigma) - \left(\mathbf{c}''(\sigma)\cdot\frac{\mathbf{c}'(\sigma)}{|\mathbf{c}'(\sigma)|}\right)\frac{\mathbf{c}'(\sigma)}{|\mathbf{c}'(\sigma)|}\right]. \tag{18.18}$$

Now assume that we have a family of C^2 Jordan arcs $(t,\sigma) \mapsto \mathbf{c}(t,\sigma)$. By projecting both sides of the intrinsic heat equation onto the unit vector $\mathbf{c}'^{\perp}/|\mathbf{c}'|$ and by using (18.18), we have the following equation:

$$\frac{\partial \mathbf{c}}{\partial t}\cdot\frac{\mathbf{c}'^{\perp}}{|\mathbf{c}'|} = \frac{\mathbf{c}''\cdot\mathbf{c}'^{\perp}}{|\mathbf{c}'|^3} \tag{18.19}$$

We say that $\mathbf{c}$ satisfies a parametric curvature equation if it satisfies equation (18.19). In the same spirit, we say that $\mathbf{c}$ satisfies a parametric affine equation if for some constant $\gamma > 0$

$$\frac{\partial \mathbf{c}}{\partial t}\cdot\mathbf{c}'^{\perp} = \gamma(\mathbf{c}''\cdot\mathbf{c}'^{\perp})^{1/3}. \tag{18.20}$$

(i) Suppose that $\sigma = s$, an arc-length parameterization of $\mathbf{c}$. Show that equation (18.19) can be written as

$$\frac{\partial \mathbf{c}}{\partial t} = \boldsymbol{\kappa}(\mathbf{c}) + \lambda\tau,$$

where λ is a real-valued function and τ is the unit tangent vector $\partial\mathbf{c}/\partial s$. (See the remark following equation (18.17).)

(ii) Let A be a 2×2 matrix with positive determinant, and define the curve $\mathbf{y}$ by $\mathbf{y}(t,\sigma) = A\mathbf{c}(t,\sigma)$. We wish to show that if $\mathbf{c}$ satisfies a parametric affine motion, then so does $\mathbf{y}$. As a first step, show that $A\mathbf{x}\cdot(A\mathbf{y})^{\perp} = |A|\mathbf{x}\cdot\mathbf{y}$ and hence that $A(\mathbf{x}^{\perp})\cdot(A\mathbf{x})^{\perp} = |A||\mathbf{x}|^2$ for any $\mathbf{x},\mathbf{y}\in\mathbb{R}^2$.

(iii) Show that if $\mathbf{c}$ satisfies equation (18.20), then $\mathbf{y}$ satisfies

$$\frac{\partial \mathbf{y}}{\partial t}\cdot\mathbf{y}'^{\perp} = \gamma|A|^{2/3}(\mathbf{y}''\cdot\mathbf{y}'^{\perp})^{1/3}. \blacksquare$$

18.6 Comments and references

Our definition of shape. The Italian mathematician Renato Caccioppoli proposed a theory of sets whose boundaries have finite length (finite Hausdorff measure). From his theory, it can be deduced that the boundary of a Caccioppoli set is composed of a countable number of Jordan curves, up to a set with zero length. This decomposition can even be made unambiguous. In other words, the set of Jordan curves associated with a given Caccioppoli set is unique and gives enough information to reconstruct the set [10]. This result justifies our focus on Jordan curves as the representatives of shapes.

The role of curvature in shape analysis. After Attneave's founding paper [20], let us mention the thesis by G. J. Agin [4] as being one of the first references dealing with the use of curvature for the representation and recognition of objects in computer vision. The now-classic paper by Asada and Brady [17] entitled "The curvature primal sketch" introduced the notion of computing a "multiscale curvature" as a tool for object recognition. (The title is an allusion to David Marr's famous "raw primal sketch," which is a set of geometric primitives extracted from and representing an image.) The Asada–Brady paper led to a long series of increasingly sophisticated attempts to represent shape from curvature [70, 71] and to compute curvature correctly [186]. The shape recognition programme we sketched in the beginning of this chapter was anticipated in a visionary paper by Attneave [20] and has been very recently fully developed in the works of José Luis Lisani, Pablo Musé, Frédéric Sur, Yann Gousseau and Frédéric Cao [188], [190], [39], [40].

Curve shortening. The mathematical study of the intrinsic heat equation (or curvature motion in two dimensions) was done is a series of brilliant papers in differential geometry between 1983 and 1987. We repeat a few of the titles, which indicate the progress: There was Gage [88] and Gage [89]: "Curve shortening makes convex curves circular." Then there was Gage and Hamilton [90]: "The heat equation shrinking convex plane curves." In this paper the authors showed that a plane convex curve became asymptotically close to a shrinking circle. In 1987 there was the paper by Epstein and Gage [72], and, in the same year, Grayson removed the convexity condition and finished the job [98]: "The heat equation shrinks embedded plane curves to round points." As the reviewer, U. Pinkall, wrote, "This paper contains the final solution of the long-standing *curve-shortening problem* for plane curves."

The first papers that brought curve shortening (and some variations) to image analysis were by Kimia, Tannenbaum, and Zucker [130] and by Mackworth and Mokhtarian [154]. Curve shortening was introduced as a way to do a multiscale analysis of curves, which were considered as shapes extracted from an image. In the latter paper, curve shortening was proposed as an efficient numerical tool for multiscale shape analysis.

Affine-invariant curve shortening. Affine-invariant geometry seems to have been founded by W. Blaschke. His three-volume work "Vorlesungen über Differentialgeometrie" (1921–1929) contains definitions of affine length and affine curvature. Curves with constant affine curvature are discussed in [155]. The term "affine shortening" and the corresponding curve evolution equation were introduced by Sapiro and Tannenbaum in [221]. Several mathematical properties were developed by the same authors in [222] and [223]. Angenent, Sapiro, and Tannenbaum gave the first existence and uniqueness proof of affine shortening in [14] and prove a theorem comparable to Grayson's theorem : they prove that a shape eventually becomes convex and thereafter evolves towards an ellipse before collapsing.

Mean curvature motion. In his famous paper entitled "Shapes of worn stones," Firey proposed a model for the natural erosion of stones on a beach [81]. He suggested that the rate of erosion of the surface of a stone was proportional to the Gaussian curvature of the surface, so that areas with high Gaussian curvature eroded faster than areas with lower curvature, and he conjectured that the final shape was a sphere. The first attempt at a mathematical definition of the mean curvature motion is found in Brakke [31]. Later in the book, we will discuss the Sethian's clever numerical implementation of the same equation

[233]. Almgren, Taylor, Wang proposed a more general formulation of mean curvature motion that is applicable to crystal growth and, in general, to the evolution of anisotropic solids [5].

Chapter 19

Finite Difference Schemes for Curvature Motions

We shall consider the classical discrete representation of an image u on a grid $u_{i,j} = u(i,j)$, with $1 \leq i \leq N$, $1 \leq j \leq N$. The image is the union of the squares centered at the points (i,j).

19.1 Case of Mean curvature motion.

We start with the "Mean curvature motion" equation (M.C.M.) given by

$$\frac{\partial u}{\partial t} = |Du|curv(u) = \frac{u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy}}{u_x^2 + u_y^2}$$

In order to discretize this equation by finite differences we shall introduce an explicit scheme which uses a fixed 3×3 stencil to discretize the differential operators. We denote by $\Delta x = \Delta y$ the pixel width. From the PDE viewpoint Δx is considered as an infinitesimal length with respect to the image scale. Thus we shall write formulas containing $o(\Delta x)$. Numerically Δx is equal to 1, and the image scale ranges from 512 to 4096 and more. By the order 1 Taylor formula one can give the following discrete versions of the first derivatives u_x and u_y at a point (i,j) of the grid:

$$(u_x)_{i,j} = \frac{2(u_{i+1,j} - u_{i-1,j}) + u_{i+1,j+1} - u_{i-1,j+1} + u_{i+1,j-1} - u_{i-1,j-1}}{8\Delta x} + O(\Delta x^2);$$

$$(u_y)_{i,j} = \frac{2(u_{i,j+1} - u_{i,j-1}) + u_{i+1,j+1} - u_{i+1,j-1} + u_{i-1,j+1} - u_{i-1,j-1}}{8\Delta x} + O(\Delta x^2);$$

$$|Du_{i,j}| = ((u_x)_{i,j}^2 + (u_y)_{i,j}^2)^{\frac{1}{2}}.$$

Definition 19.1. *A discrete scheme approximating a differential operator is said to be consistent if, when the grid mesh Δx tends to zero, the discrete scheme tends to the differential operator.*

λ4	λ2	λ3
λ1	–4 λ0	λ1
λ3	λ2	λ4

Figure 19.1: A 3×3 stencil

Clearly the above discrete versions of the partial derivatives and of the gradient of u are consistent. When $|Du| \neq 0$, we can denote by ξ the direction orthogonal to the gradient of u. It is easily deduced from Definition 17.14 that

$$|Du|curv(u) = u_{\xi\xi}.$$

Exercise 19.1. Show this formula. ■

Defining θ as the angle between the x direction and the gradient, we have

$$\xi = (-\sin\theta, \cos\theta) = (\frac{-u_y}{\sqrt{u_x^2 + u_y^2}}, \frac{u_x}{\sqrt{u_x^2 + u_y^2}}), \text{ and}$$

$$u_{\xi\xi} = \sin^2(\theta)u_{xx} - 2\sin(\theta)\cos(\theta)u_{xy} + \cos^2(\theta)u_{yy}. \tag{19.1}$$

We would like to write $u_{\xi\xi}$ as a linear combination of the values of u on the fixed 3×3 stencil. Of course, the coefficients of the linear combination will depend on ξ. Since the direction of ξ is defined modulo π, we must assume by symmetry that the coefficients of points symmetrical with respect to the central point of the stencil are equal (see Figure 19.1.)

In order to ensure consistency with the differential operator $u_{\xi\xi}$, we must find $\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4$, such that

$$(u_{\xi\xi})_{i,j} = \frac{1}{\Delta x^2}(-4\lambda_0 u_{i,j} + \lambda_1(u_{i+1,j} + u_{i-1,j}) + \lambda_2(u_{i,j+1} + u_{i,j-1})$$

$$+\lambda_3(u_{i-1,j-1} + u_{i+1,j+1}) + \lambda_4(u_{i-1,j+1} + u_{i+1,j-1})) + \varepsilon(\Delta x). \tag{19.2}$$

We write

$$u_{i+1,j} = u_{i,j} + \Delta x(u_x)_{i,j} + \frac{\Delta x^2}{2}(u_{xx})_{i,j} + o((\Delta x)^3),$$

and the corresponding relations for the other points of the stencil. By substituting these relations into (19.2) and by using (19.1) one obtains four links between the five coefficients, namely

$$\begin{cases} \lambda_1(\theta) = 2\lambda_0(\theta) - \sin^2\theta \\ \lambda_2(\theta) = 2\lambda_0(\theta) - \cos^2\theta \\ \lambda_3(\theta) = -\lambda_0(\theta) + 0.5(\sin\theta\cos\theta + 1) \\ \lambda_4(\theta) = -\lambda_0(\theta) + 0.5(-\sin\theta\cos\theta + 1) \end{cases} \tag{19.3}$$

Exercise 19.2. Prove these four relations. ■

Thus, one degree of freedom is left for our coefficients : we can for example choose $\lambda_0(\theta)$ as we wish. This choice will be driven by stability and geometric invariance requirements. Denoting by $u^n_{i,j}$ an approximation of $u(i\Delta x, j\Delta x, n\Delta t)$ we can write our explicit scheme as

$$u^{n+1}_{i,j} = u^n_{i,j} + \Delta t (u^n_{\xi\xi})_{i,j} \tag{19.4}$$

Notice that this scheme can be rewritten as $u^{n+1}_{i,j} = \sum_{k,l=-1}^{1} \alpha_{k,l} u^n_{i+k,j+l}$ where the $\alpha_{k,l}$ satisfy $\sum_{k,l=-1}^{1} \alpha_{k,l} = 1$. The following obvious lemma shows a general condition to have L^∞ *stability* in this kind of scheme.

Lemma 19.2. *Let a finite difference scheme given by*

$$T(u)_{i,j} = \sum_{k,l=-1}^{1} \alpha_{k,l} u_{i+k,j+l}$$

*where $\alpha_{k,l}$ satisfy $\sum_{k,l=-1}^{1} \alpha_{k,l} = 1$. We say that the scheme is L^∞-*stable *if for all i, j,*

$$\min_{i,j} u(i,j) \leq T(u)_{i,j} \leq \max_{i,j} u(i,j).$$

Then the scheme is L^∞ stable if and only if $\alpha_{k,l} \geq 0$ for any k, l.

Proof. If $\alpha_{k,l} \geq 0$ for any k, l, set $min = inf_{i,j}\{u_{i,j}\}$, $max = sup_{i,j}\{u_{i,j}\}$ and take a point (i, j). Then the L^∞ stability follows from the inequality:

$$min = \sum_{k,l=-1}^{1} \alpha_{k,l} min \leq \sum_{k,l=-1}^{1} \alpha_{k,l} u_{i+k,j+l} = (Tu)_{i,j} \leq \sum_{k,l=-1}^{1} \alpha_{k,l} max = max$$

On the other hand, if there exists $\alpha_{k_0,l_0} < 0$ then choosing u and (i, j) such that $u_{i+k_0,j+l_0} = min$ and $u_{i+k,j+l} = max$ for any other k, l, we obtain

$$(Tu)_{i,j} = \sum_{k\neq k_0, l\neq l_0}^{1} \alpha_{k,l} max + \alpha_{k_0,l_0} min = max + \alpha_{k_0,l_0}(min - max) > max,$$

which means that the L^∞ stability is violated.

□

Following this lemma, in order to guarantee the L^∞ stability in the scheme (19.4) we should look for λ_0 such that $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$ and $(1 - \frac{4\lambda_0}{\Delta x^2}) \geq 0$. Unfortunately the links between these coefficients make it impossible to obtain these relations, except for the particular values of $\theta = (0, \frac{\pi}{4}, \frac{\pi}{2}, ...)$. Indeed, for θ in $[0, \frac{\pi}{4}]$,

$$\lambda_1 \geq \lambda_2 \qquad \text{and} \qquad \lambda_3 \geq \lambda_4$$

But

$$\lambda_2(\theta) \geq 0 \Rightarrow \lambda_0(\theta) \geq \frac{\cos^2(\theta)}{2}$$

$$\lambda_4(\theta) \geq 0 \Rightarrow \lambda_0(\theta) \leq \frac{1 - \sin(\theta)\cos(\theta)}{2}$$

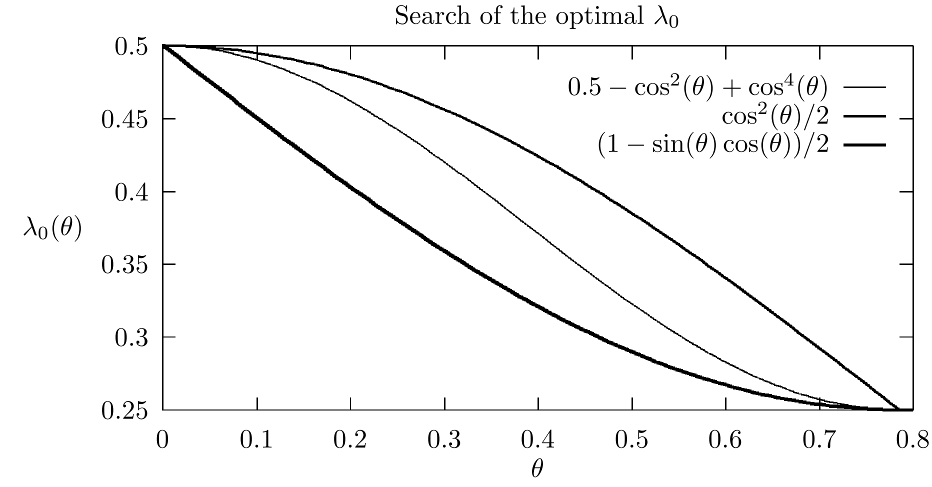

Figure 19.2: The middle curve represents the choice of the function λ_0 of Formula 19.6. The upper function represents the smallest possibility for $\lambda_0(\theta)$ securing $\lambda_2 \geq 0$ for all angles and the lower one represents the largest values of $\lambda_0(\theta)$ securing $\lambda_4(\theta) \geq 0$. Thus, it is not possible to satisfy simultaneously both conditions. The intermediate curve is the simplest trigonometric function which lies between these two bounds.

We cannot find $\lambda_0(\theta)$ satisfying both inequalities, since

$$\frac{\cos^2(\theta)}{2} \geq \frac{1 - \sin(\theta)\cos(\theta)}{2}$$

If we chose $\lambda_0(\theta) \geq \frac{\cos^2(\theta)}{2}$, $\lambda_4(\theta)$ would be significantly below zero. If we took $\lambda_0(\theta) \leq \frac{1-\sin(\theta)\cos(\theta)}{2}$, $\lambda_2(\theta)$ would be significantly below zero. Thus we shall choose λ_0 somewhere between both functions, so that λ_2 and λ_4 become only slightly negative. (see Figure 19.2.)

In addition, we can try to impose on λ_0 the following geometrical requirements

(i). Invariance by rotation of angle $\frac{\pi}{2}$

$$\lambda_0(\theta + \frac{\pi}{2}) = \lambda_0(\theta)$$

(ii). Purely one-dimensional diffusion in the case $\theta = 0, \frac{\pi}{2}, ...$

$$\lambda_0(0) = 0.5$$

This condition implies that $\lambda_2(0) = \lambda_3(0) = \lambda_4(0) = 0$

(iii). Pure one-dimensional diffusion in the case $\theta = \frac{\pi}{4}, \frac{3\pi}{4}, ...$

$$\lambda_0(\frac{\pi}{4}) = 0.25$$

This condition implies that $\lambda_1(\frac{\pi}{4}) = \lambda_2(\frac{\pi}{4}) = \lambda_4(\frac{\pi}{4}) = 0$

(iv). Symmetry with respect to the axes i+j and i-j,

$$\lambda_0(\frac{\pi}{2} - \theta) = \lambda_0(\theta)$$

We remark that by the above conditions it is enough to define the function $\lambda_0(\theta)$ in the interval $[0, \frac{\pi}{4}]$ because it can be extended by periodicity elsewhere.

Two choices for the function $\lambda_0(\theta)$ using as basis the trigonometric polynomials were tested. The first one corresponds to an average of the boundary functions:

$$\lambda_0(\theta) = \frac{\cos^2(\theta) + 1 - \sin(\theta) cos(\theta)}{4} \tag{19.5}$$

As we shall see this choice is well-adapted to the "affine curvature motion" equation. However, if we extend this function by periodicity, the extended function is not smooth at $\frac{\pi}{4}$. If we seek for a smooth function for $\lambda_0(\theta)$, we must impose $\lambda_0'(0) = \lambda_0'(\frac{\pi}{4}) = 0$. The trigonometric polynomial with least degree satisfying the above conditions and lying between both boundary functions is

$$\lambda_0(\theta)) = 0.5 - \cos^2(\theta)\sin^2(\theta) \tag{19.6}$$

The formulas of the other λ_i's are deduced using (19.3). For instance with the above choice of $\lambda_0(\theta)$ we have

$$\begin{cases} \lambda_1(\theta) = \cos^2(\theta)(\cos^2(\theta) - \sin^2(\theta)); \\ \lambda_2(\theta) = \sin^2(\theta)(\sin^2(\theta) - \cos^2(\theta)); \\ \lambda_3(\theta) = \cos^2(\theta)\sin^2(\theta) + 0.5\sin(\theta)\cos(\theta); \\ \lambda_4(\theta) = \cos^2(\theta)\sin^2(\theta) - 0.5\sin(\theta)\cos(\theta). \end{cases}$$

When $|Du| = 0$, the direction of the gradient is unknown. Therefore the diffusion term $u_{\xi\xi}$ is not defined. We chose to replace this term by half the Laplacian. (The Laplacian is equal to the sum of the two second derivatives in orthogonal directions, whereas the diffusion term $u_{\xi\xi}$ is the second derivative in just one). However, other possibilities will be considered in Section 19.6. Summarizing, a consistent, almost L^∞ stable finite difference scheme for the mean curvature motion is (iterations start with u^0 as initial function)

1. If $|Du| \geq T_g$

$$u^{n+1} = u^n + \frac{\Delta t}{\Delta x^2}(-4\lambda_0 u_{i,j} + \lambda_1(u_{i+1,j} + u_{i-1,j}) + \lambda_2(u_{i,j+1} + u_{i,j-1})$$
$$+\lambda_3(u_{i-1,j-1} + u_{i+1,j+1}) + \lambda_4(u_{i-1,j+1} + u_{i+1,j-1})).$$

2. Otherwise,

$$u^{n+1} = u^n + \frac{1}{2}\frac{\Delta t}{\Delta x^2}(-4\lambda_0 u_{i,j} + u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})$$

Two parameters have to be fixed in the previous algorithm:

• **The iteration step scale** $s := \frac{\Delta t}{\Delta x^2}$ has to be chosen as large as possible in order to reduce the number of iterations. However, $1/2$ is a natural upper bound for s. Indeed, consider the discrete image defined by $u^0_{i,j} = 0$ for all i, j, except for $i = j = 0$ where $u^0_{0,0} = 1$. Then the second formula yields $u^1_{0,0} = 1 - 2 * s$. If we want L^∞ stability to be ensured we must have $u^1(0,0) \geq 0$, which yields $s \leq 1/2$. Imposing this condition

$$\frac{\Delta t}{\Delta x^2} \leq \frac{1}{2} \tag{19.7}$$

it is an experimental observation that there is a (small with respect to 255) $\epsilon > 0$ such that for any $n \in I\!N$ and (i,j),

$$-\epsilon + \inf_{i,j}\{u^0_{i,j}\} \leq u^n_{i,j} \leq \sup_{i,j}\{u^0_{i,j}\} + \epsilon.$$

• **The threshold on the spatial gradient norm** : T_g has been fixed experimentally to 6 for 0 to 255 images.

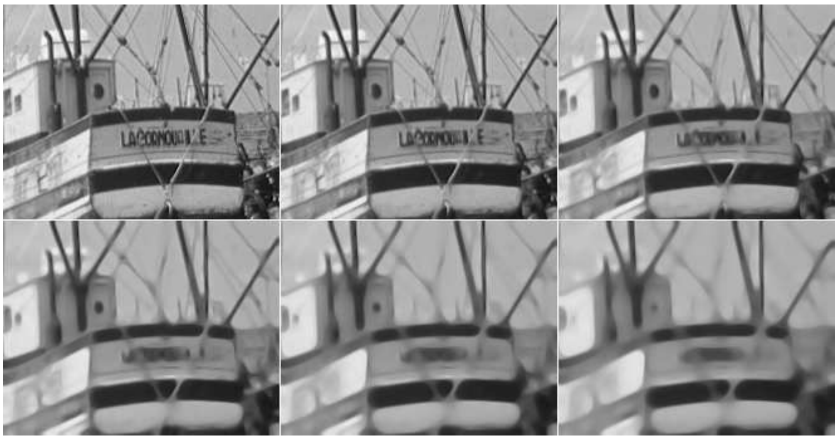

Figure 19.3: Curvature motion finite difference scheme and scale calibration. Image filtered by curvature motion at scales 1, 2, 3, 4, 5. In order to give a sound numerical meaning to the scale, a calibration of the numerical scales (number of iterations) is made in such a way that a disk with radius t shrinks to a point at scale t.

19.2 FDS for AMSS

We will use the ideas developed in the above section. We rewrite the AMSS equation as

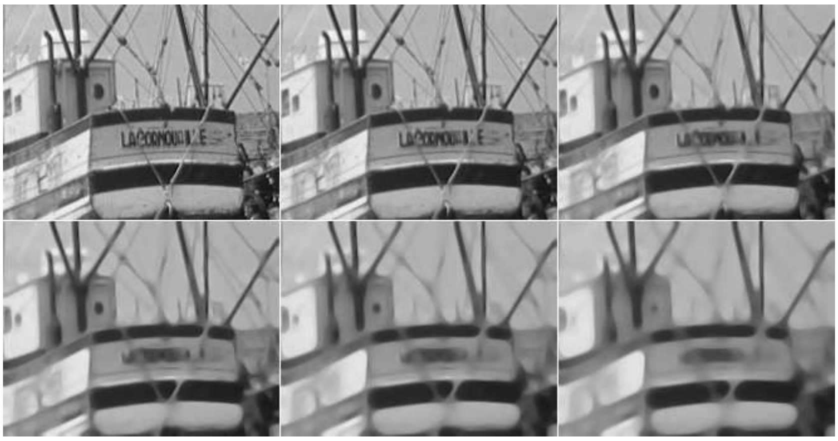

Figure 19.4: Curvature motion finite difference scheme applied on each level set separately, at scales 1, 2, 3, 4, 5. The processed image is then reconstructed by the threshold superposition principle. In contrast with the same scheme directly applied on the image, this scheme yields a fully contrast invariant smoothing. However, a comparison with Figure 19.3 shows that the resulting images are very close to each other. This shows that the contrast invariance is almost achieved when applying the finite difference scheme directly on a good quality image. As we shall see in Figure 19.6, if the initial image is noisy, the difference between both methods can be huge.

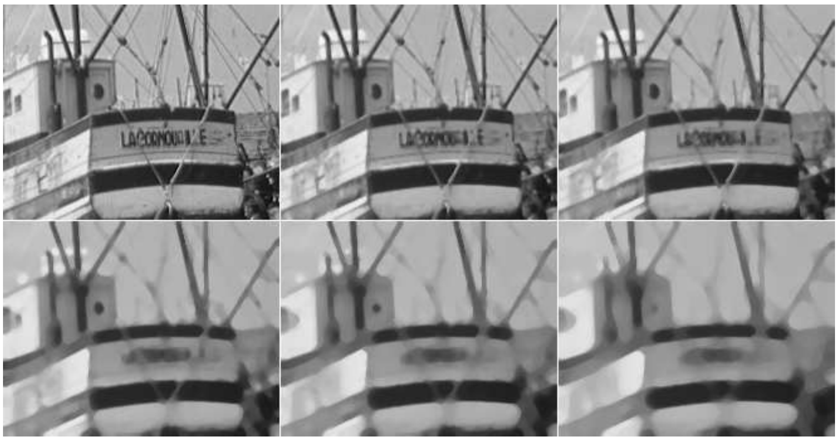

Figure 19.5: Iterated median filter with normalized scales 1, 2, 3, 4, 5. The scale normalization permits to compare very different schemes on the same images. Compare with Figure 19.4. The striking similarity of the results anticipates Theorem **??**, according to which the application of the median filter is equivalent to a mean curvature motion.

$$\frac{\partial u}{\partial t} = (|Du|^3 curv(u))^{\frac{1}{3}} = (u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy})^{\frac{1}{3}} \qquad (19.8)$$

We remark that $|Du|^3 curv(u) = |Du|^2 u_{\xi\xi}$ where ξ corresponds to the direction orthogonal to the gradient. Therefore, in order to discretize this operator, it is enough to multiply the discretization of $u_{\xi\xi}$ presented in the above section by $|Du|^2$. We choose $\lambda_0(\theta)$ given by (19.5) because it corresponds to a trigonometric polynomial of degree two and then multiplying it by $|Du|^2$ the coefficients $\eta_i = |Du|^2\lambda_i, i = 0, 1, 2, 3, 4,$ are polynomials of degree two with respect to u_x and u_y. Indeed, we obtain for $\theta \in [0, \frac{\pi}{4}]$

$$(|Du|^2 u_{\xi\xi})_{i,j} = \frac{1}{\Delta x^2}(-4\eta_0 u_{i,j} + \eta_1(u_{i+1,j} + u_{i-1,j}) + \eta_2(u_{i,j+1} + u_{i,j-1})$$

$$+\eta_3(u_{i-1,j-1} + u_{i+1,j+1}) + \eta_4(u_{i-1,j+1} + u_{i+1,j-1})) + O(\Delta x^2)$$

where $\eta_0, \eta_1, \eta_2, \eta_3, \eta_4$ are given by

$$\begin{cases} \eta_0 = 0.25(2u_x^2 + u_y^2 - u_x u_y) \\ \eta_1 = 0.5(2u_x^2 - u_y^2 - u_x u_y) \\ \eta_2 = 0.5(u_y^2 - u_x u_y) \\ \eta_3 = 0.25(u_y^2 + 3u_x u_y) \\ \eta_4 = 0.25(u_y^2 - u_x u_y) \end{cases}$$

Finally, the finite difference scheme for the A.M.S.S. equation is

$$u_{i,j}^{n+1} = u_{i,j}^n + \Delta t(|Du^n|^2 u_{\xi\xi}^n)_{i,j}^{\frac{1}{3}} \tag{19.9}$$

We have tested this algorithm and we have noticed that in this case the condition for the experimental stability (in the sense presented in the above subsection) is

$$\frac{\Delta t}{\Delta x^2} \leq \frac{1}{10}.$$

Remark. The finite difference schemes presented above are consistent. Contrast invariance can only be obtained asymptotically by taking a very small time step Δt. The experimental results presented in Figures 19.3 and **??** have been obtained by using these schemes with $\Delta t = 0.1$ in the case of mean curvature motion and $\Delta t = 0.01$ in the case of affine curvature motion. Indeed, while experimental stability is achieved with $\Delta t \leq 0.1$, the experimental affine invariance needs $\Delta t < 0.05$ (see Figure **??**.)

19.3 IL MANQUE UNE EXPERIENCE AMSS SUR L'INVARIANCE AFFINE!

19.4 Numerical normalization of scale.

(or Relation between scale and the number of iterations).

The case of the curvature motion. Setting the distance between pixels Δx to 1, the scale achieved with N iterations is simply $N \times \Delta t$. Now, the scale t associated with the PDE is somewhat arbitrary : It has no geometric meaning. In order to get it, we need a rescaling $T \to t(T)$ which we will call scale normalization.

A good way to perform this scale normalization is to define the correspondence $t(T)$ as the time for which a circle with initial radius T vanishes under curvature motion. Such a circle moves at a speed equal to its curvature, which is the inverse of its radius. Thus have for a disk with radius $R(t)$

$$\frac{dR(t)}{dt} = -\frac{1}{R(t)}$$

which yields

$$\frac{1}{2}(R^2(0) - R^2(t)) = t.$$

Exercise 19.3. Check this relation! ■

The disk disappears when $R(t) = 0$, that is, at scale $T = R^2(0)/2$. This last relation gives a scale normalization: In order to arrive at the normalized scale T (at which any disk with radius less or equal to T vanishes), we have to evolve the PDE at $t = N\Delta t = T^2/2$. This fixes the number of needed iterations as

$$N = T^2/2\Delta t.$$

The case of AMSS We can perform similar calculations. The radius of an evolving disk satisfies

$$\frac{dR(t)}{dt} = -\frac{1}{R(t)^{\frac{1}{3}}}$$

which yields

$$\frac{3}{4}(R^{\frac{4}{3}}(0) - R^{\frac{4}{3}}(t)) = t$$

The disappearance time is therefore $t = \frac{3}{4}R^{\frac{4}{3}}$. As for the curvature motion, we define the normalized scale T as the one at which a disk with radius T vanishes. In order to achieve this scale T, the needed number of iterations is

$$N = \frac{3}{4\Delta t}T^{\frac{4}{3}}.$$

Exercise 19.4. Check the last two formulas! ■

19.5 Contrast invariance and the level set extension

Both schemes (M.C.M and A.M.S.S) presented above are not numerically contrast invariant. We have seen that a contrast operator cannot create new gray levels (Exercise 13.22.) Now, starting with a binary image u^0 and applying a scheme defined by such a formula as

$$u^{n+1} = u^n + \Delta t(...)$$

does not ensure that u^{n+1} will be also a binary image.

A natural idea to overcome this problem is the following. Starting with a binary image (with values 0 and 1): apply the scheme until the expected scale is achieved, then threshold the obtained image at $\lambda = \frac{1}{2}$. This of course works only for binary images. However, the level set extension (see Section 13.3) gives us the key to extend this to general images.

The contrast invariance can be fully obtained by first applying the finite difference scheme on each level set (considered as a binary image) separately. Then by the superposition principle the evolved image is computed from the evolved level sets. The procedure is the following :

Algorithm starting with an image u_0 and evolving it to $u(t, \mathbf{x})$ by curvature motion

For each $\lambda \in [0, 255]$, in increasing order:

- Let $v_\lambda(\mathbf{x})$ be the characteristic function of $\mathcal{X}_\lambda u_0$. (This function is equal to 1 inside the level set and to 0 outside.)
- Apply to v_λ the MCM or AMSS FDS-scheme until scale t. This yields the images $w_\lambda(t, .)$.
- Set $u(t, \mathbf{x}) = \lambda$ at each point $(t, \mathbf{x})$ where $w_\lambda(t, \mathbf{x}) \geq 0.5$.

19.6 Problems at extrema

For MCM and AMSS we raised the question of performing numerically the equation when $|Du| = 0$. For MCM the right hand part of the equation is simply not defined. For AMSS one can set by continuity as $Du \to 0$, $(|Du^n|^2 u^n_{\xi\xi})^{\frac{1}{3}}_{i,j} = 0$. Now, numerically, this would imply that isolated black or white extrema will not evolve by the equation. We know that this is simply wrong, since small sets collapse by curvature motion.

In short, FDS for MCM and AMSS are not consistent with the equation at extrema. In Figure 19.6, we added to an image a strong “salt and pepper” noise. More than one fourth of the pixels have been given a uniform random value in $[0, 255]$ and most of them have become local extrema. Not only these values do not evolve but they contaminate their neighboring pixels. There are easy ways to avoid this spurious effect :

- One can first zoom by 2 the image by duplicating pixels. This, however, multiplies by 16 the number of computations.
- One can first remove pixels extrema with diameter k since they must anyway disappear by the equation at normalized scale $\frac{k}{2}$.
- One can use the level set method. This multiplies the number of computations by the initial number of gray-levels.

All of these solutions are efficient, as shown in Figure 19.6.

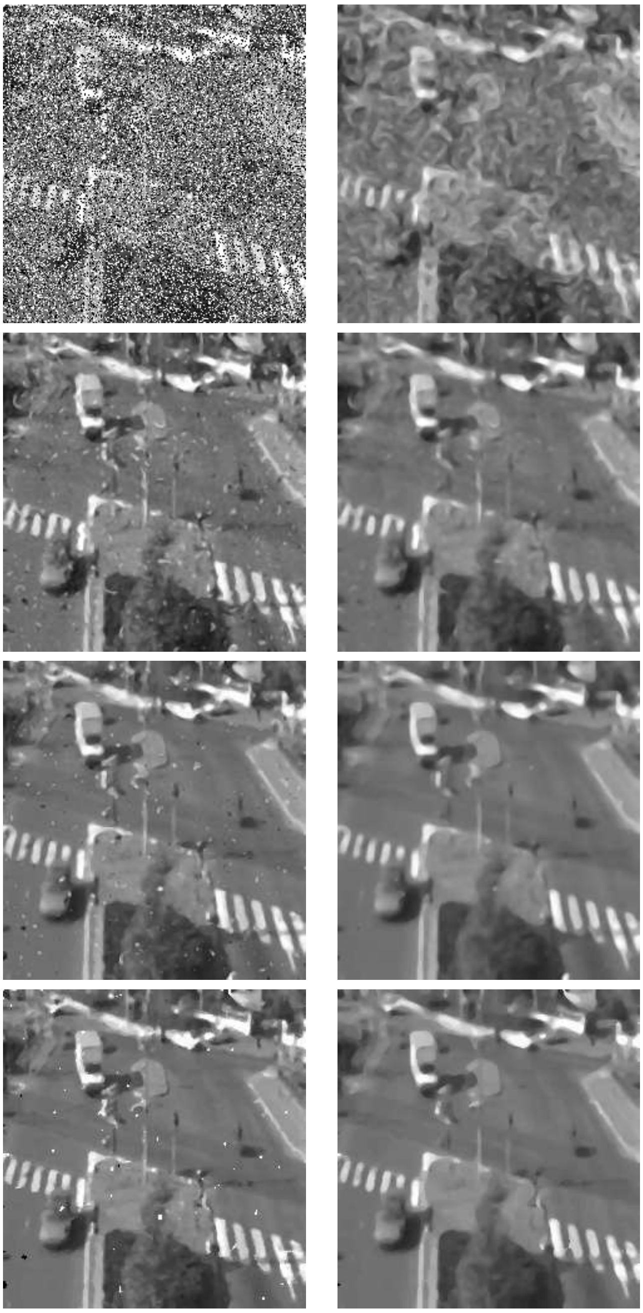

Figure 19.6: Various implementations of curvature motion on a noisy image. Top left : image with 40% pixels replaced by a uniform random value in $[0, 255]$. Top right: application of the finite difference scheme (FDS) at normalized scale 3. On the lines 2 to 4, we see various solutions to the disastrous diffusion of extrema. On the left the image is processed at normalized scale 1 and on the right at normalized scale 3. Second line: FDS applied on the image previously zoomed by a factor 2; third line: FDS applied on the image after its extrema have been "killed" (the reference area is given by the area of the disk vanishing at the desired scale). Fourth line: FDS applied separately on each level set and application of the threshold superposition principle. The third scheme offers a good speed-quality compromise.

19.7 Conclusion

We have seen that standard finite difference schemes are easy to implement but cannot handle properly the invariance properties satisfied by the equations.

1. There is no finite difference scheme that insures the monotonicity. This leads to slightly oscilatory solutions.
2. No full contrast invariance. For instance FDS create new grey levels and blur edges. Also, a spurious diffusion occurs around the image extrema. However this last problem was dealt with efficiently in the previous section. The full contrast invariance has been restored by the level set extension of the numerical schemes.
3. The worst drawback of FDS is the lack of Euclidean or affine invariance which can be only approximately obtained by grid local schemes. A much more clever strategy to achieve full invariance is to evolve all level curves of the image and the reconstruct it. This is the aim of Section **??**, but we have already seen in Chapter 7 how to evolve curves by curvature.

19.8 Comments and references

Difference schemes for the curvature motion and the AMSS The presented difference scheme follows mainly [99], improved in Alvarez et al. [9]. This scheme is somehow optimal among the rotationally invariant numerical schemes for curvature motion and the AMSS. Now, this presentation is specific of those two motions, while other many authors have analysed more general nonlinear anisotropic diffusions in image processing, namely Acton [2], Kacur and Mikula [122, 123]. Weickert and the Ütrecht school [195, 255, 1, 258] address many aspects of implementation of nonlinear scale spaces, namely speed, parallelism and robustness. Crandall and Lions [61] also proposed a finite difference scheme for mean curvature motion, valid in any dimension. Sethian's book [232] explains how to implement fast the motion of a curve or surface by the so called "level set method", where a distance function to the curve or surface is evolved. Dynamic programming allows a fast implementation (the "fast marching method").

Bibliography

[1] *Parallel implementations of AOS schemes: A fast way of nonlinear diffusion filtering.*

[2] S. Acton, *Multigrid anisotropic diffusion*, Image Processing, IEEE Transactions on, 7 (1998), pp. 280–291.

[3] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski, *Photographing long scenes with multi-viewpoint panoramas*, International Conference on Computer Graphics and Interactive Techniques, (2006), pp. 853–861.

[4] G. J. Agin, *Representation and Description of Curved Objects*, PhD thesis, Stanford U., October 1972.

[5] F. Almgren, J. E. Taylor, and L. Wang, *Curvature-driven flows: A variational approach*, SIAM J. Control Optimization, 31 (1993), pp. 387–438.

[6] L. Alvarez, Y. Gousseau, and J.-M. Morel, *The size of objects in natural and artificial images*, in Advances in Imaging and Electron Physics, vol. 111, Academic Press, 1999, pp. 167–242.

[7] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel, *Axioms and fundamental equations of image processing*, Arch. Rational Mech. Anal., 16 (1993), pp. 200–257.

[8] L. Alvarez, P.-L. Lions, and J.-M. Morel, *Image selective smoothing and edge detection by nonlinear diffusion (ii)*, SIAM J. Numer. Anal., 29 (1992), pp. 845–866.

[9] L. Alvarez and J.-M. Morel, *Formalization and computational aspects of image analysis*, Acta Numerica, (1999), pp. 1–59.

[10] L. Ambrosio, V. Caselles, S. Masnou, and J.-M. Morel, *The connected components of caccioppoli sets and applications to image processing*, J. Eur. Math. Soc., 3 (2001), pp. 213–266.

[11] L. Ambrosio, V. Caselles, S. Masnou, and J.-M. Morel, *Connected components of sets of finite perimeter and applications to image processing*, J. Eur. Math. Soc., 3 (2001), pp. 39–92.

[12] F. Andreu, C. Ballester, V. Caselles, and J. M. Mazon, *Minimizing total variation flow*, C. R. Acad. Sci. Paris Sér. I, 331 (2000), pp. 867–872.

[13] F. Andreu, C. Ballester, V. Caselles, and J. M. Mazon, *Minimizing total variation flow*, Differential and Integral Equations, 4 (2001), pp. 321–360.

[14] S. B. Angenent, G. Sapiro, and A. Tannenbaum, *On the affine heat flow for nonconvex curves*, J. Amer. Math. Soc., 11 (1998), pp. 601–634.

[15] A. Anjulan and N. Canagarajah, *Affine Invariant Feature Extraction Using Symmetry*, Advanced Concepts for Intelligent Vision Systems: 7th International Conference, ACIVS 2005, Antwerp, Belgium, September 20-23, 2005: Proceedings, (2005).

[16] G. R. Arce and M. P. McLoughlin, *Theoretical analysis of the max/median filter*, 35 (1987), pp. 60–69.

[17] H. Asada and M. Brady, *The curvature primal sketch*, IEEE Trans. Patt. Anal. Mach. Intell., 8 (1986), pp. 2–14.

[18] A. Ashbrook, N. Thacker, P. Rockett, and C. Brown, *Robust recognition of scaled shapes using pairwise geometric histograms*, Proc. BMVC, (1995), pp. 503–512.

[19] J. Astola, P. Heinonen, and Y. Neuvo, *On root structures of median and median-type filters*, IEEE Trans. Acoust. Speech Signal Process., 35 (1987), pp. 1199–1201.

[20] F. ATTNEAVE, *Some informational aspects of visual perception.*, Psychol Rev, 61 (1954), pp. 183–93.

[21] C. Ballester, V. Caselles, and P. Monasse, *The tree of shapes of an image*, tech. rep., CMLA, Ecole Normale Supérieure-Cachan, France, 2001. Available at `http://www.cmla.ens-cachan.fr/Cmla/index.html`.

[22] C. Ballester, E. Cubero-Castan, M. González, and J.-M. Morel, *Fast contrast invariant image intersection.* Preprint CEREMADE, 1998.

[23] J. A. Bangham, P. Ling, and R. Young, *Multiscale recursive medians, scale-space, and transforms with applications to image processing*, IEEE Trans. Image Process., 5 (1996).

[24] G. Barles, *Remarks on a Flame Propagation Model*, Institut national de recherche en informatique et en automatique, 1985.

[25] A. Baumberg, *Reliable feature matching across widely separated views*, Proc. IEEE CVPR, 1 (2000), pp. 774–781.

[26] H. Bay, T. Tuytelaars, and L. Van Gool, *Surf: Speeded up robust features*, European Conference on Computer Vision, 1 (2006), pp. 404–417.

[27] S. Belongie, J. Malik, and J. Puzicha, *Shape Matching and Object Recognition Using Shape Contexts*, IEEE Trans. Pattern Anal. Mach. Intell., (2002).

[28] M. Bennewitz, C. Stachniss, W. Burgard, and S. Behnke, *Metric Localization with Scale-Invariant Visual Features Using a Single Perspective Camera*, European Robotics Symposium 2006, (2006).

[29] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, *Robust anisotropic diffusion*, IEEE Trans. Image Process., 7 (1998), pp. 421–432.

[30] J. Blom, B. T. te Haar Romeny, A. Bel, and J. J. Koenderink, *Spatial derivatives and the propagation of noise in Gaussian scale-space*, Tech. Rep. 3DCV 91-03, U. of Utrecht, Dept. of Medical and Physiological Physics, 1991.

[31] K. A. Brakke, *The motion of a surface by its mean curvature*, Mathematical Notes, 20, Princeton University Press, Princeton, NJ, 1978.

[32] L. Breiman, *Probability*, Addison-Wesley, 1968.

[33] H. Brezis, *Analyse Fonctionnelle, Théorie et Applications*, Masson, Paris, 1983.

[34] M. Brown and D. Lowe, *Recognising panoramas*, Proc. ICCV, 1 (2003), p. 3.

[35] D. R. K. Brownrigg, *The weighted median filter*, Communications of the ACM, 27 (1984), pp. 807–818.

[36] P. J. Burt and E. H. Adelson, *The laplacian pyramid as a compact image code*, IEEE Trans. Comm., 31 (1983), pp. 532–540.

[37] A. R. Butz, *A class of rank order smoothers*, IEEE Trans. Acoust. Speech Signal Process., 34 (1986), pp. 157–165.

[38] J. F. Canny, *A computational approach to edge detection*, IEEE Trans. Patt. Anal. Mach. Intell., 8 (1986), pp. 679–698.

[39] F. Cao, J. Delon, A. Desolneux, P. Musé, and F. Sur, *A unified framework for detecting groups and application to shape recognition*, Rapport interne, 1746, pp. 1166–8687.

[40] F. Cao, Y. Gousseau, P. Muse, F. Sur, and J. Morel, *Accurate estimates of false alarm number in shape recognition*, tech. rep., Technical report, Cachan, France (http://www. cmla. enscachan. fr/Cmla/), 2004. 4, 5.

[41] F. Cao, J. Lisani, J. Morel, P. Muse, and F. Sur, *A Theory of Shape Identification*, Springer Verlag, 2008.

[42] L. J.-L. M. J.-M. M. P. S. F. Cao, F., *A Theory of Shape Identification*, Springer Verlag, 2008.

[43] R. A. Carmona and S. F. Zhong, *Adaptive smoothing respecting feature directions*, IEEE Trans. Image Process., 7 (1998), pp. 353–358.

[44] V. Caselles, F. Catté, T. Coll, and F. Dibos, *A geometric model for active contours*, Numerische Mathematik, 66 (1993), pp. 1–31.

[45] V. Caselles, B. Coll, and J.-M. Morel, *Topographic maps and local contrast changes in natural images*, International Journal of Computer Vision, volume=.

[46] V. Caselles, B. Coll, and J.-M. Morel, *Topographic maps and local contrast changes in natural images*, Int. J. Comput. Vision, 33 (1999), pp. 5–27.

[47] V. Caselles, R. Kimmel, and G. Sapiro, *Geodesic active contours*, Int. J. Comput. Vision, 1 (1997), pp. 61–79.

[48] V. Caselles, J. L. Lisani, J.-M. Morel, and G. Sapiro, *Shape preserving local histogram modification*, 1999.

[49] V. Caselles, J.-M. Morel, and C. Sbert, *An axiomatic approach to image interpolation*, IEEE Trans. Image Process., 7 (1998), pp. 376–386.

[50] V. Caselles, G. Sapiro, and D. H. Chung, *Vector median filters, vector morphology, and coupled PDE's: Theoretical connections*, J. Math. Imaging Vision, 12 (2000), pp. 109–120.

[51] E. Chang, *EXTENT: fusing context, content, and semantic ontology for photo annotation*, Proceedings of the 2nd international workshop on Computer vision meets databases, (2005), pp. 5–11.

[52] R. C. Chen and P. T. Yu, *On the optimal design of rational rank selection filters for image restoration*, in Proceedings International Conference on Image Processing, 1996, p. 16A9.

[53] Y. G. Chen, Y. Giga, and S. Goto, *Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations*, Proc. Japan Acad. Ser. A Math. Sci., 65 (1989), pp. 207–210.

[54] Y. G. Chen, Y. Giga, and S. Goto, *Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations*, J. Differential Geom., 33 (1991), pp. 749–786.

[55] F. Cheng and A. N. Venetsanopoulos, *An adaptive morphological filter for image processing*, IEEE Trans. Image Process., 1 (1992), pp. 533–539.

[56] H. Cornelius and G. Loy, *Detecting Bilateral Symmetry in Perspective*, 5th Workshop on Perceptual Organization in Computer Vision, (2006), pp. 191–198.

[57] R. Courant and F. John, *Introduction to Calculus and Analysis*, Springer, 2000. Reprint of the 1989 edition.

[58] I. Cox, S. Roy, and S. Hingorani, *Dynamic histogram warping of image pairs for constant image brightness*, Image Processing, 1995. Proceedings, International Conference on, 2 (1995).

[59] E. Coyle, N. Gallagher, S. Bass, J. Fitch, and R. Harber, *Digital filtering by threshold decomposition*, 1987. U.S. Patent 4,868,773.

[60] E. J. Coyle, *Rank order operators and the mean absolute error criterion*, IEEE Trans. Acoust. Speech Signal Process., 36 (1988), pp. 63–76.

[61] M. Crandall and P. Lions, *Convergent difference schemes for nonlinear parabolic equations and mean curvature motion*, Numerische Mathematik, 75 (1996), pp. 17–41.

[62] M. G. Crandall, H. Ishii, and P.-L. Lions, *User's guide to viscosity solutions of second order partial differential equations*, Bull. Amer. Math. Soc. (N.S.), 27 (1992), pp. 1–67.

[63] J. Damon, *Generic structure of two dimensional images under Gaussian blurring*, SIAM J. Appl. Math., 59 (1998), pp. 97–138.

[64] P. E. Danielsson, *Getting the median faster*, Computer Graphics Image Processing, 17 (1981), pp. 71–78.

[65] J. G. Daugman, *Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by 2D visual cortical filters*, Journal of the Optical Society of America, 2 (1985), pp. 1160–1169.

[66] E. R. Davies, *On the noise suppression and image enhancement characteristics of the median, truncated median and mode filters*, Pattern Recognition Letters, 7 (1988), pp. 87–97.

[67] J. Delon, *Midway Image Equalization*, Journal of Mathematical Imaging and Vision, 21 (2004), pp. 119–134.

[68] J. Delon, *Movie and video scale-time equalization application to flicker reduction*, Image Processing, IEEE Transactions on, 15 (2006), pp. 241–248.

[69] R. Deriche, *Using canny's criteria to derive a recursively implemented optimal edge detector*, Int. J. Comput. Vision, 1 (1987), pp. 167–187.

[70] G. Dudek, *Shape Representation from Curvature*, PhD thesis, U. Toronto, 1990.

[71] G. Dudek and J. K. Tsotsos, *Robustly recognizing curves using curvature-tuned smoothing*, in IEEE Conference on Computer Vision and Pattern Recognition, 1991, pp. 35–41.

[72] C. L. Epstein and M. Gage, *The curve shortening flow*, in Wave motion: theory, modelling, and computation, Springer, NY, 1987, pp. 15–59. Proc. Conf. Hon. 60th Birthday P. D. Lax (Berkeley, CA, 1986) Math. Sci. Res. Inst. Publ. 7.

[73] L. C. Evans, *Partial Differential Equations*, Graduate Studies in Mathematics, 19, American Mathematical Society, Providence, RI, 1998.

[74] L. C. Evans and J. Spruck, *Motion of level sets by mean curvature. I.*, J. Differential Geom., 33 (1991), pp. 635–681.

[75] L. C. Evans and J. Spruck, *Motion of level sets by mean curvature. II.*, Trans. Am. Math. Soc., 330 (1992), pp. 321–332.

[76] L. C. Evans and J. Spruck, *Motion of level sets by mean curvature. III.*, J. Geom. Anal., 2 (1992), pp. 121–150.

[77] L. C. Evans and J. Spruck, *Motion of level sets by mean curvature. IV.*, J. Geom. Anal., 5 (1995), pp. 79–116.

[78] F. Falzon and G. Giraudon, *Singularity analysis and derivative scale-space*, in IEEE Computer Vision and Pattern Recognition, 1994, pp. 245–250.

[79] Q. Fan, K. Barnard, A. Amir, A. Efrat, and M. Lin, *Matching slides to presentation videos using SIFT and scene background matching*, Proceedings of the 8th ACM international workshop on Multimedia information retrieval, (2006), pp. 239–248.

[80] L. Fevrier, *A wide-baseline matching library for Zeno*, Technical report, (2007).

[81] W. J. Firey, *Shapes of worn stones*, Mathematika, 21 (1974), pp. 1–11.

[82] J. P. Fitch, E. J. Coyle, and N. C. Gallagher, Jr., *Median filtering by threshold decomposition*, IEEE Trans. Acoust. Speech Signal Process., 32 (1984), pp. 1183–1188.

[83] L. M. J. Florack, *Image Structure*, vol. 10 of Computational Imaging and Vision, Kluwer Academic Publishers, Dordrecht, NL, 1997.

[84] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever, *Linear scale-space*, J. Math. Imaging Vision, 4 (1994), pp. 325–351.

[85] J. Foo and R. Sinha, *Pruning SIFT for scalable near-duplicate image matching*, Proceedings of the eighteenth conference on Australasian database-Volume 63, (2007), pp. 63–71.

[86] G. Fritz, C. Seifert, M. Kumar, and L. Paletta, *Building detection from mobile imagery using informative SIFT descriptors*, Lecture notes in computer science, pp. 629–638.

[87] D. Gabor, *Information theory in electron microscopy*, Journal of Laboratory Investigation, 14 (1965), pp. 801–807. The Journal of Laboratory Investigation was published by the International Academy of Pathology.

[88] M. Gage, *An isometric inequality with applications to curve shortening*, Duke Math. J., 50 (1983), pp. 1225–1229.

[89] M. Gage, *Curve shortening makes convex curves circular*, Invent. Math., 76 (1984), pp. 357–364.

[90] M. Gage and R. S. Hamilton, *The heat equation shrinking convex plane curves*, J. Differential Geom., 23 (1986), pp. 69–96.

[91] C. Gasquet and P. Witomski, *Fourier Analysis and Applications: Filtering, Numerical Computation, Wavelets*, Springer Verlag, 1999.

[92] E. D. Giorgi and L. Ambrosio, *Un nuovo tipo di funzionale del calcolo delle variazioni*, Atti. Accad. Naz. Lincei, s.8 (1988), pp. 199–210.

[93] M. J. E. Golay, *Smoothing data by least squares procedures and by filtering*, IEEE Trans. Computer, 21 (1972), pp. 299–301.

[94] G. Golub and C. Van Loan, *Matrix computations*, 1996.

[95] R. Gonzalez and R. Woods, *Digital image processing*, Addison-Wesley Reading, Mass, 1987.

[96] I. Gordon and D. Lowe, *What and Where: 3D Object Recognition with Accurate Pose*, Lecture Notes in Computer Science, 4170 (2006), p. 67.

[97] M. Grabner, H. Grabner, and H. Bischof, *Fast Approximated SIFT*, in Asian Conference on Computer Vision, Hyderabad, India, 2006, pp. 918–927.

[98] M. A. Grayson, *The heat equation shrinks embedded plane curves to round points*, J. Differential Geom., 26 (1987), pp. 285–314.

[99] F. Guichard, *Axiomatisation des analyses multi-échelles d'images et de films - Axiomatization of images and movies scale-space.*, PhD thesis, Univ. Paris IX-Dauphine, 1994.

[100] F. Guichard and J. Morel, *Mathematical morphology almost everywhere*, Proceedings of ISMM, pp. 293–303.

[101] R. M. Haralick, *Digital step edges from zero-crossing of second directional derivatives*, IEEE Trans. Patt. Anal. Mach. Intell., 6 (1984), pp. 58–68.

[102] R. M. Haralick, S. R. Sternberg, and X. Zhuang, *Image analysis using mathematical morphology*, IEEE Trans. Patt. Anal. Mach. Intell., 9 (1987), pp. 532–550.

[103] R. C. Hardie and K. E. Barner, *Rank conditioned rank selection filters for signal restoration*, IEEE Trans. Image Process., 3 (1994), pp. 192–206.

[104] J. Hare and P. Lewis, *Salient regions for query by image content*, Image and Video Retrieval: Third International Conference, CIVR, (2004), pp. 317–325.

[105] C. Harris and M. Stephens, *A combined corner and edge detector*, Alvey Vision Conference, 15 (1988), p. 50.

[106] H. J. A. M. Heijmans and C. Ronse, *The algebraic basis of mathematical morphology: I. dilations and erosions*, Computer Vision, Graphics, and Image Processing, 50 (1990), pp. 245–295.

[107] C. W. Helstrom, *Image restoration by the method of least squares*, J. Opt. Soc. Amer., 57 (1967), pp. 297–303.

[108] J. M. Hereford and W. T. Rhodes, *Nonlinear optical image filtering by time-sequential threshold decomposition*, Opt. Eng., 27 (1988), pp. 274–279.

[109] B. K. P. Horn and E. J. Weldon, Jr., *Filtering closed curves*, IEEE Trans. Patt. Anal. Mach. Intell., 8 (1986), pp. 665–668.

[110] C. T. Huang and O. R. Mitchell, *A Euclidean distance transform using grayscale morphology decomposition*, IEEE Trans. Patt. Anal. Mach. Intell., 16 (1994), pp. 443–448.

[111] T. S. Huang, G. J. Yang, and G. Y. Tang, *A fast two-dimensional median filtering algorithm*, IEEE Trans. Acoust. Speech Signal Process., 27 (1979), pp. 13–18.

[112] D. H. Hubel and T. N. Wiesel, *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex*, J. Physiol., 160 (1962), pp. 106–154.

[113] R. A. Hummel, *Representations based on zero-crossing in scale-space*, in IEEE Computer Vision and Pattern Recognition, 1986, pp. 204–209.

[114] R. A. Hummel, B. Kimia, and S. W. Zucker, *Gaussian blur and the heat equation: Forward and inverse solutions*, in IEEE Computer Vision and Pattern Recognition, 1983, pp. 416–421.

[115] R. A. Hummel, B. Kimia, and S. W. Zucker, *Deblurring gaussian blur*, Computer Vision, Graphics, and Image Processing, 38 (1987), pp. 66–80.

[116] R. A. Hummel and R. Moniot, *Reconstructions from zero-crossings in scale-space*, IEEE Trans. Acoust. Speech Signal Process., 37 (1989), pp. 2111–2130.

[117] H. H. S. Ip, D. J. Potter, and D. S. Lebedev, *Impulse noise cleaning by iterative threshold median filtering*, Pattern Recognition Letters, 2 (1983), pp. 89–93.

[118] H. Ishii and P. Souganidis, *Generalized motion of noncompact hypersurfaces with velocity having arbitrary growth on the curvature tensor*, Tohoku Math. J. II, 47 (1995), pp. 227–250.

[119] S. Jaffard, Y. Meyer, and R. D. Ryan, *Wavelets: Tools for Science and Technology*, Society for Industrial and Applied Mathematics, Philadelphia, 2001.

[120] A. M. Joseph, *Building local safety maps for a wheelchair robot using vision and lasers.*

[121] B. Julesz, *A method of coding TV signals based on edge detection*, Bell System Tech. J. 38 (1959), pp. 1001–1020.

[122] J. Kacur and K. Mikula, *Solution of nonlinear diffusion appearing in image smoothing and edge detection*, Applied Numerical Mathematics, 17 (1995), pp. 47–59.

[123] J. Kacur and K. Mikula, *Slowed Anisotropic Diffusion*, Scale-Space Theory in Computer Vision: First International Conference, Scale-Space'97, Utrecht, The Netherlands, July 2-4, 1997: Proceedings, (1997).

[124] T. Kadir, A. Zisserman, and M. Brady, *An Affine Invariant Salient Region Detector*, in European Conference on Computer Vision, 2004, pp. 228–241.

[125] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models*, Int. J. Comput. Vision, 1 (1998), pp. 321–331.

[126] Y. Ke and R. Sukthankar, *PCA-SIFT: A more distinctive representation for local image descriptors*, Proc. CVPR, 2 (2004), pp. 506–513.

[127] S. Kichenassamy, *The Perona–Malik paradox*, SIAM J. Appl. Math., 57 (1997), pp. 1328–1342.

[128] J. Kim, S. Seitz, and M. Agrawala, *Video-based document tracking: unifying your physical and electronic desktops*, Symposium on User Interface Software and Technology: Proceedings of the 17 th annual ACM symposium on User interface software and technology, 24 (2004), pp. 99–107.

[129] V. Kim and L. Yaroslavskii, *Rank algorithms for picture processing*, Computer Vision, Graphics, and Image Processing, 35 (1986), pp. 234–258.

[130] B. Kimia, A. Tannenbaum, and S. W. Zucker, *On the evolution of curves via a function of curvature. I. The classical case*, J. Math. Anal. Appl., 163 (1992), pp. 438–458.

[131] B. Kimia, A. R. Tannenbaum, and S. W. Zucker, *Shapes, shocks, and deformations I: The components of 2-dimensional shape and the reaction-diffusion space*, Int. J. Comput. Vision, 15 (1995), pp. 189–224.

[132] R. Kimmel and A. M. Bruckstein, *Shape offsets via level sets.*, Computer Aided Design, 5 (1993), pp. 154–162.

[133] R. Kimmel, D. Shaked, N. Kiryati, and A. M. Bruckstein, *Skeletonization via distance maps and level sets*, Computer Vision and Image Understanding, 3 (1995), pp. 382–391.

[134] J. J. Koenderink, *The structure of images*, Biological Cybernetics, 50 (1984), pp. 363–370.

[135] J. J. Koenderink and A. J. van Doorn, *Dynamic shape*, Biological Cybernetics, 53 (1986), pp. 383–396.

[136] L. S. G. Kovasznay and H. M. Joseph, *Image processing*, Proc. IRE, 43 (1955), p. 560.

[137] H. P. Kramer and J. B. Bruckner, *Iterations of a non-linear transformation for enhancement of digital images*, Pattern Recognition, 7 (1975).

[138] S. G. Krantz and H. R. Parks, *The Geometry of Domains in Space*, Birkäuser, 1999.

[139] A. S. Kronrod, *On functions of two variables*, Uspehi Matem. Nauk (N. S.), 5 (1950), pp. 24–134.

[140] P. D. Lax, *Numerical solution of partial differential equations*, Amer. Math. Monthly, 72 (1965), pp. 74–85.

[141] B. Lee, W. Chen, and E. Chang, *Fotofiti: web service for photo management*, Proceedings of the 14th annual ACM international conference on Multimedia, (2006), pp. 485–486.

[142] Y. H. Lee and S. A. Kassam, *Generalized median filtering and related nonlinear filtering techniques*, IEEE Trans. Acoust. Speech Signal Process., 33 (1985), pp. 672–683.

[143] H. Lejsek, F. Ásmundsson, B. Jónsson, and L. Amsaleg, *Scalability of local image descriptors: a comparative study*, Proceedings of the 14th annual ACM international conference on Multimedia, (2006), pp. 589–598.

[144] T. Lindeberg, *Scale-Space Theory in Computer Vision*, vol. 256 of Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, Dordrecht, NL, 1993.

[145] T. Lindeberg, *Scale-space theory: a basic tool for analyzing structures at different scales*, Journal of Applied Statistics, 21 (1994), pp. 225–270.

[146] T. Lindeberg and J. Garding, *Shape-adapted smoothing in estimation of 3-d depth cues from affine distortions of local 2-d brightness structure*, Proc. ECCV, (1994), pp. 389–400.

[147] M. Lindenbaum, M. Fischer, and A. M. Bruckstein, *On gabor's contribution to image enhancement*, Pattern Recognition, 27 (1994), pp. 1–8.

[148] J. Lisani, P. Monasse, and L. Rudin, *Fast Shape Extraction and Applications*, to appear, prepublication of CMLA ENS Cachan, 16 (2001).

[149] J.-L. Lisani, L. Moisan, P. Monasse, and J.-M. Morel, *Affine invariant mathematical morphology applied to a generic shape recognition algorithm*, in Proceedings of the International Symposium of Mathematical Morphology, Palo Alto, CA, 2000. Available at `http://pascal.monasse.free.fr/index.html`.

[150] D. Lowe, *SIFT Keypoint Detector: online demo http://www.cs.ubc.ca/∼lowe/keypoints/.*

[151] D. Lowe, *Distinctive image features from scale-invariant key points*, International Journal of Computer Vision, 60 (2004), pp. 91–110.

[152] G. Loy and J. Eklundh, *Detecting symmetry and symmetric constellations of features*, Proceedings of ECCV, 2 (2006), pp. 508–521.

[153] A. Mackworth and F. Mokhtarian, *Scale based description and recognition of planar curves and two-dimensional shapes*, IEEE Trans. Patt. Anal. Mach. Intell., 8 (1986), pp. 34–43.

[154] A. K. Mackworth and F. Mokhtarian.

[155] J. Maeda, *A characteristic property of space curves of constant first affine curvature*, Tohoku Math. J., 48 (1941), pp. 148–151.

[156] R. Malladi, J. A. Sethian, and B. C. Vemuri, *Shape modeling with front propagation: A level set approach*, IEEE Trans. Patt. Anal. Mach. Intell., 17 (1995), pp. 158–175.

[157] P. Maragos, *A representation theory for morphological image and signal processing*, IEEE Trans. Patt. Anal. Mach. Intell., 11 (1989), pp. 586–599.

[158] P. Maragos, *Differential morphology and image-processing*, IEEE Trans. Image Process., 5 (1996), pp. 922–937.

[159] P. Maragos, *Partial differential equations in image analysis: Continuous modeling, discrete processing*, in Eusipco, 1998.

[160] P. Maragos and R. W. Schafer, *Morphological filters. Part I: Their set-theoretic analysis and relations to linear shift-invariant filters*, IEEE Trans. Acoust. Speech Signal Process., 35 (1987), pp. 1153–1169.

[161] P. Maragos and R. W. Schafer, *Morphological filters. Part II: Their relations to median, order-statistic, and stack filters*, IEEE Trans. Acoust. Speech Signal Process., 35 (1987), pp. 1170–1184.

[162] P. Maragos and R. D. Ziff, *Threshold superposition in morphological image analysis systems*, IEEE Trans. Patt. Anal. Mach. Intell., 12 (1990), pp. 498–504.

[163] D. Marr, *Vision*, W. H. Freeman and Company, New York, NY, 1982.

[164] D. Marr, *Vision*, Freeman Publishers, 1982.

[165] D. Marr and E. Hildreth, *Theory of edge detection*, Proc. Roy. Soc. London Ser. B, 207 (1980), pp. 187–217.

[166] S. Masnou, *Image restoration involving connectedness*, in Proceedings of the 6^{th} International Workshop on Digital Image Processing and Computer Graphics, 1998.

[167] J. Matas, O. Chum, M. Urban, and T. Pajdla, *Wbs image matcher: online demo http://cmp.felk.cvut.cz/∼wbsdemo/demo/.*

[168] J. Matas, O. Chum, M. Urban, and T. Pajdla, *Robust wide-baseline stereo from maximally stable extremal regions*, Image and Vision Computing, 22 (2004), pp. 761–767.

[169] G. Matheron, *Random Sets and Integral Geometry*, John Wiley & Sons, New York, NY, 1975.

[170] B. Merriman, J. Bence, and S. Osher, *Diffusion generated motion by mean curvature*, in Computational Crystal Growers Workshop. Proceedings of the Geometry Center Workshop held in Minneapolis, Minnesota, February 22–28, 1992., J. E. Taylor, ed., Selected Lectures in Mathematics, American Mathematical Society, 1992, pp. 73–83.

[171] K. Mikolajczyk, *http://www.robots.ox.ac.uk/∼vgg/research/affine/.*

[172] K. Mikolajczyk and C. Schmid, *Indexing based on scale invariant interest points*, Proc. ICCV, 1 (2001), pp. 525–531.

[173] K. Mikolajczyk and C. Schmid, *An affine invariant interest point detector*, Proc. ECCV, 1 (2002), pp. 128–142.

[174] K. Mikolajczyk and C. Schmid, *A Performance Evaluation of Local Descriptors*, in International Conference on Computer Vision and Pattern Recognition, vol. 2, June 2003, pp. 257–263.

[175] K. Mikolajczyk and C. Schmid, *Scale and Affine Invariant Interest Point Detectors*, International Journal of Computer Vision, 60 (2004), pp. 63–86.

[176] K. Mikolajczyk and C. Schmid, *A Performance Evaluation of Local Descriptors*, IEEE Trans. PAMI, (2005), pp. 1615–1630.

[177] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool, *A Comparison of Affine Region Detectors*, International Journal of Computer Vision, 65 (2005), pp. 43–72.

[178] L. Moisan, *Affine plane curve evolution: A fully consistent scheme*, IEEE Trans. Image Process., 7 (1998), pp. 411–420.

[179] P. Monasse, *Contrast invariant image registration*, Proc. of the International Conf. on Acoustics, Speech and Signal Processing, Phoenix, Arizona, 6 (1999), pp. 3221–3224.

[180] P. Monasse, *Représentation morphologique d'images numériques et application au recalage*, PhD thesis, Ecole Normale Supérieure-Cachan, France, 2000. French and English versions available at `http://pascal.monasse.free.fr`.

[181] P. Monasse and F. Guichard, *Scale-space from a level lines tree*, Journal of Visual Communication and Image Representation, 11 (2000), pp. 224–236. Available at `http://pascal.monasse.free.fr`.

[182] P. Moreels and P. Perona, *Common-frame model for object recognition*, Proc. NIPS, (2004).

[183] P. Moreels and P. Perona, *Evaluation of Features Detectors and Descriptors based on 3D Objects*, International Journal of Computer Vision, 73 (2007), pp. 263–284.

[184] J. Morel and G. Yu, *On the consistency of the SIFT method*, Tech. Rep. Prepublication, CMLA, ENS Cachan, 2008.

[185] D. Mukherjee, A. Zisserman, and J. Brady, *Shape from symmetry—detecting and exploiting symmetry in affine images*, Philosophical Transactions: Physical Sciences and Engineering, 351 (1995), pp. 77–106.

[186] J. L. Mundy, *Object recognition: The search for representation*, in ORCV95, 1995, pp. 19–50.

[187] P. Musé, F. Sur, F. Cao, and Y. Gousseau, *Unsupervised thresholds for shape matching*, Image Processing, 2003. Proceedings. 2003 International Conference on, 2 (2003).

[188] P. Muse, F. Sur, F. Cao, and Y. Gousseau, *Unsupervised thresholds for shape matching*, Image Processing, 2003. Proceedings. 2003 International Conference on, 2 (2003).

[189] P. Musé, F. Sur, F. Cao, Y. Gousseau, and J. Morel, *An A Contrario Decision Method for Shape Element Recognition*, International Journal of Computer Vision, 69 (2006), pp. 295–315.

[190] P. Musé, F. Sur, F. Cao, Y. Gousseau, and J. Morel, *An A Contrario Decision Method for Shape Element Recognition*, International Journal of Computer Vision, 69 (2006), pp. 295–315.

[191] P. Musé, F. Sur, F. Cao, J. Lisani, and J. Morel, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, Mit Press, 2007.

[192] Y. Nakagawa and A. Rosenfeld, *A note on the use of local min and max operations in digital image processing*, IEEE Trans. Systems, Man and Cybernetics, 8 (1978), pp. 632–635.

[193] P. M. Narendra, *Noise reduction by median filtering*, in IEEE Conf. Pattern Recogn. Image Processing, 1978.

[194] A. Negre, H. Tran, N. Gourier, D. Hall, A. Lux, and J. Crowley, *Comparative study of People Detection in Surveillance Scenes*, Structural, Syntactic and Statistical Pattern Recognition, Proceedings Lecture Notes in Computer Science, 4109 (2006), pp. 100–108.

[195] W. Niessen, B. ter Haar Romeny, and M. Viergever, *Numerical Analysis of Geometry-Driven Diffusion Equations*, Geometry Driven Diffusion in Computer Vision, pp. 393–410.

[196] D. Nister and H. Stewenius, *Scalable recognition with a vocabulary tree*, Proc. CVPR, (2006), pp. 2161–2168.

[197] E. Ochoa, J. P. Allebach, and D. W. Sweeney, *Optical median filtering by threshold decomposition*, Appl. Opt., 26 (1987), pp. 252–260.

[198] S. Osher and L. I. Rudin, *Feature-oriented image enhancement using shock filters*, SIAM J. Numer. Anal., 27 (1990), pp. 919–940.

[199] S. Osher and J. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on the Hamilton–Jacobi formulation*, J. Comput. Phys., 79 (1988), pp. 12–49.

[200] H. C. Park and R. T. Chin, *Decomposition of arbitrarily-shaped morphological structuring elements*, IEEE Trans. Patt. Anal. Mach. Intell., 17 (1995), pp. 2–15.

[201] D. Pasquignon, *Computation of skeleton by PDE*, in International Conference in Image Processing, IEEE, ed., 1995.

[202] S. S. Perlman, S. Eisenhandler, P. W. Lyons, and M. J. Shumila, *Adaptive median filtering for impulse noise elimination in real-time TV signals*, IEEE Trans. Communications, 35 (1987), pp. 646–652.

[203] P. Perona and J. Malik, *Scale space and edge detection using anisotropic diffusion*, in Proc. Workshop on Computer Vision, Representation and Control, IEEE Computer Society, IEEE Press, Piscataway, NJ, 1987, pp. 16–22.

[204] P. Perona and J. Malik, *Scale space and edge detection using anisotropic diffusion*, IEEE Trans. Patt. Anal. Mach. Intell., 12 (1990), pp. 629–639.

[205] D. Pritchard and W. Heidrich, *Cloth Motion Capture*, Computer Graphics Forum, 22 (2003), pp. 263–271.

[206] J. Rabin, Y. Gousseau, and J. Delon, *A contrario matching of local descriptors*, Tech. Rep. hal-00168285, Ecole Nationale Supérieure des Télécommunications, Paris, France, 2007.

[207] L. Rawtani, *Image smoothing by median filtering – an efficient algorithm*, J.M.A.C.T., 28 (1995), pp. 93–101.

[208] C. S. Regazzoni and A. Teschioni, *A new approach to vector median filtering based on space-filling curves*, IEEE Trans. Image Process., 6 (1997), pp. 1025–1037.

[209] C. H. Richardson and R. W. Schafer, *A lower bound for structuring element decompositions*, IEEE Trans. Patt. Anal. Mach. Intell., 13 (1991), pp. 365–369.

[210] F. Riesz and B. Sz.-Nagy, *Functional Analysis*, Frederick Ungar Publishing Co., New York, 1955.

[211] F. Riggi, M. Toews, and T. Arbel, *Fundamental Matrix Estimation via TIP-Transfer of Invariant Parameters*, Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 02, (2006), pp. 21–24.

[212] C. Ronse and H. J. A. M. Heijmans, *The algebraic basis of mathematical morphology: II. openings and closings*, Computer Vision, Graphics, and Image Processing, 54 (1991), pp. 74–97.

[213] E. Rouy and A. Tourin, *A Viscosity Solutions Approach to Shape-From-Shading*, SIAM Journal on Numerical Analysis, 29 (1992), pp. 867–884.

[214] L. I. Rudin, *Images, Numerical Analysis of Singularities and Shock Filters*, PhD thesis, California Institute of Technology, Pasadena, CA, 1987.

[215] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 259–268.

[216] W. Rudin, *Real and Complex Analysis. Third edition*, McGraw-Hill Book Co., New York, 1987.

[217] J. Ruiz-del Solar, P. Loncomilla, and C. Devia, *A New Approach for Fingerprint Verification Based on Wide Baseline Matching Using Local Interest Points and Descriptors*, Lecture Notes in Computer Science, 4872 (2007), p. 586.

[218] P. Salembier, *Structuring element adaptation for morphological filters*, Journal of Visual Communication and Image Representation, 3 (1992), pp. 115–136.

[219] P. Salembier and J. Serra, *Flat zones filtering, connected operators and filters by reconstruction*, IEEE Trans. Image Process., 8 (1995).

[220] G. Sapiro, R. Kimmel, D. Shaked, B. Kimia, and A. M. Bruckstein, *Implementing continuous-scale morphology via curve evolution*, Pattern Recognition, 9 (1993), pp. 1363–1372.

[221] G. Sapiro and A. Tannenbaum, *Affine invariant scale space*, International Journal of Computer Vision, 11 (1993), pp. 25–44.

[222] G. Sapiro and A. Tannenbaum, *On invariant curve evolution and image analysis*, Indiana Univ. Math. J., 42 (1993), pp. 985–1009.

[223] G. Sapiro and A. Tannenbaum, *On affine plane curve evolution*, J. Funct. Anal., 119 (1994), pp. 79–120.

[224] F. Schaffalitzky and A. Zisserman, *Multi-view matching for unordered image sets, or How do I organize my holiday snaps?*, Proc. ECCV, 1 (2002), pp. 414–431.

[225] D. Schonfeld, *Optimal structuring elements for the morphological pattern restoration of binary images*, IEEE Trans. Patt. Anal. Mach. Intell., 16 (1994), pp. 589–601.

[226] P. Scovanner, S. Ali, and M. Shah, *A 3-dimensional SIFT descriptor and its application to action recognition*, Proceedings of the 15th international conference on Multimedia, (2007), pp. 357–360.

[227] S. Se, D. Lowe, and J. Little, *Vision-based mobile robot localization and mapping using scale-invariant features*, Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, 2 (2001).

[228] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, San Diego, CA, 1982.

[229] J. Serra, *Introduction to mathematical morphology*, Computer Vision, Graphics, and Image Processing, 35 (1986), pp. 283–305.

[230] J. Serra, *Image Analysis and Mathematical Morphology. Volume2: Theoretical Advances*, Academic Press, 1988.

[231] J. Sethian, *An analysis of flow propagation*, PhD thesis, UCLA, 1982.

[232] J. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1996.

[233] J. A. Sethian, *Curvature and the evolution of fronts*, Comm. Math. Phys., 101 (1985).

[234] F. Y. Shih and O. R. Mitchell, *A mathematical morphology approach to Euclidean distance transformation*, IEEE Trans. Image Process., 1 (1992), pp. 197–204.

[235] N. Snavely, S. Seitz, and R. Szeliski, *Photo tourism: exploring photo collections in 3D*, ACM Transactions on Graphics (TOG), 25 (2006), pp. 835–846.

[236] N. Sochen, R. Kimmel, and R. Malladi, *A general framework for low-level vision*, IEEE Trans. Image Process., 7 (1998), pp. 310–318.

[237] D. Struik, *Differential Geometry*, Addison-Wesley, 1950.

[238] M. P. Sudipta N Sinha, Jan-Michael Frahm and Y. Genc, *Gpu-based video feature tracking and matching*, EDGE 2006, workshop on Edge Computing Using New Commodity Architectures, Chapel Hill, (2006).

[239] T. Sun and Y. Neuvo, *Detail-preserving median based filters in image-processing*, Pattern Recognition Letters, 15.

[240] I. Svalbe and R. Jones, *The design of morphological filters using multiple structuring elements, part I: Openings and closings*, Pattern Recognition Letters, 13 (1992), pp. 123–129.

[241] H. D. Tagare and R. J. P. deFigueiredo, *Order filters*, Proceedings of IEEE, 73 (1985), pp. 163–165.

[242] M. E. Taylor, *Partial Differential Equations: Basic Theory*, Springer, New York, 1996.

[243] B. M. ter Haar Romeny, *Front-End Vision and Multiscale Image Analysis: Introduction to Scale-Space Theory*, Kluwer Academic Publishers, Dordrecht, NL, In preparation.

[244] T. Tuytelaars and L. Van Gool, *Wide baseline stereo matching based on local, affinely invariant regions*, British Machine Vision Conference, (2000), pp. 412–425.

[245] T. Tuytelaars and L. Van Gool, *Matching Widely Separated Views Based on Affine Invariant Regions*, International Journal of Computer Vision, 59 (2004), pp. 61–85.

[246] T. Tuytelaars, L. Van Gool, et al., *Content-based image retrieval based on local affinely invariant regions*, Int. Conf. on Visual Information Systems, (1999), pp. 493–500.

[247] L. Vacchetti, V. Lepetit, and P. Fua, *Stable Real-Time 3D Tracking Using Online and Offline Information*, IEEE Trans PAMI, (2004), pp. 1385–1391.

[248] R. van den Boomgaard, *Numerical Solution Schemes for Continuous-Scale Morphology*, Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision, (1999), pp. 199–210.

[249] R. van den Boomgaard and A. Smeulders, *The morphological structure of images: The differential equations of morphological scale-space*, IEEE Trans. Patt. Anal. Mach. Intell., 16 (1994), pp. 1101–1113.

[250] L. Van Gool, T. Moons, and D. Ungureanu, *Affine/Photometric Invariants for Planar Intensity Patterns*, Proceedings of the 4th European Conference on Computer Vision-Volume I-Volume I, (1996), pp. 642–651.

[251] M. Veloso, F. von Hundelshausen, and P. Rybski, *Learning visual object definitions by observing human activities*, Humanoid Robots, 2005 5th IEEE-RAS International Conference on, (2005), pp. 148–153.

[252] M. Vergauwen and L. Van Gool, *Web-based 3D Reconstruction Service*, Machine Vision and Applications, 17 (2005), pp. 411–426.

[253] L. Vincent, *Grayscale area openings and closings, their efficient implementation and applications*, in Proceedings of the 1^{st} Workshop on Mathematical Morphology and its Applications to Signal Processing, J. Serra and P. Salembier, eds., Barcelona, Spain, 1993, pp. 22–27.

[254] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton U. Press, Princeton, NJ, 1947.

[255] J. Weickert, *Nonlinear diffusion scale-spaces: from the continuous to the discrete setting*, Proc. ICAOS96, pp. 111–118.

[256] J. Weickert, *Anisotropic Diffusion in Image Processing*, ECMI Series, Teubner-Verlag, Stuttgart, 1998.

[257] J. Weickert and B. Benhamouda, *A semidiscrete nonlinear scale-space theory and its relation to the Perona–Malik paradox*, in Advances in Computer Vision, F. Solina, W. G. Kropatsch, R. Klette, and R. Bajcsy, eds., Springer, 1997, pp. 1–10.

[258] J. Weickert, B. Romeny, and M. Viergever, *Efficient and reliable schemes for nonlinear diffusion filtering*, Image Processing, IEEE Transactions on, 7 (1998), pp. 398–410.

[259] P. D. Wendt, E. J. Coyle, and N. C. Gallagher, Jr., *Stack filters*, IEEE Trans. Acoust. Speech Signal Process., 34 (1986), pp. 898–911.

[260] M. Wertheimer, *Untersuchungen zur Lehre der Gestalt, II*, Psychologische Forschung, 4 (1923), pp. 301–350.

[261] J. Wilburn, *Developments in generalized ranked-order filters*, Journal of the Optical Society of America, 15 (1998), pp. 1084–1099.

[262] A. P. Witkin, *Scale-space filtering*, in Proceedings of the International Joint Conference on Artificial Intelligence, Karlsruhe, Germany, 1983, Morgan Kaufmann Publishers, San Francisco, pp. 1019–1021.

[263] K. Yanai, *Image collector III: a web image-gathering system with bag-of-keypoints*, Proceedings of the 16th international conference on World Wide Web, (2007), pp. 1295–1296.

[264] G. Yang, C. Stewart, M. Sofka, and C. Tsai, *Alignment of challenging image pairs: Refinement and region growing starting from a single keypoint correspondence*, IEEE Trans. Pattern Anal. Machine Intell, (2007).

[265] J. Yao and W. Cham, *Robust multi-view feature matching from multiple unordered views*, Pattern Recognition, 40 (2007), pp. 3081–3099.

[266] O. Yli-Harja, J. Astola, and Y. Neuvo, *Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation*, IEEE Trans. Signal Process., 39 (1991), pp. 395–410.

[267] A. L. Yuille, *The creation of structure in dynamic shape*, in Proceedings Second International Conference on Computer Vision, 1988, pp. 685–689.

[268] A. L. Yuille and T. A. Poggio, *Fingerprint theorems for zero-crossings*, J. Opt. Soc. Amer., 2 (1985), pp. 683–692.

[269] A. L. Yuille and T. A. Poggio, *Scaling theorems for zero-crossings*, IEEE Trans. Patt. Anal. Mach. Intell., 8 (1986), pp. 15–25.

[270] X. Zhuang, *Decomposition of morphological structuring elements*, J. Math. Imaging Vision, 4 (1994), pp. 5–18.

[271] X. Zhuang and R. M. Haralick, *Morphological structural element decomposition*, Computer Vision, Graphics, and Image Processing, 35 (1986), pp. 370–382.