Nonparametric Statistical Methods for Image Segmentation and Shape Analysis

by

Junmo Kim

B.S., Electrical Engineering
Seoul National University, 1998

S.M., Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 2000

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering and Computer Science
at the
Massachusetts Institute of Technology

February, 2005

Signature of Author: _____

Department of Electrical Engineering and Computer Science
January 31, 2005

Certified by: _____

Alan S. Willsky
Professor of EECS, MIT
Thesis Supervisor

Accepted by: _____

Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Nonparametric Statistical Methods for Image Segmentation and Shape Analysis

by Junmo Kim

Submitted to the Department of Electrical Engineering
and Computer Science on January 31, 2005
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Image segmentation, the process of decomposing an image into meaningful regions, is a fundamental problem in image processing and computer vision. Recently, image segmentation techniques based on active contour models with level set implementation have received considerable attention. The objective of this thesis is in the development of advanced active contour-based image segmentation methods that incorporate complex statistical information into the segmentation process, either about the image intensities or about the shapes of the objects to be segmented. To this end, we use nonparametric statistical methods for modeling both the intensity distributions and the shape distributions.

Previous work on active contour-based segmentation considered the class of images in which each region can be distinguished from others by second order statistical features such as the mean or variance of image intensities of that region. This thesis addresses the problem of segmenting a more general class of images in which each region has a distinct arbitrary intensity distribution. To this end, we develop a nonparametric information-theoretic method for image segmentation. In particular, we cast the segmentation problem as the maximization of the mutual information between the region labels and the image pixel intensities. The resulting curve evolution equation is given in terms of nonparametric density estimates of intensity distributions, and the segmentation method can deal with a variety of intensity distributions in an unsupervised fashion.

The second component of this thesis addresses the problem of estimating shape densities from training shapes and incorporating such shape prior densities into the image segmentation process. To this end, we propose nonparametric density estimation methods in the space of curves and the space of signed distance functions. We then derive a corresponding curve evolution equation for shape-based image segmentation. Finally, we consider the case in which the shape density is estimated from training shapes that form multiple clusters. This case leads to the construction of complex, potentially multi-modal prior densities for shapes. As compared to existing methods, our shape priors can: (a) model more complex shape distributions; (b) deal with shape variability in a more principled way; and (c) represent more complex shapes.

Thesis Supervisor: Alan S. Willsky
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to express my sincere gratitude to those who helped me throughout the years at MIT. First of all, I would like to thank my thesis supervisor, Alan Willsky, for his advice, support and encouragement. I have been impressed by his enthusiasm, the breadth of his knowledge, and his deep insight. In regular meetings, he has not only shown me the big picture but also challenged me with critical questions, which often transformed my half-baked ideas into gems. I am also grateful to him for his favor of quickly and yet thoroughly reviewing each chapter of my thesis even over weekends, tremendously increasing the clarity of my work.

I am deeply indebted to Müjdat Çetin for his collaboration and advice during various stages of research and writing. Without his help, this thesis would not have been possible. He helped me to revise the draft several times to reach the desired quality. I also appreciate that he initiated a journal club, where we studied papers on shape analysis.

I am grateful to John Fisher, who introduced me to information theoretic signal processing and the use of nonparametric statistical methods. As an expert in this field, he made a significant contribution to the work presented in Chapter Three. I also thank my thesis committee member Polina Golland for carefully reading the thesis draft, pointing out relevant work in the literature, and providing constructive feedback. I also thank Anthony Yezzi for his contributions to the derivation of the curve evolution formula for the nested region integrals.

I feel grateful for all the opportunities I've had to interact with my colleagues at SSG. In particular, I would like to thank Andy Tsai for introducing me to his previous work on curve evolution. This enabled me to get a jump start in the field. I thank my curve evolution and geometry colleagues Ayres Fan and Walter Sun for all the discussions and collaborations. The journal club, with them and Müjdat Çetin, was especially helpful. I thank Alex Ihler, who has been a friendly officemate since the beginning of my life at SSG. I have enjoyed discussions with him, particularly those on nonparametric density estimation, which were especially helpful for me. I thank Erik Sudderth for answering my questions on SSG machines, Lei Chen and Dimitry Malioutov for their cheerfulness, Jason Johnson for ensuring I was not alone on late nights, and Patrick Kreidl and Jason Williams for their friendly gestures.

Finally, I would like to give my thanks to my parents, who have always loved and supported me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The objective of this thesis is in the development of advanced image segmentation methods that incorporate complex statistical information into the segmentation process, either about the image intensities or about the shapes of the objects to be segmented. To this end, we use nonparametric statistical methods for modeling both the intensity distributions and the shape distributions. By using nonparametric statistical information about the image intensities and the shapes of the objects to be segmented, we develop algorithms which can segment large classes of images and can cope with difficulties due to occlusion, severe noise, or low image contrast.

In this chapter, we first introduce the image segmentation problem. Then we discuss the main problems addressed in this thesis and contributions of this work. Finally, we provide an overview of the thesis.

■ 1.1 The Image Segmentation Problem

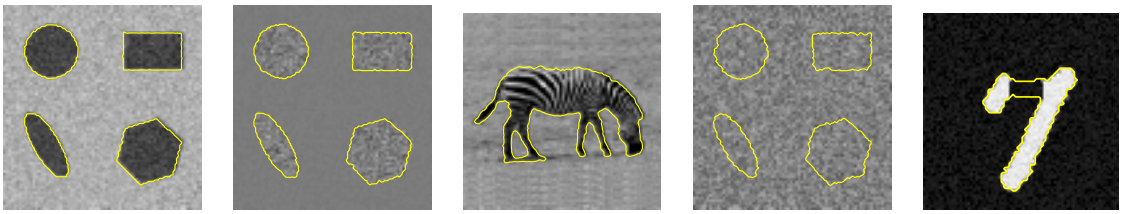
Image segmentation, the process of decomposing an image into meaningful regions, is one of the most fundamental problems in image processing and computer vision, with applications to object recognition, motion detection, medical image analysis, edge-preserving restoration of images, image magnification, and image coding. For instance, image segmentation enables selective editing of objects in image and video data, which is supported by the MPEG-4 standard [57]. In computer vision, image segmentation is an essential preprocessing step for object recognition, where we need to single out objects from the background before performing a recognition task [70]. As another example, segmentation of medical images can provide information about both the location and the anatomical structure of internal organs and parts of the human body, thereby assisting medical diagnosis, surgical planning, and therapy evaluation [40].

The current endeavors for the development of general purpose image segmentation algorithms include those based on edge detection, graph-theoretic methods, and active contour models, among others. The methods based on edge detection [7, 29, 42] start with finding edges, often including fragmented and redundant edges, and connect these edges in order to form a closed contour separating image regions. This approach, which is motivated by models of biological low-level vision, involves grouping or linking edges based on so-called Gestalt laws, which include "proximity, continuity, colinear-

ity, cocircularity, parallelism, symmetry, closure, familiarity" [70, 76]. Graph-theoretic methods [41, 59] construct a weighted graph whose nodes correspond to image pixels, and an edge between any two nodes is weighted by the likelihood that the two nodes belong to same region. Segmentation is obtained by finding a cut that minimizes the cost of the cut, which is given in terms of the sum of the weights of edges on the cut. Often the optimization process is NP-hard, and thus approximation schemes are used [59]. On the other hand, active contour models pioneered by Kass et al. [32] represent the boundary between regions as a closed contour and the contour is evolved until it converges to the boundaries of objects. Evolution of the contour usually corresponds to iterative optimization of an associated variational problem. The power of active contour models lies in: 1) that we can avoid the linking or grouping process, which is the most difficult part of methods based on edge detection; and 2) that we can optimize a cost functional in polynomial time (in the number of pixels) by evolving a curve over the image domain. Furthermore, active contour-based variational techniques became even more popularized with the emergence and use of Osher and Sethian's level set methods [48, 58]. By using level set methods, boundaries with complex geometries involving e.g. cusps and triple points, as well as topological changes in the boundaries during the curve evolution process can be handled naturally and automatically. For these reasons, active contour models based on level set methods have received considerable attention, and the focus our thesis work is also on active contour models for image segmentation.

However, most of the active contour-based image segmentation methods are often based on fairly simple statistical models for both the intensities of the regions to be segmented and the shapes of the objects in the scene. For the intensities, either simple Gaussian intensity models are assumed, or a particular discriminative feature (such as the intensity mean or variance) is used. For instance, the method proposed by Yezzi et al. [74] can segment the images in Figure 1.1(a) and Figure 1.1(b) by using the means and the variances of intensities as the statistical features, respectively, but the method requires *a priori* selection of the statistical feature to be used. Paragios et al. [51] developed a parametric model for supervised segmentation of textured images such as the image in Figure 1.1(c). This method is supervised in that it takes patches from the object region and background region and learns the statistical features of both regions (from the patches) by prior training. The information learned during the training process is then used in segmentation. The techniques in both Paragios et al. [51] and Zhu and Yuille [77] are based on parametric models, and they require that the parametric model fits the image to be segmented. Figure 1.1(d) shows an image, where the intensity distributions for the object and the background regions have the same mean and variance but the two distributions are different. Developing a segmentation method that can segment all the images in Figure 1.1(a)–(d) without *a priori* selection of discriminative statistical features and without prior training is a challenging problem.

In curve evolution methods, a penalty on the length of the segmenting curves is often used as a simple shape prior for the objects in the scene. However, in many applications, more information is available regarding the shapes of the objects, e.g. training shapes



(a) (b) (c) (d) (e)

Figure 1.1. Examples of image segmentations.

are available. The problem of incorporating shape information into segmentation using training shapes has received considerable attention, and several approaches based on Principal Component Analysis (PCA) of signed distance functions [40, 67, 68], differential geometry [12, 38], and nonlinear shape statistics [17] have been proposed, which we summarize in Section 2.3. In particular, Leventon et al. [40] represent each training shape as a signed distance function and perform a principal component analysis (PCA) on the training signed distance functions. This level-set-based method can easily represent variety of shapes with complex topology and also 3D shapes. Through the use of PCA, Leventon's method models the shape variation as a finite-dimensional Gaussian distribution. However, the approaches based on PCA of signed distance functions have the theoretical problem that the space of signed distance functions is not closed under linear operations; hence the use of linear analysis tool such as PCA gives rise to an inconsistent framework for shape modeling [68] as well as the practical problem that the Gaussian distribution is not rich enough to model the case where the training shapes form multiple clusters [17]. In order to address the latter issue and achieve more modeling capacity, Cremers et al. [17] used a nonlinear density estimate for a finite number of control points for the spline representation of the segmenting curves. However, their spline model is less powerful than the level set framework for representing shapes with complex topology and 3D shapes. Existing approaches [12, 38] based on differential geometry of curves also have the same limitation, as they can deal with only planar shapes represented by simply connected curves and can not deal with shapes with more complex topology or 3D shapes.

This thesis proposes new active contour-based image segmentation methods which can deal with general intensity distributions in an unsupervised fashion, and which: (a) can incorporate more complex statistical shape descriptions than can be captured with Gaussian models; (b) deal with shape variability in a more principled way than PCA; and (c) have more power to represent complex shapes than the approaches based on differential geometry and spline representation.

1.2 Main Problems and Contributions

1.2.1 A Nonparametric Information-Theoretic Method for Image Segmentation

The first major contribution of this thesis is the development of a nonparametric information-theoretic image segmentation method that can deal with a variety of intensity distributions. This method can address problems in which the regions to be segmented are not separable by a simple discriminative feature, or by using simple Gaussian probability densities. For instance, in order to segment the images in Figure 1.1(c) and Figure 1.1(d), one needs to use a complex discriminative feature beyond second order statistics.

In this work, we consider the mutual information (MI) between region labels and the image intensities. MI has been previously used in image registration by Viola and Wells [72, 73], where they used MI to measure the degree of agreement between two images. We use MI as a measure of goodness of the region label assignment, and we estimate region labels that maximize the MI between the region labels and the image intensities. For the class of images in which each region has a distinct intensity distribution, we show that the MI between the label and the intensity has the desirable property that it is maximized if and only if the curve is at the true boundary. Since the MI is a functional of the intensity distribution of each region partitioned by region labels, we employ nonparametric density estimates to compute the MI. The energy functional for segmentation is given in terms of the MI and the curve length, and we derive a gradient flow for curve evolution from the energy functional. The resulting curve evolution involves a log-likelihood-ratio of two nonparametric density estimates. In that sense, we use the nonparametric density estimate as the statistical feature of each region to be segmented. This feature has more modeling capacity than first and second order statistics such as means and variances.

Our strategy is different from previous curve evolution-based methods [51, 74, 77] in three major ways. First, unlike the previous techniques, our approach is based on nonparametric statistics. The performance of parametric methods can be severely affected when the assumed parametric model is not correct. This limits the class of images that can be segmented using such methods with a particular parametric model. In response to the need for robustness and a larger modeling capacity in statistical analysis, nonparametric methods [53] have been widely used in machine learning problems. Nonparametric methods estimate the underlying distributions from the data without making strong assumptions about the structures of the distributions. The nonparametric aspect of our approach makes it especially appealing when there is little or no prior information about the statistical properties of the regions to be segmented. Note that there is a trade-off, namely, with a nonparametric approach we expect some performance loss when the image fits a parametric model. However, we will give examples that clearly make the case that there are rich classes of real images for which our method is advantageous.

The second aspect of our technique is that no training is required. Again this has advantages and disadvantages. Obviously if one *has* training data from which to learn the distributions of the image regions, one should take advantage of this, as in Paragios et. al. [51]. However, it is also of practical interest to develop methods that do not require prior knowledge. We will see that the method developed here can yield results as good as those of other methods which take advantage of prior training (which our method does not, and simply must perform segmentation based on the image presented to it without *any* prior training.)

The third aspect of our technique is that this is a principled information-theoretic framework (using mutual information) that allows us to understand the several key quantities that drive the resulting curve evolution. In particular, the first such term is a likelihood ratio (LR) term that is similar to that used by Zhu et al. [77], the difference being that in [77] the LR is computed using parametric distributions whose parameters are estimated at each iteration, while ours uses distributions that are learned and dynamically adapted in a nonparametric way. If the particular parametric model is not well-matched to data, the nonparametric method will outperform the parametric counterpart. Moreover, even if the image fits a parametric model, our distribution estimates approach the quality achieved by parametric estimates. The formalism we describe also includes two additional terms which capture the sensitivity of the estimated distributions (and hence the LR) to changes in the segmenting curve as it evolves.

■ 1.2.2 Nonparametric Shape Priors

The second major contribution of this thesis is the development of an image segmentation method that utilizes not only the image intensity information but also information about the shape of the object to be segmented. Information about the shape of the object is especially useful when the image has severe noise or when some portion of the scene of interest is missing due to, e.g., occlusion. Figure 1.1(e) shows an example image of digit "7" with missing data. In this case, the image data alone is not sufficient to find the boundary of the digit 7, and we need some prior information about the shape of the object. We consider two kinds of prior information: (1) we know that the object in the scene is a "7", and (2) we only know that the scene contains a handwritten digit. We demonstrate that our framework is able to take advantage of either level of prior information. In particular, the segmentation in Figure 1.1(e) is obtained by incorporating the prior information that the image contains a digit, without the knowledge that it is a "7".

The problem we address is to obtain the information about the shape of the object to be segmented in terms of a probability density function of the shapes of the objects of the same category. In particular, we estimate such a shape probability density from available training shapes of the given category of objects. With a shape probability density, we can perform various types of statistical analysis on shapes. In particular, we can use the shape probability as a prior probability density and formulate the shape-

based segmentation problem as a maximum *a posteriori* estimation problem.

Our method is a first attempt to estimate a probability density function nonparametrically in the *space of curves* or *space of signed distance functions*. In particular, we develop a Parzen density estimator in the space of curves or the space of signed distance functions, and estimate the probability density from example curves. The key aspect of the density estimate is that evaluation of the probability density of a candidate curve is given in terms of the *distances* between the candidate curve and example curves. We demonstrate our shape density estimation approach with a number of specific distance metrics. Since the shape prior is nonparametric, it has more modeling capacity than existing approaches. In addition, our method is flexible in that it can be combined with any metric in the space of curves (or space of signed distance functions). Also it can be easily extended to modeling of 3D shapes by use of level set representation of shapes.

■ 1.3 Organization

This thesis is organized as follows.

Chapter 2: Background

This chapter provides the background and context that underlie the work described in subsequent chapters. It begins with a brief review of key pieces of work in the development of active contour models and some previous work on image segmentation. We then introduce the level set method for implementing curve evolution. We provide a review of previous work in shape analysis and shape-based segmentation. We present some background on nonparametric density estimation.

Chapter 3: A Nonparametric Information-Theoretic Method for Image Segmentation

This chapter presents our nonparametric information theoretic image segmentation method. We propose an energy functional based on the mutual information between region labels and image intensities, and derive the corresponding curve evolution equation. We then extend our method in order to segment more than two regions in the image by evolving multiple curves. We present experimental results such as unsupervised segmentation of textured images.

Chapter 4: Nonparametric Shape Priors

In this chapter, we formulate the problem of estimating shape densities from training shapes. We extend a finite dimensional nonparametric Parzen density estimator to the infinite dimensional case in order to estimate a density on the space of curves (or the space of signed distance functions). We then derive a curve evolution equation such that the curve evolves in the direction of increasing the shape prior. We show experimental results of segmenting partially-occluded images. Finally, we consider the case in which the shape density is estimated from training shapes that form multiple clusters. This case leads to the construction of complex, multi-modal prior densities for shapes.

Chapter 5: Contributions and Suggestions

We conclude by summarizing the contributions of this thesis. We also suggest some possible extensions and future research directions.

Chapter 2

Background

In this chapter, we provide background material relevant to the development of the material in later chapters. In Section 2.1, we briefly review key pieces of work in the development of active contours and discuss some related work on image segmentation. In Section 2.2, we provide preliminary information on level set methods, which are used in the implementation of active contours. In Section 2.3, we review previous work on shape analysis and shape-based segmentation, and introduce the topic of metrics in shape space, which is an essential ingredient of any shape analysis. In Section 2.4, we present background on nonparametric density estimation, which is used in this thesis for the estimation of probability densities of intensities (Chapter 3) and of shapes (Chapter 4).

■ 2.1 Curve Evolution Theory for Image Segmentation

Image segmentation is the process of partitioning an image into homogeneous regions or equivalently the process of finding the boundaries between such regions (e.g. objects in the scene). One line of work on image segmentation involves first detecting edges (which are parts of the boundary and often disconnected from one another), followed by linking the edges into a closed boundary [41]. An alternative idea is to start from closed contours and then locate the region boundaries through the evolution of such contours, leading to the so-called active contour or curve evolution methods. The segmentation algorithms we propose in this thesis are also based on the active contour models, where we evolve an active contour such that an energy functional $E(\vec{C})$ is minimized. In this section, we first provide an overview of active contour models and then provide some mathematical tools for deriving curve evolution equations from the associated energy functionals.

■ 2.1.1 Previous Work on Active Contour-Based Image Segmentation

Figure 2.1 contains a graphical depiction of some previous work based on active contour models. In the upper box we list key pieces of work in the earlier development of active contour models involving "boundary-based" models, and in the lower box we list more recent work involving "region-based" models. We should note that there has been great interest in active contours recently, and our intention is not to provide

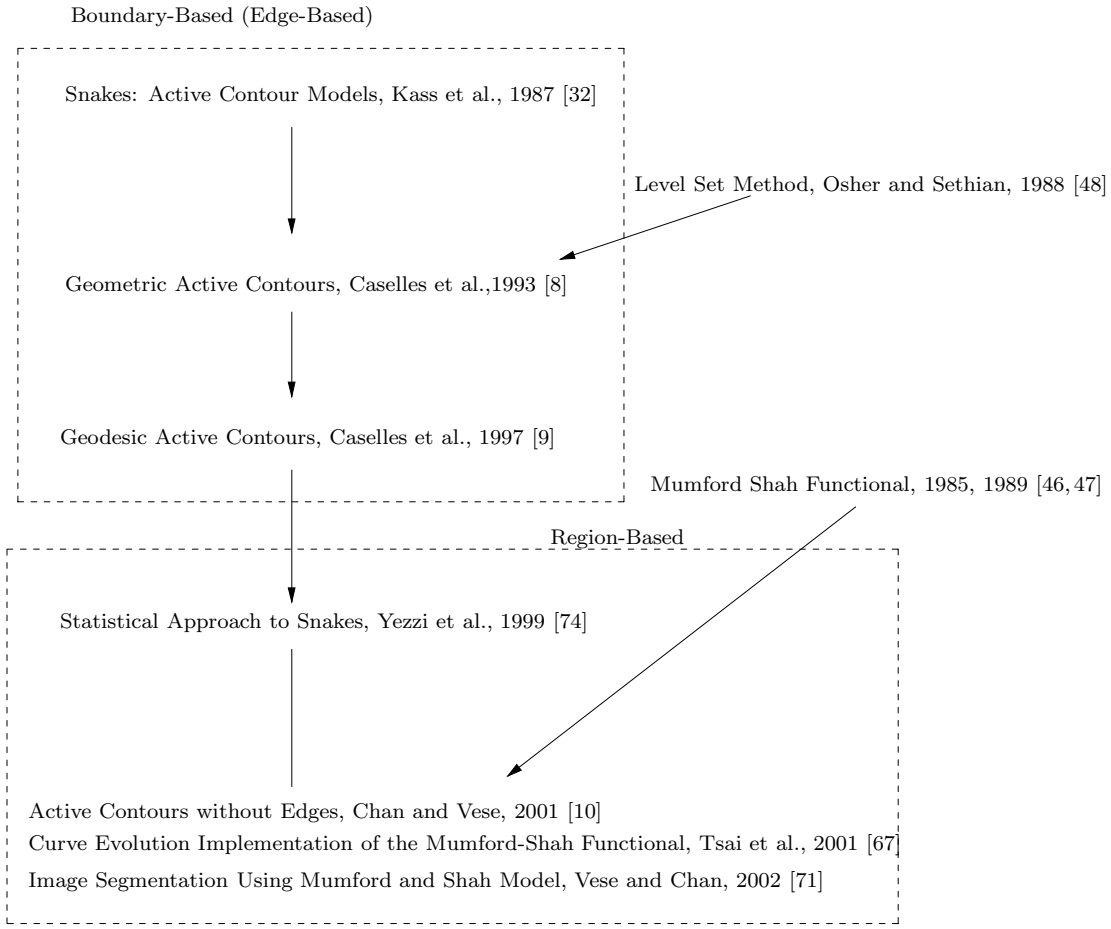


Figure 2.1. Previous work in active contour models

an extensive survey, but rather to illuminate some key pieces of work that are most relevant for the line of thought in this thesis. The common theme of all this work is that a curve is evolved such that it converges to the boundary of an object often through the minimization of an energy functional $E(\vec{C})$. In order to minimize the types of energy functionals $E(\vec{C})$ introduced later in this section, a variational analysis is used to compute the velocity field $\frac{\partial \vec{C}}{\partial t} = \vec{V}(\vec{C})$, by which we evolve the curve iteratively over time t. The velocity field $\vec{V}(\vec{C})$ is often the best deformation δC in that it maximizes

$$\lim_{h \to 0} -\frac{E(\vec{C} + h\delta\vec{C}) - E(\vec{C})}{h} \tag{2.1}$$

In this sense, such a velocity field is called a gradient flow. In boundary-based models the energy functional depends solely on the information in the local neighborhood of the active contour, whereas a region-based energy functional depends on the information obtained from the region inside or outside the curve, or from the entire image. We first introduce some of the earlier key work in active contours and then focus on some of the region-based approaches.

Overview of the Development of Active Contour Models

Active contour models were initially introduced in the seminal work of Kass et al. [32] entitled "Snakes: Active Contour Models".[1] In that work, the energy functional to be minimized is given by

$$E(\vec{C}) = \alpha \int_0^1 |\vec{C}'(p)|^2 dp + \beta \int_0^1 |\vec{C}''(p)| dp - \lambda \int_0^1 |\nabla I((\vec{C}(p))| dp, \tag{2.2}$$

where α, β, and λ are positive parameters, $\vec{C}$ is parameterized by $p \in [0, 1]$, and I is the image intensity. The first two terms (the internal energy) control the smoothness of the contour, while the third term (the external energy) attracts the contour toward edges, which are points of high intensity gradient $|\nabla I|$. The energy makes use of image information only along the boundary. In this sense, this method is boundary-based (edge-based). A drawback of this method is that it requires a good initial curve for accurate segmentation [77].

The geometric active contour model[2] proposed by Caselles et al. [8] evolves a curve by

$$\frac{\partial \vec{C}}{\partial t} = -g(I)(c + \kappa)\vec{N} \tag{2.3}$$

[1] Currently, the term "Active Contour" is often used in a broader sense to refer to a general framework based on curve evolution with a variety of energy functionals, whereas "snakes" is often used to refer to Kass's original work.

[2] They directly proposed a curve evolution equation instead of proposing an energy functional and deriving a corresponding curve evolution equation.

where κ is the curvature, $\vec{N}$ is an outward normal vector, c is chosen such that $c + \kappa$ is guaranteed to be positive, and $g(\cdot)$ is an edge indicator function. The edge indicator function used in [8] is given by

$$g(I) = \frac{1}{1 + |\nabla G_\sigma(x,y) * I(x,y)|^2} \tag{2.4}$$

where $G_\sigma * I$, a convolution of the image I with a Gaussian filter G_σ, is smoothed version of I. The edge indicator function $g(\cdot)$ is a decreasing function of the image gradient $|\nabla G_\sigma(x,y) * I(x,y)|$ and hence becomes smaller as the gradient gets larger, which is the case at the edges. The curve evolution equation (2.3) basically shrinks a curve until $g(I)$ converges to zero. In this sense, $g(I)$ behaves as a stopping function. A drawback of this approach is that for an image whose boundaries are not of high gradient, the geometric active contour evolved by (2.3) may pass through the boundary since the stopping function may not become small enough to stop the evolution.

Later, Caselles et al. proposed the geodesic active contour model [9], whose energy functional is given by

$$E(\vec{C}) = \oint_{\vec{C}} g(I(\vec{C}(s)))ds, \tag{2.5}$$

where $\vec{C}$ is parameterized by curve length s and $g(\cdot)$ is the edge indicator function in (2.4). This energy functional is obtained by weighting the Euclidean element of length ds by $g(I(C(s)))$[3]. Finding a curve minimizing this energy can be interpreted as finding a geodesic curve in a Riemannian space with a metric derived from the image content [9]. The energy is minimized by the following curve evolution

$$\frac{\partial \vec{C}}{\partial t} = -g(I)\kappa\vec{N} - (\nabla g \cdot \vec{N})\vec{N} \tag{2.6}$$

The first term is similar to the one in the geometric active contour model (2.3), but the second term is new here. This second term attracts the curve to valleys of the edge function. Thus with this model, the curve is attracted to the edges, whereas the geometric active contour in (2.3) has only a stopping term and lacks an attraction force.

All these classical active contour models are based on an edge function, which is a function of the image gradient. However, some boundaries may not be of high gradient. In that case, the curve will miss the boundary. Another problem is that the gradient is sensitive to noise. If the image is very noisy, we need significant smoothing to prevent detection of false boundaries. However, such filtering will also smooth edges making the edges less prominent and increase the probability of missing the edges. These observations have motivated the development of region-based models, which we discuss next.

[3]The energy functional itself is a measure of curve length in a Riemannian space and it still prefers a shorter curve. Hence, the energy functional does not need an additional curve length penalty term.

Region-Based Approaches

We now discuss some of the more recent approaches, which are region-based. In particular, we describe a number of methods, all of which are inspired by the so-called Mumford-Shah functional [44, 46]. Because of that, let us first briefly discuss the Mumford-Shah functional.

The original Mumford-Shah functional emerged before the active contour framework, and is given in terms of a general set of discontinuities (boundary) Γ as follows:

$$E(f,\Gamma) = \beta \int_{\Omega} (f-g)^2 dx + \alpha \int_{\Omega-\Gamma} |\nabla f|^2 dx + \gamma |\Gamma| \tag{2.7}$$

where $|\Gamma|$ stands for the total length of the arcs making up Γ. Here g is the image intensity observed, and f is a piecewise smooth approximation of the observed image. The first term says that the estimate f should be as close to the data g as possible. The second term says that the estimate f should be smooth at all points except those on the boundary. The third term says that the boundary should be as short as possible.

Tsai et al. [67] and Vese et al. [71] independently considered the Mumford-Shah functional, and wrote it in a form where the set of discontinuities Γ is given by an active contour $\vec{C}$:

$$E(f,\vec{C}) = \beta \int_{\Omega} (f-g)^2 dx + \alpha \int_{\Omega-\vec{C}} |\nabla f|^2 dx + \gamma \oint_{\vec{C}} ds. \tag{2.8}$$

They also developed a curve evolution solution for minimizing the functional in (2.8).

As a special case of the Mumford-Shah functional, Chan and Vese [10] solved the following modified Mumford Shah functional, where f is a piecewise constant approximation of the observed image g:

$$E(\vec{C}) = \int_{R} |g(x)-c_1|^2 dx + \int_{R^c} |g(x)-c_2|^2 dx + \gamma \oint_{\vec{C}} ds \tag{2.9}$$

where R is the region inside the curve, R^c is the region outside the curve, the constants c_1, c_2 are the averages of intensities g inside $\vec{C}$ and outside $\vec{C}$ respectively, thus c_1 and c_2 also depend on the curve $\vec{C}$. This corresponds to a reduced form of the Mumford-Shah -based functional in (2.8), where f is restricted to be constant (∇f=0) in each region R and R^c. Thus the observed image is approximated by a piecewise constant image $f(x) = c_1$ if $x \in R$ and $f(x) = c_2$ otherwise. This method essentially takes the mean intensity of each region as the discriminative statistical feature for segmentation.

On the other hand, Yezzi et al. [74] proposed another region-based approach, whose energy functional is given by

$$E(\vec{C}) = -\frac{1}{2}(u-v)^2 + \alpha \oint_{\vec{C}} ds, \tag{2.10}$$

where u is the mean intensity of the region inside the curve as given by $u = \frac{\int_R I(x)dx}{\int_R dx}$, and v is the mean intensity of the region outside the curve $v = \frac{\int_{R^c} I(x)dx}{\int_{R^c} dx}$. This energy

functional also takes the mean intensity of each region as the statistical feature. The curve evolution based on this energy functional basically tries to maximally separate the mean intensities of the two regions, region inside the curve and region outside the curve. Also they proposed an energy functional which is given in terms of difference of the intensity variances:

$$E(\vec{C}) = -\frac{1}{2}(\sigma_u^2 - \sigma_v^2)^2 + \alpha \oint_{\vec{C}} ds \tag{2.11}$$

where $\sigma_u^2 = \frac{\int_R (I(x)-u)^2 dx}{\int_R dx}$ denotes the sample variance inside the curve $\vec{C}$ and $\sigma_v^2 = \frac{\int_{R^c} (I(x)-v)^2 dx}{\int_{R^c} dx}$ denotes the sample variance outside the curve. With this energy functional they can segment a different class of images where the distinguishing feature is the intensity variance, for which the original Mumford Shah functional does not work. It is also possible to use other discriminative features than the mean and the variance. However, this approach (like all other approaches described in this subsection) requires *a priori* choice of the statistical feature that distinguishes each region.

We also mention Zhu and Yuille's region competition [77], which was motivated by the minimum description length (MDL) criterion of Leclerc [39]. In that work, they proposed the following energy functional:

$$E(\vec{C}, \{\alpha_i\}) = \alpha \oint_{\vec{C}} ds - \sum_{i=1}^{2} \log P(\{I_{(x,y)} : (x,y) \in R_i\} | \alpha_i) + \lambda \tag{2.12}$$

This energy functional can be interpreted as follows:

- The first term says that code length (description length) for encoding the curve is proportional to the curve length.
- The second term is the cost for coding the intensity of every pixel (x, y) inside region R_i according to a distribution $P(\{I_{(x,y)} : (x,y) \in R_i\} | \alpha_i)$.
- The third term λ is the code length needed to describe the distribution and code system (encoder) for region R_i.

They minimize the above energy functional $E(\vec{C}, \{\alpha_i\})$ by iteration of two steps: 1) with the parameters $\{\alpha_i\}$ fixed, evolve the curve $\vec{C}$; 2) with the curve $\vec{C}$ fixed, re-estimate the parameters $\{\alpha_i\}$.

■ 2.1.2 Gradient Flows for Region Integrals

Computing a gradient flow for a general energy functional $E(\vec{C})$ is a difficult task. However, in most cases the energy functional is given in the form of region integrals.

$$E(\vec{C}) = \int_R f(x) dx \tag{2.13}$$

where $f(\cdot)$ does not depend on the curve $\vec{C}$. The gradient flow that decreases this type of region integral most rapidly is derived in Appendix A and is given by

$$\frac{\partial \vec{C}}{\partial t} = -f\vec{N} \tag{2.14}$$

We refer readers to [66] for a detailed derivation. Similarly for a region integral

$$E(\vec{C}) = \int_{R^c} f(x)dx \tag{2.15}$$

The gradient flow is obtained by flipping the sign of outward normal vector $\vec{N}$,

$$\frac{\partial \vec{C}}{\partial t} = f\vec{N} \tag{2.16}$$

In the region integrals (2.13) and (2.15), the integrand does not depend on the curve. In contrast, if the integrand of a region integral depends on the curve, the gradient flow becomes more complicated, which we will see in Chapter 3.

All the curve evolution equations mentioned in this section involve motion in the normal direction. This is because a tangential component of the velocity does not have anything to do with the geometry of the evolving curve and only reparameterizes the curve. We will see this fact again in formulating the curve evolution in terms of level set functions in Section 2.2

■ 2.2 Level Set Methods

There are two approaches for the numerical implementation of a curve evolution given by $\frac{\partial \vec{C}}{\partial t}$: Lagrangian and Eulerian (fixed coordinate system). A Lagrangian approach first discretizes the boundary into segments (2D) or triangles (3D) and evolves the endpoints (marker points) of these segments or triangles. Although this sounds like a natural approach, there are a number of problems associated with this idea. First, a Lagrangian approach requires very small time steps for stable evolution of the boundary [58]. In addition, in the case of topological changes such as merging of two disconnected pieces of the boundary, it requires an ad-hoc procedure such as removal of the marker points on the portions of the boundary that have disappeared during the merging. On the contrary, the level set method, which is an Eulerian approach, can avoid the stability problem of the Lagrangian approach and can naturally handle topological changes. Level set methods become even more powerful than the Lagrangian approach when dealing with the evolution of 3D boundaries.

In this section, we provide some background on level set methods for curve evolution. In particular, we show how boundary evolution equations of the type mentioned in Section 2.1 can be turned into evolution equations for a *level set function*. We also discuss properties of such evolution equations for level set functions, and some issues on numerical implementation.

■ 2.2.1 Implicit Representation of Boundary by Level Set Function

We now introduce the concept of implicit representation of boundaries. Consider a closed curve $\vec{C}$ in $\mathbb{R}^2$ which divides the image domain Ω into three parts: the region inside the curve R, the region outside the curve R^c, and the boundary $\vec{C}$. The idea of Osher and Sethian's seminal work [48] is to define a smooth function $\phi(\mathbf{x})$ such that the set where $\phi(\mathbf{x}) = 0$ represents the boundary $\vec{C}$. Such a function ϕ is said to be a *level set function* for the boundary $\vec{C}$, if ϕ has the following property:

$$\begin{aligned} \phi(\mathbf{x}) &< 0 \text{ for } \mathbf{x} \in R \\ \phi(\mathbf{x}) &> 0 \text{ for } \mathbf{x} \in R^c \\ \phi(\mathbf{x}) &= 0 \text{ for } \mathbf{x} \in \vec{C} \end{aligned} \tag{2.17}$$

Figure 2.2 illustrates level set representation of an evolving curve.

There are many such level set functions given a boundary $\vec{C}$, but given a level set function the boundary is uniquely determined. In this sense, the level set function has redundancy in representing a boundary.

Since the gradient $\nabla\phi$ is normal to equipotential lines $\{x|\phi(x) = c\}, \forall c$, $\nabla\phi$ evaluated at a point on the zero level set $\{x|\phi(x) = 0\}$ is normal to the curve. Thus the outward unit normal to $\vec{C}$ is given by

$$\vec{N} = \frac{\nabla\phi}{|\nabla\phi|}. \tag{2.18}$$

The curvature κ of $\vec{C}$ is defined as the divergence of the normal $\vec{N}$: [4]

$$\kappa = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right). \tag{2.19}$$

The characteristic function χ of a region inside the curve, which is used in Section 3.3 and Section 4.3.2, is given by

$$\chi(x) = H(-\phi(x)) \tag{2.20}$$

where the Heaviside function $H(z)$ is

$$\begin{aligned} H(z) &\triangleq 1 \text{ if } z \geq 0 \\ H(z) &\triangleq 0 \text{ if } z < 0 \end{aligned}$$

Similarly, the characteristic function of a region outside the curve is $H(\phi(x))$.

[4] Since the computation of the normal vector and curvature involves derivatives of ϕ, one should make sure that ϕ is not noisy in numerical implementation.

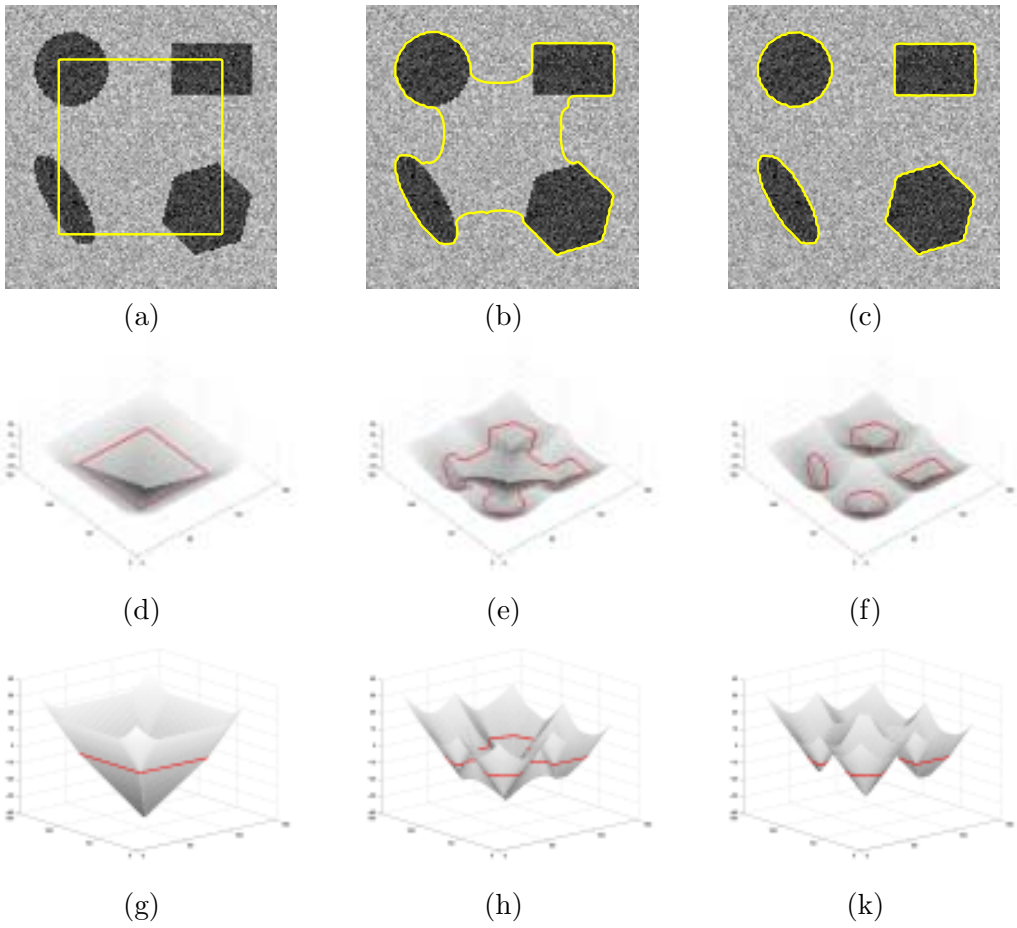


Figure 2.2. Illustration of the level set method. (a), (b), (c) show the initial, intermediate, and final stages of the curve. (d), (e), (f) show the corresponding level set function seen from high elevation. (g), (h), (k) show the corresponding level set function seen from lower elevation.

Signed Distance Function

A level set function ϕ is said to be a *signed distance function* if it not only satisfies the condition (2.17) but also its magnitude is the distance to the boundary, i.e. $|\phi(x)| = \min_{x_I \in \vec{C}} d(x, x_I)$. If ϕ is a signed distance function, it satisfies the Eikonal equation [65]

$$|\nabla \phi| = 1 \tag{2.21}$$

For signed distance functions, geometric quantities such as the outward normal vector or curvature are much simpler to compute. The outward normal vector is given by

$$\vec{N} = \nabla \phi \tag{2.22}$$

and the curvature is given by

$$\kappa = \triangle\phi \tag{2.23}$$

where $\triangle\phi$ is the Laplacian of ϕ defined as

$$\triangle\phi = \phi_{xx} + \phi_{yy} \tag{2.24}$$

Using signed distance functions not only simplifies computations of several quantities, but also makes these computations stable. When we implement an evolution of a curve by evolving the corresponding level set function, a numerical error can accumulate and the level set function can develop noisy features. For this reason, it is always advisable to reinitialize a level set function to a signed distance function occasionally during the evolution of the curve. For such a reinitialization, Tsitsiklis [69] developed a fast algorithm, the so-called fast marching method, and we refer the readers to [49, 69] for details.

■ 2.2.2 Deriving Evolution Equations for Level Set Functions

In this section, we explain how to derive the evolution equation for a level set function to implement a curve evolution. Suppose that the curve evolution is specified by a velocity field $\vec{V}(x)$, whose values are defined on every point x on the evolving curve $\vec{C}(t)$. With the velocity $\vec{V}(\cdot)$, we can describe the motion of each point on the curve $C(p,t)$ as follows:

$$\frac{\partial C(p,t)}{\partial t} = \vec{V}(C(p,t)), \forall p \in [0,1] \tag{2.25}$$

We can rewrite the above equation in vector form:

$$\frac{\partial \vec{C}}{\partial t} = \vec{V}(\vec{C}) \tag{2.26}$$

These equations (2.25) and (2.26) are the *Lagrangian* formulations of the curve evolution equation.

When the curve $\vec{C}(t)$ evolves according to (2.26), the zero level set $\{x|\phi(x,t) = 0\}$ will evolve in exactly the same way. Now we derive the update equation for the level set function. The level set function $\phi(\mathbf{x}, t)$ and the boundary $\vec{C}(t)$ are related by the following equation:

$$\phi(C(p,t),t) = 0 \text{ for all } p, t. \tag{2.27}$$

$$\phi(x,t) < 0 \text{ if } x \text{ is inside } \vec{C}(t) \tag{2.28}$$

$$\phi(x,t) > 0 \text{ if } x \text{ is outside } \vec{C}(t) \tag{2.29}$$

By differentiating (2.27) w.r.t. t, we obtain

$$\phi_t(C(p,t),t) + \nabla\phi(C(p,t),t) \cdot C_t(p,t) = 0 \tag{2.30}$$

Substituting the velocity $\vec{V}(C(p,t))$ for $C_t(p,t)$ gives us the following PDE for the update of level set function.

$$\phi_t(C(p,t),t) + \vec{V}(C(p,t)) \cdot \nabla\phi(C(p,t),t) = 0, \text{ for all } p,t \tag{2.31}$$

We can rewrite (2.31) in vector form

$$\phi_t(\vec{C}) + \vec{V}(\vec{C}) \cdot \nabla\phi(\vec{C}) = 0. \tag{2.32}$$

or simply

$$\phi_t + \vec{V} \cdot \nabla\phi = 0 \tag{2.33}$$

Since $\vec{N}$ and $\nabla\phi$ point in the same direction, $\vec{T} \cdot \nabla\phi$ is zero for any tangent vector $\vec{T}$. Hence, if we write $\vec{V} = V_n\vec{N} + V_t\vec{T}$, we have

$$\phi_t(\vec{C}) + V_n(\vec{C})\vec{N} \cdot \nabla\phi(\vec{C},t), \text{ for all } t \tag{2.34}$$

indicating that only the normal component of $\vec{V}$ is relevant for the evolution of the level set function. Since $\vec{N} = \frac{\nabla\phi}{|\nabla\phi|}$, we have

$$\phi_t + V_n|\nabla\phi| = 0 \tag{2.35}$$

Note that in (2.26) the velocity $\vec{V}$ is defined only on the boundary, and the value of the velocity off the boundary is not defined, but for (2.33) we need the velocity everywhere on the field. Hence, we need to choose the velocity value off the boundary. An important property of Equation (2.33) is that the velocity off the boundary has nothing to do with the evolution of the curve and that only the velocity value on the boundary matters.

However, in practice, we cannot assign the velocity $\vec{V}(x)$ off the boundary arbitrarily, since the way we determine the velocity off the boundary affects the performance of numerical implementation. For instance, if the velocity is assigned zero at all points off the boundary and the curve does not cross any grid point[5] the velocity on the grid points will be all zero. Hence, we will lose the necessary information for correct boundary evolution. In order to keep the velocity information, we need the velocity on the boundary to be also available on the grid points near the curve. In other words, we would like the velocity field to vary slowly as we move away from and normal to the curve. For this reason, keeping the velocity constant along the normal direction is a desirable way to determine the velocity field off the boundary.

On the other hand, if we choose to use a signed distance function, and want to constrain the evolving level set function such that it remains a signed distance function, then the velocity on the boundary is sufficient to specify the velocity at every point

[5]In numerical implementation, the set of points in Ω where the level set function $\phi(\cdot)$ is defined is called a grid. Here we use the Cartesian grid defined as $\{(x_i, y_j) = (i\Delta x, j\Delta y) | i, j \text{ are integers}\}$, where the pair $(\Delta x, \Delta y)$ denotes the grid spacing.

$x \in \Omega$. In particular, such an evolving level set function remains a signed distance function if and only if the velocity off the boundary remains constant along the normal direction to the boundary [75].

In the discussion on level set methods so far, we assumed that the value of the level set function ϕ is updated on the grid points in the image domain. We can increase the speed of the level set implementation of the curve evolution by the so-called *narrow band method* proposed by Chopp [13]. In the narrow band method, we update the values of the level set function only at the grid points in the local neighborhood of the zero level set $\{x; |\phi(x)| < c\}$. Such local neighborhood will look like a band around the zero level set, and is called a *narrow band*, where the constant c specifies the width of the narrow band. More information about the narrow band method can be found in [1, 49, 58].

■ 2.3 Shape Analysis

Shape analysis has been an important subject whose applications include object recognition, medical image analysis, and image segmentation. This section reviews some previous work in shape analysis and introduces several metrics, each of which measures a distance between two shapes.

■ 2.3.1 Previous Work

The theory of shape analysis has been well established by Kendall [33] and Small [61] when representing the shape of an object by finite number of salient points or landmarks. Cootes et al. [14] also used landmark representations and proposed 'Active Shape Models' which compute both a typical shape and typical variability from a set of training shapes via principal component analysis (PCA). However, the use of landmarks has a drawback that the performance of shape analysis depends on the quality of those landmarks. The condition for good landmarks is that there should one to one correspondence between landmarks of one training shape and those of another training shape in such a way that the corresponding landmarks are at the same location of the shape boundaries. Such choice of landmarks is difficult to automate, especially for 3D shapes, and thus landmarks are often chosen manually. For instance, in the work of Cootes et al, the landmarks were chosen and labeled manually in order to solve the correspondence problem. Later, in [19, 20], they proposed a way to automate the choice of landmarks using the minimum description length (MDL) criterion, but the process is still computationally expensive often taking several hours.

Recently, there has been increasing interest in building shape priors from training shapes, where a shape is represented as a zero level set of a signed distance function [40, 52, 68]. In [40] and [68], PCA of the signed distance functions of training data is used to capture the variability in shapes. These techniques not only provide mean shapes and principal modes of shape variation but also are useful in segmenting low SNR images or occluded images.

However, the major problem with such techniques is that the space of signed distance functions is not closed under linear operations. For instance, the direct average of distance functions, which is commonly used as a mean shape, is not a distance function. Therefore, the use of linear analysis tools such as PCA gives rise to an inconsistent framework for shape modeling [68]. As an attempt to avoid this problem, Paragios et al. [52] find a mean shape in the space of distance functions. In that work, they obtain the mean shape estimate by evolving a shape in the direction of reducing both its distance from example shapes and its distance from the space of signed distance functions. Cremers and Soatto [18] also considered level set representation of training shapes and proposed several distance measures between two signed distance functions for shape-based image segmentation.

Besides the work that involves level set methods, there has also been some other interesting work on analysis of shape. Klassen and Srivastava et al. [38] represent shapes by so-called direction functions and define the space of shapes as a sub-manifold embedded in the L_2 space of direction functions. The key element in that work is the numerical computation of a geodesic path on the shape space connecting any two different shapes, where the distance between two shapes is defined as the length of the geodesic path. However, this method can not be easily extended to deal with 3D shapes. Minchor and Mumford [43] also considered a space of curves and obtained a numerical computation of a geodesic path. Charpiat et al. [12] used an approximation of the Hausdorff metric (see next section) in order to make it differentiable and used a gradient of the approximate Hausdorff metric to warp one shape into another shape. Soatto and Yezzi [62] proposed a method of extracting both the motion and the deformation of moving deformable objects. In that work, they propose the notion of shape average and motions such that all the example shapes are obtained by rigid transformation (motion) of the shape average followed by diffeomorphism (deformation), where the shape average and motions are defined such that the total amount of deformation is minimized. In that work, the amount of such diffeomorphism is measured by a simple template metric (see next section), i.e. the area of set-symmetric difference. There is also recent work by Cootes et al. [15], which constructs a model that obeys such diffeomorphic constraint.

■ 2.3.2 Metrics for the Space of Shapes

Notion of similarity and dissimilarity between shapes is a key concept in computer vision. In order to measure such similarity and dissimilarity, several metrics for the space of shapes have been proposed. Following Mumford [45], we introduce some of the metrics below, where each shape (interior region of the shape) is represented as a subset the plane $\mathbb{R}^2$.

Hausdorff Metric

For two shapes $S_1 \subset \mathbb{R}^2$ and $S_2 \subset \mathbb{R}^2$, the Hausdorff metric is defined as follows:

$$d_H(S_1, S_2) = max \left\{ \sup_{x_1 \in S_1} \left[\inf_{x_2 \in S_2} ||x_1 - x_2|| \right], \sup_{x_2 \in S_2} \left[\inf_{x_1 \in S_1} ||x_1 - x_2|| \right] \right\} \tag{2.36}$$

Considering that for any point x_1, $\inf_{x_2 \in S_2} ||x_1 - x_2||$ is the minimum amount of dilation required in order that dilated S_2 contain the point x_1, $\sup_{x_1 \in S_1} [\inf_{x_2 \in S_2} ||x_1 - x_2||]$ can be viewed as the minimum amount of dilation required in order that S_1 is inside the dilated S_2.

Since this metric is an L^∞-type metric, the Hausdorff metric has a drawback that it is very sensitive to any outlier points in S_1 or S_2.

Template Metric

For two shapes $S_1 \subset \mathbb{R}^2$ and $S_2 \subset \mathbb{R}^2$, the template metric is defined as follows:

$$d_T(S_1, S_2) = \text{area}(S_1 - S_2) + \text{area}(S_2 - S_1) \tag{2.37}$$

where $S_1 - S_2$ is the set difference of the two regions S_1 and S_2. The template metric can be interpreted as an L_1 distance between two binary maps I_1 and I_2, whose values are 1 inside the shape and 0 outside.

$$d_T(S_1, S_2) = \int_\Omega ||I_1(x) - I_2(x)|| dx \tag{2.38}$$

Hence, it equally emphasizes all the points in each shape and is robust to outliers. However, this metric has a drawback that it is insensitive to shape difference, if the difference is of small area, e.g. when a blob grows a large but thin appendage, the template metric gives this difference little weight.

Transport Metric

For two shapes $S_1 \subset \mathbb{R}^2$ and $S_2 \subset \mathbb{R}^2$, the transport metric is defined as follows:

$$d_M(S_1, S_2) = \inf_{u \in MP} \int_{S_1} ||u(x) - x|| \mu_1(x) dx \tag{2.39}$$

where, $u : S_1 \to S_2$ has a mass preservation (MP) property. Roughly speaking, this means that we fill S_1 with mass and find the shortest paths along which to move this mass so that it now fills S_2. This is also known as the Monge-Kantorovich (MK) problem, and a survey of the MK problem can be found in [55].

Optimal Diffeomorphism

$$d_O(S_1, S_2) = \inf_{\phi} \left[\int_{S_1} ||J\phi||^2 + \int_{S_2} ||J(\phi^{-1})||^2 \right] \tag{2.40}$$

where $\phi : S_1 \to S_2$ is a 1-1, onto differentiable map with differentiable inverse ϕ^{-1} and J denotes the matrix of first derivatives.

The problem with this metric is that if S_1 and S_2 are topologically different, the distance becomes infinite. For instance, if S_1 is S_2 minus a pinhole, $d_O(S_1, S_2)$ is infinite.

■ 2.4 Nonparametric Density Estimation

Most statistical analysis tasks involve the use of probability density functions. For instance, Bayesian detection is based on the likelihood ratio, which is a ratio of two density functions. If we know the underlying densities, we can use them for the statistical analysis. However, it is often the case that we do not know these densities and thus need to estimate them. There are two classes of density estimators: parametric and nonparametric. Parametric density estimation first imposes a mathematical structure on the density, e.g., Gaussian. Then the parameters of the density are estimated, e.g. mean and covariance matrix for a Gaussian density. The computational cost for such parametric density estimator is low, but its validity depends heavily on the underlying assumption on the structure of the density. On the other hand, nonparametric density estimation does not impose a structure on the density and learns the density function from data samples drawn from the unknown density, possibly with only mild assumptions such as smoothness of the density function. The computational cost is larger than parametric approaches, but nonparametric density estimation has much larger modeling capacity than its parametric counterpart.

This thesis work makes use of nonparametric statistical methods for both image segmentation and shape analysis. In particular, we estimate densities for image intensity in Chapter 3, and we estimate densities for an infinite dimensional quantity, shape, in Chapter 4. In this section we introduce a way of nonparametrically estimating densities using the so-called Parzen density estimator and introduce other relevant machinery that will be used in later chapters.

■ 2.4.1 Parzen Density Estimator

Nonparametric density estimation was originally studied by Parzen [54], Rosenblatt [56], and Cacoullos [6], where the problem is to estimate an unknown underlying density $p(x)$ from N i.i.d. samples $x_1, \ldots, x_N$ drawn from $p(x)$.

The Parzen density estimate is a kernel-based density estimate given by

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sigma} k\left(\frac{x - x_i}{\sigma}\right) \tag{2.41}$$

where $k(\cdot)$ is said to be a kernel function, which satisfies $\int k(x)dx = 1$ and $k(\cdot) \geq 0$. The parameter σ is commonly called the kernel size, bandwidth, or smoothing parameter. In the later chapters, we only use Gaussian kernels $k(x, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/(2\sigma^2)}$. For Gaussian kernels, we let $k(x, \sigma)$ denote $\frac{1}{\sigma} k\left(\frac{x-x_i}{\sigma}\right) = N(x; 0, \sigma^2)$. Then the Parzen

density estimate with Gaussian kernel could be expressed as:

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} k(x - x_i, \sigma) \tag{2.42}$$

One can think of the Parzen density estimate as a smooth version of the histogram, which we now explain.

Histogram and Kernel Density Estimator

Let $x_1, \ldots, x_N$ be the N i.i.d. samples drawn from an unknown density $p(x)$, and consider a histogram of the data with bins $\{B_i\}$ of width w. Based on the histogram, we can construct the following density estimate $\hat{p}_{\text{hist}}$,

$$\hat{p}_{\text{hist}}(x)w = \frac{n(B_i)}{N} \tag{2.43}$$

where $n(B_i)$ is the number of samples in the ith bin. The problem with this density estimate is that $\hat{p}_{\text{hist}}$ is less accurate at the borders of the bins than at the centers of the bins.

As a remedy, we can have a smoother density estimate counting the number of samples in a moving window $B_{w/2}(x)$, which is an interval (closed ball) $[x-w/2, x+w/2]$.

$$\hat{p}_{\text{window}}(x)w = \frac{n(B_{w/2}(x))}{N} \tag{2.44}$$

where $n(B_{w/2}(x))$ is the number of samples in the window $B_{w/2}(x)$. This density estimate is equivalent to the following kernel based density estimate with a uniform kernel

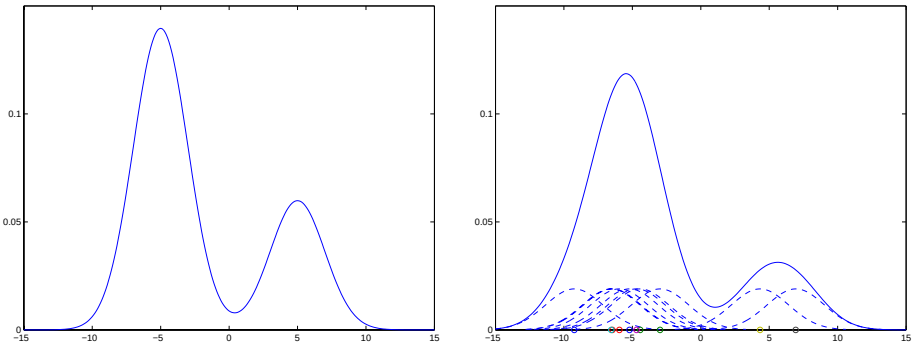
$$\hat{p}_{\text{kernel}}(x) = \frac{1}{N}\sum_{i=1}^{N} k(x - x_i) \tag{2.45}$$

where the uniform kernel is

$$k(x) = \begin{cases} \frac{1}{w}, \text{ if } x \in [-\frac{w}{2}, \frac{w}{2}] \\ 0, \text{ o.w.} \end{cases} \tag{2.46}$$

since each x_i within the interval $[x - w/2, x + w/2]$ contributes $1/(Nw)$ to $\hat{p}_{\text{kernel}}(x)$.

With a uniform kernel, all the data samples in the window $B_{w/2}(x)$ are equally weighted for estimating $p(x)$. If we use other kernels rather than the uniform one, we can put variable weights on samples x_i. Typically, we would like such weights to increase as the distance $|x - x_i|$ decreases, indicating that $k(x)$ should be chosen as a decreasing function of $|x|$.



(a) true density $p(x)$ (b) density estimate $\hat{p}(x)$

Figure 2.3. Example of density estimates: the true density shown in (a) is a mixture of Gaussians $p(x) = 0.7N(x; -5, 2^2) + 0.3N(x; 5, 2^2)$. (b) shows samples (circles), density estimates (solid line), and contribution from each kernel (dashed line).

The Parzen density estimate can also be written as a filter output of an impulse train, where the impulse train is the derivative of an empirical distribution $F_N(x) \triangleq \frac{n(\{x_i|x_i<x\})}{N}$ (where $n(\{x_i|x_i < x\})$ is the number of elements in the set $\{x_i|x_i < x\}$):

$$\hat{p}(x) = k(x) * (\frac{1}{N}\sum \delta(x - x_i)) \tag{2.47}$$

$$= k(x) * \frac{d}{dx}F_N(x) \tag{2.48}$$

An Example

As an example, Figure 2.3(a) shows a density $p(x)$ from which 10 data samples $x_1, \ldots, x_N$ are drawn. The true density $p(x)$ is a Gaussian mixture $p(x) = 0.7N(x; -5, 2^2) + 0.3N(x; 5, 2^2)$. Figure 2.3(b) shows the Parzen density estimate with a Gaussian kernel, where the dashed lines correspond to the individual terms, i.e. the $\frac{1}{N}k(x - x_i, \sigma)$'s.

■ 2.4.2 Kernel Size

By varying the shape and size of the kernel, we obtain different density estimates. For instance, a larger kernel size will produce a more spread out density estimate, and a small kernel size will make the density estimate more peaky. For an accurate estimation of the density, it is known that proper choice of the kernel size is more important than the choice of the kernel shape [60].

Asymptotically, a good kernel size is expected to decrease as the number of samples grow. In particular, Parzen [54] showed that the following conditions are necessary for asymptotic consistency of the density estimator:

$$\lim_{N\to\infty} \sigma = 0, \lim_{N\to\infty} N\sigma = \infty \tag{2.49}$$

In general, for d-dimensional random vector, it is known that $\sigma = cN^{-1/(d+4)}$ for some constant c is asymptotically optimal in density estimation [22, 60].

However, for finite N, the asymptotic results give little guidance for choosing σ. In this case, we need to use data to determine the kernel size. One possible criterion for kernel size is that of minimizing the Kullback-Leibler divergence $D(p\|\hat{p})$ [28]. Minimizing the KL divergence w.r.t. kernel size σ is equivalent to maximizing $\int p(x) \log \hat{p}(x) dx$. Since we do not have the true density p, we instead maximize an estimate of this quantity

$$\int p(x) \log \hat{p}(x) dx = E_p[\log \hat{p}(X)] \tag{2.50}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \log \hat{p}(x_i) \tag{2.51}$$

Thus the following ML kernel size with leave one out is a good choice [60]:

$$\sigma_{\mathrm{ML}} = \arg\max_{\sigma} \sum_i \log \hat{p}(x_i) \tag{2.52}$$

$$= \arg\max_{\sigma} \sum_i \log \frac{1}{N-1} \sum_{j \neq i} \frac{1}{\sigma} k(\frac{x_i - x_j}{\sigma}) \tag{2.53}$$

■ 2.4.3 Estimation of Entropy

In Chapter 3, we propose an information theoretic approach to image segmentation, which makes use of a mutual information, which in turn is given in terms of entropies. In our approach, we estimate entropies from data via nonparametric density estimation. Here we introduce techniques for entropy estimation. For a more comprehensive review for entropy estimation, we refer the reader to [5].

A differential entropy is an entropy defined for a continuous random variable. If X is a random vector taking values in $\mathbb{R}^d$ with probability density function $p(x)$, then its *differential entropy* is defined by

$$h(X) = -\int p(x) \log p(x) dx \tag{2.54}$$

$$= -E[\log p(X)] \tag{2.55}$$

For instance, if $X \sim N(0, \sigma^2)$, $h(X) = \frac{1}{2} \log 2\pi e \sigma^2$, and if $X \sim \mathrm{Unif}[0, a]$, $h(X) = \log a$.

The problem is to estimate $h(X)$ from the i.i.d. samples $X_1, \ldots, X_N$ drawn from the *unknown* pdf $p(x)$. Since the entropy is a functional of the underlying density, there are natural ways of estimating the entropy based on a nonparametric density estimate. One is given by integral form as follows:

$$h_N(X) = -\int \hat{p}(x) \log \hat{p}(x) dx, \tag{2.56}$$

where $\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} k(x - X_i, \sigma)$ is a kernel density estimator. Since this estimate requires numerical integration, the calculation becomes cumbersome for dimension $d \geq 2$ [31].

Since the entropy is given in terms of an expectation

$$h(X) = -E[\log p(X)], \tag{2.57}$$

there is another estimator avoiding the numerical integration by approximating the above expectation by a sample mean as follows:

$$h_N = -\frac{1}{N}\sum_{i=1}^{N} \log \hat{p}(X_i) \tag{2.58}$$

$$= -\frac{1}{N}\sum_{i=1}^{N} \log \frac{1}{N}\sum_{j=1}^{N} k(X_i - X_j, \sigma) \tag{2.59}$$

Ahmad and Lin [2] proposed this form of the estimator and showed mean square consistency:

$$\lim_{N\to\infty} E(h_N - h(X))^2 = 0 \tag{2.60}$$

In Chapter 3, we use this sample-mean-based entropy estimator to estimate the entropy of an intensity distribution over a certain region of an image. Since we are dealing with an image, the number of data samples is large enough for good estimation of entropy, but the issue is in computational complexity involved in density estimation. We introduce a fast way of computing Parzen density estimates next.

■ 2.4.4 Density Estimation via Fast Gauss Transform

Consider N random samples (sources) $s_1, \ldots, s_N$ drawn from $p(s)$. We have the following density estimate

$$\hat{p}(t) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(t-s_i)^2/2\sigma^2}. \tag{2.61}$$

Suppose we would like to evaluate $\hat{p}(t)$ at M different points $t_1, \ldots, t_M$. Since it takes $\mathcal{O}(N)$ time for each evaluation of $\hat{p}(t_i)$, it will take $\mathcal{O}(MN)$ time to evaluate $\hat{p}(t)$ at $t = t_1, \ldots, t_M$.

The Parzen density estimates can be obtained from a discrete Gauss transform

$$G(t) = \sum_{i=1}^{N} q_i e^{-(t-s_i)^2/\delta} \tag{2.62}$$

by substituting $\delta = 2\sigma^2$ and $q_i = \frac{1}{N\sqrt{2\pi\sigma^2}}$.

The computational cost for (2.62) can be reduced to $\mathcal{O}(c(N+M))$ by an approximation scheme known as the fast Gauss transform (FGT) [25,26,64], where c is a precision number which grows with the required precision of the approximation. The main idea involves the following decoupling of target t and source s_i in $e^{-(t-s_i)^2/\delta}$:

$$e^{-(t-s_i)^2/\delta} \approx \sum_{n=0}^{c} \frac{1}{n!} \left(\frac{s_i - s_0}{\sqrt{\delta}} \right)^n h_n \left(\frac{t - s_0}{\sqrt{\delta}} \right) \tag{2.63}$$

where s_0 is the center of $s_1, \ldots, s_N$, $h_n(t) = e^{-t^2} H_n(t)$, and $H_n(t) = (-1)^n e^{t^2} \frac{d^n}{dt^n} e^{-t^2}$ is the Hermite polynomial of order n. Using this formula, we can also approximate and factor the summation in (2.62) as follows:

$$\sum_{i=1}^{N} q_i e^{-(t-s_i)^2/\delta} \approx \sum_{n=0}^{c} \frac{1}{n!} \left[\sum_{i=1}^{N} q_i \left(\frac{s_i - s_0}{\sqrt{\delta}} \right)^n \right] h_n \left(\frac{t - s_0}{\sqrt{\delta}} \right) \tag{2.64}$$

Since computing $\left[\sum_{i=1}^{N} q_i \left(\frac{s_i - s_0}{\sqrt{\delta}} \right)^n \right]$ for all $0 \leq n \leq c$ takes $\mathcal{O}(cN)$ time, the corresponding computational cost is $\mathcal{O}(cN + cM)$ for evaluation of $\hat{p}(t) = \sum_{i=1}^{N} q_i e^{-(t-s_i)^2/\delta}$ at $t_1, \ldots, t_M$.

When each of the data vectors $s_1, \ldots, s_N$ is d dimensional, the computational cost of the FGT is $\mathcal{O}(c^d(M+N))$ [25].

The FGT is especially useful for estimating densities of image pixel intensities, where the number of pixels is often very large. We use this FGT in our nonparametric method for segmentation, which we present in Chapter 3.

Chapter 3

A Nonparametric Statistical Method for Image Segmentation

Image segmentation, the process of decomposing an image into meaningful regions, remains as one of most difficult problems in image processing and computer vision. One major challenge is in determining the defining features unique to each meaningful region or to its boundary. Such features include image brightness, color, texture, or sharpness of edges measured by gradient of brightness. Since a single feature that works well with one image may not work with another image, finding a universal feature that can be used in segmenting large classes of images is also a challenging problem.

This chapter addresses problems where the regions to be segmented are not separable by a simple discriminative feature such as a mean or a variance of image intensities of each region. To this end, we develop a nonparametric information-theoretic method for image segmentation, which can deal with a variety of intensity distributions.

In particular, we cast the segmentation problem as the maximization of the mutual information between the region labels and the image pixel intensities, subject to a constraint on the total length of the region boundaries. We assume that the probability densities associated with the image pixel intensities within each region are completely unknown a priori, and we formulate the problem based on nonparametric density estimates. Due to the nonparametric structure, our method does not require the image regions to have a particular type of probability distribution, and does not require the extraction and use of a particular statistic. We solve the information-theoretic optimization problem by deriving the associated gradient flows and applying curve evolution techniques.

This chapter is organized as follows. Section 3.1 presents the information-theoretic energy functional for two-region image segmentation. Section 3.2 contains our curve evolution-based approach to minimizing this energy functional. Section 3.3 presents an extension of the two-region version of the technique to the multi-phase segmentation problem. We then present experimental results in Section 3.4, using both synthetic images with a variety of distributions and real images. Finally, we conclude in Section 3.5 with a summary.

3.1 Information-Theoretic Cost Functional for Image Segmentation

3.1.1 Problem Statement

In this section, we consider a two-region image segmentation problem. The two regions are distinct in the sense that they have different probability density functions for the pixel intensities. We assume that the pixel intensities in each region are independent, identically distributed (i.i.d.)[1]. The associated probability density functions are unknown, and we impose no constraints on the form of these densities. More formally, the image intensity at pixel x, denoted by $G(x)$, is drawn from the density p_1 if $x \in R_1$, and from p_2 if $x \in R_2$ as follows:

$$\begin{aligned} \{G(x)|x \in R_1\} &\overset{i.i.d.}{\sim} p_1 \\ \{G(x)|x \in R_2\} &\overset{i.i.d.}{\sim} p_2, \end{aligned} \tag{3.1}$$

where R_1 and R_2 denote the two regions which are unknown, and the associated densities p_1 and p_2 are also unknown. Note that the lower case x is not a random variable but a pixel index. Later we will introduce a random variable X, which is written in a capital letter. The left-hand side of Figure 3.1 illustrates this image model. Note that a region can be composed of several topologically separate components, as shown in this figure. This image model is similar to that of the region competition method of Zhu and Yuille [77] in that both models assume that pixel intensities in each region are i.i.d. The difference is that here the distributions are unknown, whereas the model in [77] uses a family of pre-specified probability distributions.

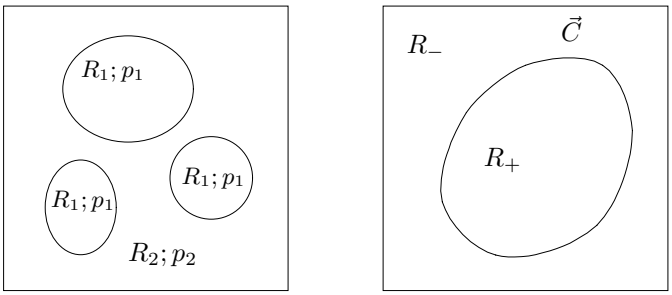


Figure 3.1. Left: Illustration of the foreground region (R_1), the background region (R_2), and the associated distributions (p_1 and p_2). Right: Illustration of the curve ($\vec{C}$), the region inside the curve (R_+), and the region outside the curve (R_-).

The goal of two-region image segmentation by curve evolution is to move a curve $\vec{C}$ such that it matches the boundary between R_1 and R_2, i.e. the region inside the curve

[1]The segmentation method we propose in this chapter can segment all the images that follow this image model. Moreover, our segmentation method can also segment non i.i.d. images such as textured images, if each region has distinct intensity distribution.

R_+ and the region outside the curve R_- converge to R_1 and R_2 respectively or vice versa. The right-hand side of Figure 3.1 illustrates two regions, R_+ and R_-. This partitioning of the image domain by the curve $\vec{C}$ gives us a binary label $L_{\vec{C}} : \Omega \to \{L_+, L_-\}$, which is a mapping from the image domain Ω to a set of two labeling symbols $\{L_+, L_-\}$ defined as follows:

$$L_{\vec{C}}(x) = \begin{cases} L_+ & \text{if } x \in R_+ \\ L_- & \text{if } x \in R_- \end{cases} \tag{3.2}$$

By this correspondence between labels and curves, image segmentation is equivalent to the binary labeling problem[2].

■ 3.1.2 Mutual Information between the Image Intensity and the Label

We now introduce the mutual information (MI) between the image intensity and the label and discuss its properties. Let us initially consider the case where p_1 and p_2 are known. As mentioned before, we have a candidate segmenting curve $\vec{C}$, and R_1, R_2 are the true unknown regions. Now suppose that we randomly choose a point X in Ω such that X is a uniformly distributed random location in the image domain[3]. In this case, the label $L_{\vec{C}}(X)$ is a binary random variable that depends on the curve $\vec{C}$. It takes the values L_+ and L_- with probability $\frac{|R_+|}{|\Omega|}$ and $\frac{|R_-|}{|\Omega|}$ respectively, where $|R_+|$ denotes the area of the region R_+.

On the other hand, the image intensity $G(X)$ is a random variable that depends on the true regions R_1 and R_2, and has the following density

$$\begin{aligned} p_{G(X)}(z) &= Pr(X \in R_1) p_{G(X)|X\in R_1}(z) + Pr(X \in R_2) p_{G(X)|X\in R_2}(z) && (3.3) \\ &= \frac{|R_1|}{|\Omega|} p_1(z) + \frac{|R_2|}{|\Omega|} p_2(z), && (3.4) \end{aligned}$$

where z is an argument for the densities. Note that this density $p_{G(X)}$ is a mixture of p_1 and p_2 due to the randomness of the pixel location X. As can be seen in (3.3), $G(X)$ has two sources of uncertainty, namely the uncertainty of pixel location being in R_1 or R_2, and the uncertainty of the intensity given the pixel location. The binary label $L_{\vec{C}}(X)$ contains some information about the former uncertainty, namely X being in R_1 or R_2. Therefore, intuitively speaking, the more accurately the label $L_{\vec{C}}(X)$ can determine whether $X \in R_1$ or $X \in R_2$, the less uncertainty $G(X)$ has, and the more information about $G(X)$ the label will have. This motivates using the mutual information $I(G(X); L_{\vec{C}}(X))$ as a segmentation criterion.

[2]In our formulation, the label is defined for each pixel, but we do not define sub-pixel level labels. Some medical applications may require sub-pixel accuracy, but it is beyond the scope of our work.

[3]This is similar to the work of Viola et al. [72], where they measure the amount of dependence between two images $u(x)$ and $v(x)$ by mutual information $I(u(X); v(X))$, where X is a random variable, which ranges over the domain of $u(\cdot)$ and $v(\cdot)$.

Now let us consider more formally the mutual information $I(G(X); L_{\vec{C}}(X))$

$$\begin{aligned} I(G(X); L_{\vec{C}}(X)) &= h(G(X)) - h(G(X)|L_{\vec{C}}(X)) \\ &= h(G(X)) - Pr(L_{\vec{C}}(X) = L_+)h(G(X)|L_{\vec{C}}(X) = L_+) \\ &\quad - Pr(L_{\vec{C}}(X) = L_-)h(G(X)|L_{\vec{C}}(X) = L_-) \end{aligned} \tag{3.5}$$

where the differential entropy $h(Z)$ of a continuous random variable Z with support S is defined by $h(Z) = -\int_S p_Z(z) \log p_Z(z) dz$. The three entropies in (3.5) are functionals of $p_{G(X)}$, $p_{G(X)|L_{\vec{C}}(X)=L_+}$, and $p_{G(X)|L_{\vec{C}}(X)=L_-}$ respectively. The two conditional distributions are given as follows:

$$\begin{aligned} p_{G(X)|L_{\vec{C}}(X)=L_+}(z) &= \sum_{i=1}^{2} Pr(X \in R_i | L_{\vec{C}}(X) = L_+) p_{G(X)|X \in R_i, L_{\vec{C}}(X)=L_+}(z) \\ &= \frac{|R_+ \cap R_1|}{|R_+|} p_1(z) + \frac{|R_+ \cap R_2|}{|R_+|} p_2(z) \end{aligned} \tag{3.6}$$

$$p_{G(X)|L_{\vec{C}}(X)=L_-}(z) = \frac{|R_- \cap R_1|}{|R_-|} p_1(z) + \frac{|R_- \cap R_2|}{|R_-|} p_2(z) \tag{3.7}$$

Each conditional entropy measures the degree of heterogeneity in each region determined by the curve $\vec{C}$. In other words, the more homogeneous the segmented regions, the smaller the conditional entropies, and the higher the mutual information is, which is a desirable property for segmentation.

We can show that the mutual information $I(G(X); L_{\vec{C}}(X))$ is maximized if and only if $\vec{C}$ is the correct segmentation, i.e. if $R_+ = R_1$, $R_- = R_2$ (or equivalently $R_+ = R_2$, $R_- = R_1$). The proof is given in Appendix B.1. This result suggests that mutual information is a reasonable criterion for the segmentation problem we have formulated.

However, in practice, we really cannot compute the mutual information $I(G(X); L_{\vec{C}}(X))$ for two reasons. First, the computations above involve the regions R_1 and R_2, which are unknown to us (otherwise the segmentation problem would be solved). Second, unlike what we assumed in the above discussion, we would like to solve the segmentation problem when p_1 and p_2 are unknown.

We thus need to estimate the mutual information as follows:

$$\begin{aligned} \hat{I}(G(X); L_{\vec{C}}(X)) &= \hat{h}(G(X)) - Pr(L_{\vec{C}}(X) = L_+)\hat{h}(G(X)|L_{\vec{C}}(X) = L_+) \\ &\quad - Pr(L_{\vec{C}}(X) = L_-)\hat{h}(G(X)|L_{\vec{C}}(X) = L_-) \end{aligned} \tag{3.8}$$

This in turn requires us to estimate the densities $p_{G(X)}$, $p_{G(X)|L_{\vec{C}}(X)=L_+}$, and $p_{G(X)|L_{\vec{C}}(X)=L_-}$. The way we estimate these densities is presented in Section 3.2.

■ 3.1.3 The Energy Functional

Finally, we combine the mutual information estimate with the typical regularization penalizing the length of the curve in order to construct our overall energy functional

to be used for segmentation. This regularization prevents the formation of a longer jagged boundary. Depending on the prior information one might have about the region boundaries, constraints other than the curve length penalty can also be used in our framework, which is the main topic of Chapter 4.

In the energy functional, the mutual information is weighted by the area of the image domain in order to represent the total amount of information between the label and the image, since $I(G(X); L_{\vec{C}}(X))$ corresponds to the contribution of a single pixel to the total information[4]. The resulting energy functional to minimize is then given by

$$E(\vec{C}) \;=\; -|\Omega|\hat{I}(G(X); L_{\vec{C}}(X)) + \alpha \oint_{\vec{C}} ds, \tag{3.9}$$

where $\oint_{\vec{C}} ds$ is the length of the curve and α is a scalar parameter. The statistical interpretation of this energy functional is given in the next section.

■ 3.1.4 MAP Estimation Interpretation of the Energy Functional

The curve that minimizes the energy functional is given by

$$\arg\min_{\vec{C}} E(\vec{C}) \;=\; \arg\min_{\vec{C}} |\Omega|\hat{h}(G(X)|L_{\vec{C}}(X)) + \alpha \oint_{\vec{C}} ds. \tag{3.10}$$

Now the conditional entropy term corresponds to the negative logarithm of the likelihood as follows:

$$\begin{aligned}
|\Omega|\hat{h}(G(X)|L_{\vec{C}}(X)) &= |\Omega|Pr(L_{\vec{C}}(X) = L_+)\hat{h}(G(X)|L_{\vec{C}}(X) = L_+) \\
&\quad + |\Omega|Pr(L_{\vec{C}}(X) = L_-)\hat{h}(G(X)|L_{\vec{C}}(X) = L_-) \\
&= -|\Omega|\frac{|R_+|}{|\Omega|}\frac{1}{|R_+|}\int_{R_+} \log \hat{p}_+(G(x))dx \\
&\quad - |\Omega|\frac{|R_-|}{|\Omega|}\frac{1}{|R_-|}\int_{R_-} \log \hat{p}_-(G(x))dx \\
&= -\int_{\Omega} \log \hat{p}_{G(X)|L_{\vec{C}}(X)=L_{\vec{C}}(x)}(G(x)|L_{\vec{C}}(X) = L_{\vec{C}}(x))dx,
\end{aligned} \tag{3.11}$$

where the last expression is the negative log-likelihood of the data $\{G(x)|x \in \Omega\}$ in terms of the estimated density. On the other hand, the curve length term can be interpreted as the negative logarithm of prior probability for the curve, $\oint_{\vec{C}} ds = -\log p(\vec{C})$. Therefore, minimizing the energy functional corresponds to finding the maximum a posteriori estimate of the label.

[4]Considering that we have a regularization parameter α to choose, weighting the mutual information by the area of the image domain does not change the performance of the algorithm. It is only of theoretical interest.

■ 3.2 Nonparametric Density Estimation and Gradient Flows

This section contains the derivation of the curve evolution formula for minimizing the energy functional $E(\vec{C})$ of (3.9) using nonparametric Parzen density estimates. First, we present the way the nonparametric Parzen density estimates are used in estimating the conditional entropy terms in (3.8). This results in the expression of the energy functional $E(\vec{C})$ in the form of nested region integrals. We then calculate the gradient flow for $E(\vec{C})$ and discuss the properties of the curve evolution formula.

■ 3.2.1 Estimation of the Differential Entropy

The expression (3.8) involves differential entropy estimates, and we use nonparametric Parzen density estimates in order to estimate the differential entropies. A brief introduction to nonparametric entropy estimation is in Section 2.4.3.

Since $\hat{h}(G(X))$ in (3.8) is independent of the curve, we just consider $\hat{h}(G(X)|L_{\vec{C}}(X) = L_+)$ and $\hat{h}(G(X)|L_{\vec{C}}(X) = L_-)$ as follows:

$$\begin{aligned}
&\hat{h}(G(X)|L_{\vec{C}}(X) = L_+) \\
&= -\frac{1}{|R_+|}\int_{R_+} \log\, \hat{p}_+(G(\mathbf{x}))\, d\mathbf{x} && (3.12)\\
&= -\frac{1}{|R_+|}\int_{R_+} \log\left(\frac{1}{|R_+|}\int_{R_+} K(G(\mathbf{x}) - G(\hat{\mathbf{x}}))d\hat{\mathbf{x}}\right) d\mathbf{x}, && (3.13)
\end{aligned}$$

Note that $h(G(X)|L_{\vec{C}}(X) = L_+)$ involves the expected value of the logarithm of $p_+ \triangleq p_{G(X)|L_{\vec{C}}(X)=L_+}$, and we approximate this expected value by the sample mean of $\log p_+$ in (3.12). We then use a continuous version of the Parzen density estimate [53] of p_+ in (3.13). We use the kernel $K(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{z^2}{2\sigma^2}}$, where σ is a scalar parameter. Similarly, we have:

$$\begin{aligned}
&\hat{h}(G(X)|L_{\vec{C}}(X) = L_-) \\
&= -\frac{1}{|R_-|}\int_{R_-} \log\left(\frac{1}{|R_-|}\int_{R_-} K(G(\mathbf{x}) - G(\hat{\mathbf{x}}))d\hat{\mathbf{x}}\right) d\mathbf{x} && (3.14)
\end{aligned}$$

■ 3.2.2 Gradient Flows for General Nested Region Integrals

Note that (3.13) and (3.14) have nested region integrals. Let us consider a general nested region integral of the form

$$\int_R f(\varepsilon(\mathbf{x}, t))\, d\mathbf{x} \quad \text{where} \quad \varepsilon(\mathbf{x}, t) = \int_R g(\mathbf{x}, \hat{\mathbf{x}})\, d\hat{\mathbf{x}}, \tag{3.15}$$

where $g(\cdot, \cdot)$ does not depend on $\vec{C}$ given its arguments, R is the region inside the curve $\vec{C}$, and t is a time index for the evolution of $\vec{C}$ (which we often drop for notational convenience as in $R = R(t)$ and $\vec{C} = \vec{C}(t)$). Note that $\varepsilon(\mathbf{x}, t)$, which is a region

integral, depends on $\vec{C}$ for an arbitrary fixed value of the argument $\mathbf{x}$, whereas $g(\mathbf{x}, \hat{\mathbf{x}})$ is a function of only $\mathbf{x}$ and $\hat{\mathbf{x}}$ and does not depend on $\vec{C}$ given $\mathbf{x}$ and $\hat{\mathbf{x}}$. For nested region integrals of the form (3.15), we have derived the gradient flow (the negative of the gradient so that the region integral decreases most rapidly), which is given by

$$\frac{\partial \vec{C}}{\partial t} = -\left[f\big(\varepsilon(\vec{C})\big) + \int_R f'\big(\varepsilon(\mathbf{x})\big) g(\mathbf{x}, \vec{C})\, d\mathbf{x} \right] \vec{N}, \tag{3.16}$$

where $\vec{N}$ is the outward unit normal vector. The detailed derivation can be found in Appendix B.3. The above equation is a shorthand way of saying that for every point $C(p)$ parameterized by p on the curve $\vec{C}$, we move the point $C(p)$ by

$$\frac{\partial C(p)}{\partial t} = -\left[f\big(\varepsilon(C(p))\big) + \int_R f'\big(\varepsilon(\mathbf{x})\big) g(\mathbf{x}, C(p))\, d\mathbf{x} \right] \vec{N}, \tag{3.17}$$

The second term appears in (3.16) because the integrand $f(\varepsilon(\mathbf{x}, t))$ in (3.15) depends on the curve $\vec{C}$ as $\varepsilon(\mathbf{x}, t)$ depends on the curve.

■ 3.2.3 The Gradient Flow for the Information-Theoretic Energy Functional

Now that we have the nonparametric estimates of the mutual information in the form of nested region integrals as in (3.13) and (3.14), it is straightforward to calculate the gradient flow for the energy functional $E(\vec{C})$ using the result of Section 3.2.2. We provide the details of this computation in Appendix B.4. Here, we state the main result, namely the overall gradient flow for $E(\vec{C})$ of (3.9):

$$\begin{aligned} \frac{\partial \vec{C}}{\partial t} \;=\; & \left[\log \frac{\hat{p}_+(G(\vec{C}))}{\hat{p}_-(G(\vec{C}))} + \frac{1}{|R_+|} \int_{R_+} \frac{K\big(G(\mathbf{x}) - G(\vec{C})\big)}{\hat{p}_+(G(\mathbf{x}))}\, d\mathbf{x} \right. \\ & \left. - \frac{1}{|R_-|} \int_{R_-} \frac{K\big(G(\mathbf{x}) - G(\vec{C})\big)}{\hat{p}_-(G(\mathbf{x}))}\, d\mathbf{x} \right] \vec{N} - \alpha\kappa\vec{N}, \end{aligned} \tag{3.18}$$

where κ is the curvature of the curve and $-\alpha\kappa\vec{N}$ is the gradient flow for the curve length penalty, whose derivation can be found in [24]. We implement the curve evolution for the gradient flow in (3.18) using the level set method [48, 58] together with the narrow band approach mentioned in Section 2.2.2.

A direct computation of this gradient flow is expensive. In particular, the bottleneck is in the computation of the second and the third terms. If we use a direct computation, it takes $\mathcal{O}((\#\text{ of pixels})^2)$ time per each iteration, which we now explain. Since the evaluation of the density estimate in the form of $\hat{p}_+(G(x)) = \frac{1}{N}\sum_{i=1}^{N} K(G(x) - G(x_i))$ at each pixel x on the curve takes $\mathcal{O}(N)$ time, evaluation of $\hat{p}_+(G(\vec{C}))$ at each pixel on the curve takes $\mathcal{O}(|R_+|)$ time, where $|R+|$ is the number of pixels in region inside the curve. Thus the computation of the first term at all the points on the curve takes

$\mathcal{O}(M(|R_+| + |R_-|))$ time, where M is the number of pixels along the curve (i.e. the size of the narrow band). In order to compute the second term, we compute and store $\hat{p}_+(G(x))$ for all $x \in R_+$, which takes $\mathcal{O}(|R_+|^2)$ time and then compute the integral using the stored values of $\hat{p}_+(G(x))$. The computation of this integral at all the points on the curve takes $\mathcal{O}(M|R_+|)$ time. Therefore, the complexity of a direct computation of the gradient flow is $\mathcal{O}(M(|R_+|+|R_-|)+|R_+|^2+M|R_+|+|R_-|^2+M|R_-|) \sim \mathcal{O}((\#\text{ of pixels})^2)$ per each step.

However, we reduce the complexity by using an approximation method based on the fast Gauss transform (FGT) [25, 26, 64] as we have mentioned in Section 2.4.4. FGT can evaluate density estimates based on N data points in the form of $\hat{P}(x) = \frac{1}{N}\sum_{i=1}^{N} K(x - x_i)$ at M different points in $\mathcal{O}(c(M + N))$ time instead of $\mathcal{O}(MN)$ time, where c is the precision number which grows with the required precision of the approximation. The precision number c is the order of the Taylor series expansions used in FGT, and c less than 10 is often sufficient in most cases. Furthermore, in evaluating $\hat{p}_+$, we observe that using only a randomly selected subset of R_+ is sufficient instead of using all the pixel intensities in R_+. If we select N points from R_+ in order to estimate $\hat{p}_+$ and another N points from R_-, the computational cost using FGT per each iteration is $\mathcal{O}(c(M+N+N)+c(N+N)+c(M+N)+c(N+N)+c(M+N))$, where the integral in the second and third term in (3.18) takes $\mathcal{O}(c(M + N))$ time by FGT. Given the size of the narrow band, a reasonable choice of N will be a linear function of M. This results in the overall complexity of $\mathcal{O}(M)$, i.e. linear in the size of the narrow band.

In general, FGT is also possible for estimation of multi-dimensional density functions, which will allow us to extend our framework to color and vector-valued images. For d dimensional data, the complexity of FGT is now $\mathcal{O}(c^d(M + N))$ [25], with the same M and N as the above. The only difference in computational complexity from the case of gray level images is in the constant factor c^d. Therefore, the computational complexity is still linear in the size of the narrow band, if our method is extended to vector-valued images.

■ 3.2.4 Discussion on the Gradient Flow

The first term of the gradient flow expression in (3.18) is a log-likelihood ratio which compares the hypotheses that the observed image intensity $G(\vec{C})$ at a given point on the active contour $\vec{C}$ belongs to the foreground region R_+ or the background region R_- based upon the current estimates of the distributions p_+ and p_-. This log-likelihood ratio term favors the movement of the curve in the direction to make the updated regions more homogeneous.

To understand the second and third terms in (3.18), let us consider the analogy to the generic flow in (3.16). We have the second term of (3.16) because the integrand $\varepsilon(\cdot)$ in (3.15) depends on the curve. Similarly, we have the second and third terms in the gradient flow (3.18) because the integrands of the entropy estimates (3.13) and (3.14), which are logarithms of Parzen density estimates, depend on the curve.

These second and third terms reinforce and refine what the first term does. The

first term alone does not take into account the fact that a deformation of the curve results in updating the data samples used for the two density estimates. It is the two additional terms that compensate for the change of density estimates.

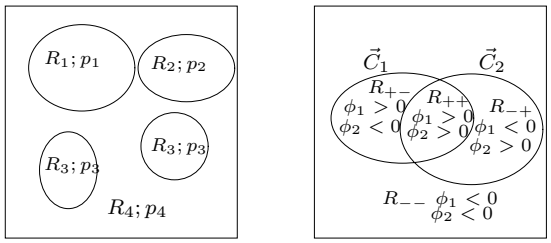
These second and third terms, as well as the use of the nonparametric density estimates distinguish this active contour model from the region competition algorithm of Zhu and Yuille [77], which involves alternating iterations of two operations: estimating the distribution parameters inside and outside the curve; and likelihood ratio tests to evolve the curve. In that algorithm, changes in the distributions are not directly coupled with likelihood ratio tests. In contrast, the changes in the nonparametric density estimates are built directly into our curve evolution equation through these two terms.

■ 3.3 Extension to Multi-phase Segmentation

In this section, we provide an extension of the two-region version of our technique to images with more than two regions. To this end, we incorporate the multi-phase segmentation formulation of [11] into our information-theoretic, nonparametric segmentation framework. Our method uses m level set functions to segment up to 2^m regions, and the resulting curve evolution equation (motion equation) turns out to be a natural generalization of nonparametric region competition.

■ 3.3.1 n-ary Segmentation Problem and Mutual Information

We extend the two-region image segmentation problem to an n-ary (i.e. n-region) version, where $R_1, \ldots, R_n$ denote the true unknown regions, and the image intensity at pixel x, denoted by $G(x)$, is drawn from the density p_i if $x \in R_i$, where p_i's are unknown. Figure 3.2(a) illustrates this image model when $n = 4$.



(a) (b)

Figure 3.2. Multi-phase segmentation image model. (a) Illustration of the case where $n = 4$: true regions $R_1, \ldots, R_4$, with the associated distributions $p_1, \ldots, p_4$. (b) Illustration of the two curves $(\vec{C}_1, \vec{C}_2)$ and the regions $R_{++}, R_{+-}, R_{-+}, R_{--}$ partitioned by the curves .

The goal of n-ary image segmentation by curve evolution is to move a set of curves $\{\vec{C}_1, \ldots, \vec{C}_m\}$ (equivalently, a set of level set functions $\{\phi_1, \ldots, \phi_m\}$) such that these curves partition the image domain into the true regions $R_1, \ldots, R_n$. Each curve C_i

partitions the image domain into the two regions, the region inside the curve and the region outside the curve (ϕ_i does the same thing by its sign). Thus the m level set functions partition the image domain into 2^m regions, each of which we label by the signs of the level set functions in that region. For instance, when $m = 2$, we have 4 regions, $R_{++}, R_{+-}, R_{-+}, R_{--}$ as illustrated in Figure 3.2(b).

Given the partitioning by the curves $\mathbf{C} \triangleq \{\vec{C}_i\}_{i=1}^m$, we can label each pixel x by its label $L_{\mathbf{C}}(x)$. For instance, if $x \in R_{++}$, $L_{\mathbf{C}}(x) = L_{++}$. More formally, this partitioning of the image domain by the curves $\mathbf{C}$ gives us a label

$$L_{\mathbf{C}} : \Omega \to \{L_{++\cdots+}, \ldots, L_{--\cdots-}\},$$

which is a mapping from the image domain Ω to a set of 2^m labeling symbols $\{L_{++\cdots+}, \ldots, L_{--\cdots-}\}$ defined as follows:

$$L_{\mathbf{C}}(x) = L_{s(i)} \text{ if } x \in R_{s(i)}, 1 \leq i \leq 2^m, \tag{3.19}$$

where $s(i)$ is the ith element in the set $\{+ + \cdots +, \ldots, - - \cdots -\}$. By a straightforward generalization of (3.9), we propose the following energy functional for multi-phase segmentation:

$$E(\mathbf{C}) \;=\; -|\Omega|\hat{I}(G(X); L_{\mathbf{C}}(X)) + \alpha \sum_{i=1}^{m} \oint_{\vec{C}_i} ds, \tag{3.20}$$

where the mutual information estimate is naturally extended to:

$$\hat{I}(G(X); L_{\mathbf{C}}(X)) \;= \hat{h}(G(X)) - \sum_{i=1}^{2^m} Pr(L_{\mathbf{C}}(X) = L_{s(i)}) \hat{h}(G(X)|L_{\mathbf{C}}(X) = L_{s(i)}) \tag{3.21}$$

■ 3.3.2 The Gradient Flows

We now compute the gradient flow to minimize $E(\mathbf{C})$ of (3.20). For notational convenience, we consider the case where $m = 2$, but the development could easily be generalized to any m.

In (3.21), we have $2^m = 4$ conditional entropies to estimate, namely, $\hat{h}(G(X)|L_{\mathbf{C}}(X) = L_{++}), \ldots, \hat{h}(G(X)|L_{\mathbf{C}}(X) = L_{--})$. We compute these estimates in a way that is analogous to what we did for the two-region case. For example, $\hat{h}(G(X)|L_{\mathbf{C}}(X) = L_{++})$ is given by

$$\begin{aligned} \hat{h}(G(X)|L_{\mathbf{C}}(X) = L_{++}) \;&=\; -\frac{1}{|R_{++}|} \int_{R_{++}} \log \hat{p}_{++}(G(\mathbf{x}))\, d\mathbf{x} \qquad (3.22) \\ &=\; -\frac{1}{|R_{++}|} \int_{R_{++}} \log \left(\frac{1}{|R_{++}|} \int_{R_{++}} K(G(\mathbf{x}) - G(\hat{\mathbf{x}})) d\hat{\mathbf{x}} \right) d\mathbf{x}, \end{aligned}$$

and the other entropy estimates are obtained in a similar way.

Generalizing our results from Section 3.2, and using the multi-phase segmentation formulation of [11], we compute the first variation of the energy functional $E(\mathbf{C})$ in (3.20), and obtain the following coupled motion equations:

$$\begin{aligned}\frac{\partial \vec{C}_1}{\partial t} &= \vec{N}_1 \Bigg[-\alpha\kappa_1 + H(\phi_2(\vec{C}_1)) \left(\log \frac{\hat{p}_{++}(G(\vec{C}_1))}{\hat{p}_{-+}(G(\vec{C}_1))} + \frac{1}{|R_{++}|} \int_{R_{++}} \frac{K(G(\mathbf{x}) - G(\vec{C}_1))}{\hat{p}_{++}(G(\mathbf{x}))} d\mathbf{x} \right. \\ &\quad \left. - \frac{1}{|R_{-+}|} \int_{R_{-+}} \frac{K(G(\mathbf{x}) - G(\vec{C}_1))}{\hat{p}_{-+}(G(\mathbf{x}))} d\mathbf{x} \right) + (1 - H(\phi_2(\vec{C}_1)) \left(\log \frac{\hat{p}_{+-}(G(\vec{C}_1))}{\hat{p}_{--}(G(\vec{C}_1))} \right. \\ &\quad \left. + \frac{1}{|R_{+-}|} \int_{R_{+-}} \frac{K(G(\mathbf{x}) - G(\vec{C}_1))}{\hat{p}_{+-}(G(\mathbf{x}))} d\mathbf{x} - \frac{1}{|R_{--}|} \int_{R_{--}} \frac{K(G(\mathbf{x}) - G(\vec{C}_1))}{\hat{p}_{--}(G(\mathbf{x}))} d\mathbf{x} \right) \Bigg] \end{aligned} \tag{3.23}$$

$$\begin{aligned}\frac{\partial \vec{C}_2}{\partial t} &= \vec{N}_2 \Bigg[-\alpha\kappa_2 + H(\phi_1(\vec{C}_2)) \left(\log \frac{\hat{p}_{++}(G(\vec{C}_2))}{\hat{p}_{+-}(G(\vec{C}_2))} + \frac{1}{|R_{++}|} \int_{R_{++}} \frac{K(G(\mathbf{x}) - G(\vec{C}_2))}{\hat{p}_{++}(G(\mathbf{x}))} d\mathbf{x} \right. \\ &\quad \left. - \frac{1}{|R_{+-}|} \int_{R_{+-}} \frac{K(G(\mathbf{x}) - G(\vec{C}_2))}{\hat{p}_{+-}(G(\mathbf{x}))} d\mathbf{x} \right) + (1 - H(\phi_1(\vec{C}_2)) \left(\log \frac{\hat{p}_{-+}(G(\vec{C}_2))}{\hat{p}_{--}(G(\vec{C}_2))} \right. \\ &\quad \left. + \frac{1}{|R_{-+}|} \int_{R_{-+}} \frac{K(G(\mathbf{x}) - G(\vec{C}_2))}{\hat{p}_{-+}(G(\mathbf{x}))} d\mathbf{x} - \frac{1}{|R_{--}|} \int_{R_{--}} \frac{K(G(\mathbf{x}) - G(\vec{C}_2))}{\hat{p}_{--}(G(\mathbf{x}))} d\mathbf{x} \right) \Bigg] \end{aligned} \tag{3.24}$$

where $H(\cdot)$ is the Heaviside function ($H(\phi)$=1 if $\phi \geq 0$ and $H(\phi) = 0$ if $\phi < 0$).

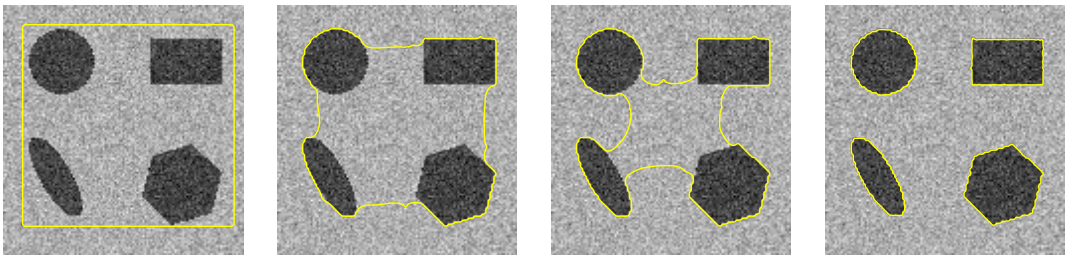
Equations (3.23), (3.24) involve log-likelihood ratio tests comparing the hypotheses that the observed image intensity $G(\vec{C}_i)$ at a given point on the active contour $\vec{C}_i$ belongs to one region or the other.

As illustrated in Figure 3.2(b), $\vec{C}_1$ delineates either the boundary between R_{++} and R_{-+}, or the boundary between R_{+-} and R_{--}, when $\vec{C}_1$ lies inside or outside curve $\vec{C}_2$, respectively. Equation (3.23) exactly reflects this situation and reveals the region competition between regions adjacent to curve $\vec{C}_1$. Similarly, Equation (3.24) expresses the region competition between regions adjacent to curve $\vec{C}_2$.

■ 3.4 Experimental Results

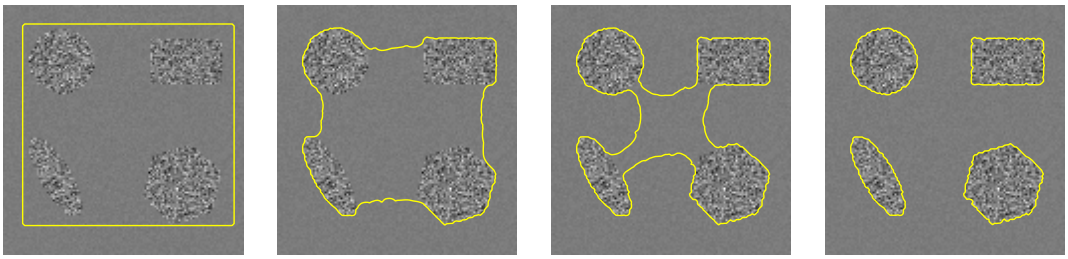
We present experimental results on synthetic images of geometric objects, and a number of real images. In all the examples, the regularization parameter α in (3.9) or (3.20) is chosen subjectively based upon our qualitative assessment of the segmented imagery. In cases where prior information is available about the objects in the scene, it may be possible to learn an appropriate distribution of regularizers based upon the known smoothness characteristics of the object boundaries coupled with the signal-to-noise ratios of the images to be segmented.

We use synthetic images generated by several sets of distributions. Figure 3.3



(a) initial (b) intermediate (c) intermediate (d) final

Figure 3.3. Evolution of the curve on a synthetic image; the different mean case.



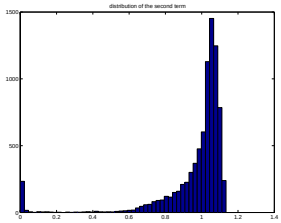
(a) initial (b) intermediate (c) intermediate (d) final

Figure 3.4. Evolution of the curve on a synthetic image; the different variance case.

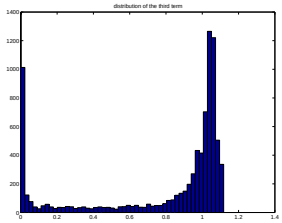
shows the result produced by our technique for the case where the two distributions for the foreground and the background are Gaussian with different means and the same variance. Figure 3.4 shows the result for the case where the two distributions for the foreground and the background are Gaussian with different variances and the same mean. For these two cases, the method of Yezzi et al. [74] would require the selection of the appropriate statistic (i.e. mean and variance for the first and second cases respectively) *a priori*, whereas our method solves the segmentation problem without that information.

For the result in Figure 3.3, we measured the run time for both our nonparametric method and parametric counterpart in [74]. On an Intel Xeon 2.2 GHz cpu, the nonparametric method took 167 seconds (image size is 126 by 121), whereas the parametric method took 26 seconds. The parametric method is of less computational cost. However, if there is a mismatch between the image and the parametric model, there will be losses in the accuracy of the segmentation.

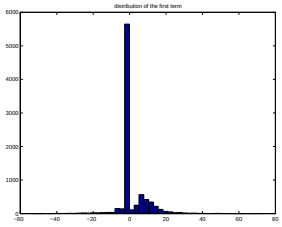
As we mentioned in Section 3.2.4, the motion equation for the curve (3.18) contains three data-driven terms and a curvature term. We now provide an empirical analysis of the relative contribution of the first data-driven term (the log-likelihood ratio) versus the other two data-driven terms, to the overall curve evolution. To this end, we consider the example in Figure 3.3. We compute the numerical values of the log-likelihood ratio, the



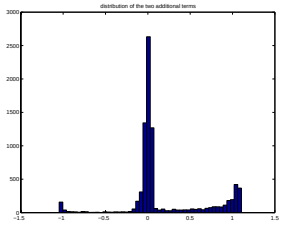
(a) histogram of the second term



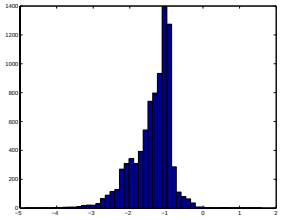
(b) histogram of the third term



(c) histogram of the first term



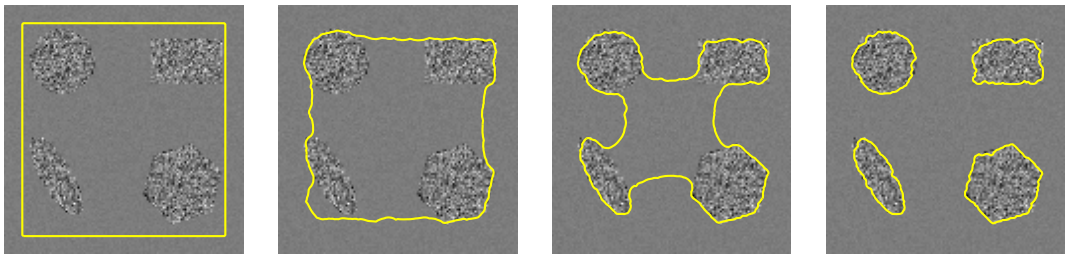
(d) histogram of (2nd term - 3rd term)



(e) histogram of $\log_{10} \frac{|\text{2nd term - 3rd term}|}{|\text{1st term}|}$

Figure 3.5. Histograms of the three terms of the gradient flow for the points on the boundaries of Figure 3.3.

second term, and the third term of the gradient flow (3.18) at each point on the curve, for multiple snapshots during the iterative curve evolution process. In order to analyze the general behavior of these terms, we combine all those data obtained throughout the curve evolution process and show their histograms in Figure 3.5. Figure 3.5(a) and Figure 3.5(b) show histograms of the values taken by the second term and the third term respectively. We observe that the values of both terms are often close to 1, and lie in a limited range (mostly between 0 and 1.5). We analyze this observation in more detail in Appendix B.5. Figure 3.5(c) and Figure 3.5(d) show histograms of the values taken by the first term and the other two terms (i.e. the second term minus the third term). Since both the second and the third term have a limited range, their difference (which is their overall contribution to the evolution) is also in a limited range (mostly between -1.5 and 1.5), as is shown in Figure 3.5(d). Finally, Figure 3.5(e) shows a histogram of $\log_{10} \frac{|\text{2nd term - 3rd term}|}{|\text{1st term}|}$. We can observe that the first term mostly has a larger magnitude than the other two terms; hence it is the dominant contributor to the curve evolution. Consequently, for the experiment in Figure 3.3, we obtain a similar



(a) initial (b) intermediate (c) intermediate (d) final

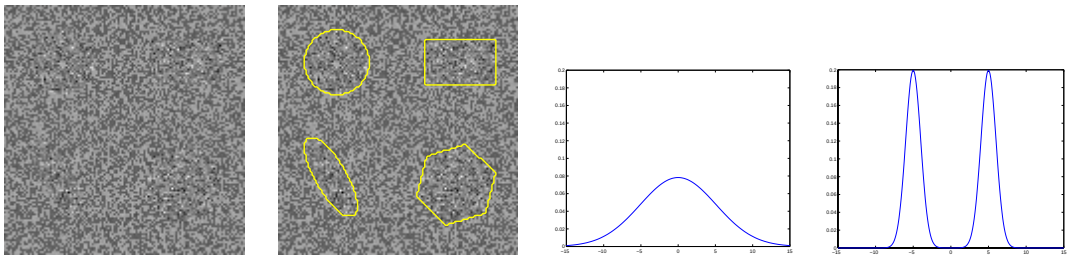
Figure 3.6. Evolution of the curve on a synthetic image without the additional two terms; the different variance case.

segmentation results without including the two additional term. Without computing the two additional terms, the run time was 117 seconds.

However, for other types of images, the log-likelihood ratio can be small, and the other two terms can become more important affecting the performance of the segmentation. For instance, if we do not include the additional two terms for the segmentation of the image in Figure 3.4(a), we observe a loss in the accuracy of the segmentation as illustrated in Figure 3.6. We observe that the sharp corners of the rectangle are missed. A similar performance loss due to excluding these additional terms is also pointed out by Jehan-Besson [30][5]. Based on these empirical observations, we believe this is an issue that requires further analysis in future work.

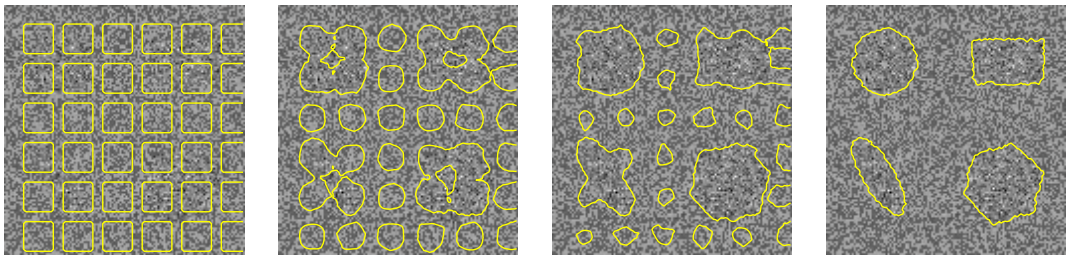
The next synthetic example we consider involves a more challenging image shown in Figure 3.7(a). The underlying distributions of the foreground and the background are a unimodal Gaussian density and a bimodal density with two Gaussian components as illustrated in Figure 3.7(c) and Figure 3.7(d) respectively. The two distributions have the same mean and same variance, so it is hard even for a human observer to separate the foreground from the background. In order to let the readers see the foreground, we show the actual boundaries in Figure 3.7(b). For this kind of image, the methods based on means and variances such as that proposed by Yezzi et al. [74] would no longer work. Figure 3.8 shows our segmentation results. As shown in Figure 3.8(a), we have used an automatic initialization with multiple seeds. The power of the multiple-seed initialization is that it provides sensitivity to the possible presence of boundaries throughout the entire region. Figure 3.8(b) and Figure 3.8(c) show the intermediate

[5]The technique proposed by Jehan-Besson et al. [30] is related to our work regarding these additional terms. The work in [30] considers general region-based active contours, where the energy functionals to minimize are given as region-integrals of so-called descriptors. In particular, they consider the case where the descriptors themselves depend on the region, and formulate an optimization method. Their formulation can also be applied to our energy functional, which is also region-based. What is new with our method is that our energy functional is based on mutual information and that our "descriptor" involves nonparametric density estimates, whereas they consider means, variances, determinants of covariance matrices, and histograms (in their subsequent work [4]) as the descriptors.



(a) image (b) with boundary (c) p_1 (d) p_2

Figure 3.7. Example image with two regions (boundaries marked in (b)), where the foreground has a unimodal density p_1, and the background has a bimodal density p_2. The two densities p_1 and p_2 have the same mean and the same variance.

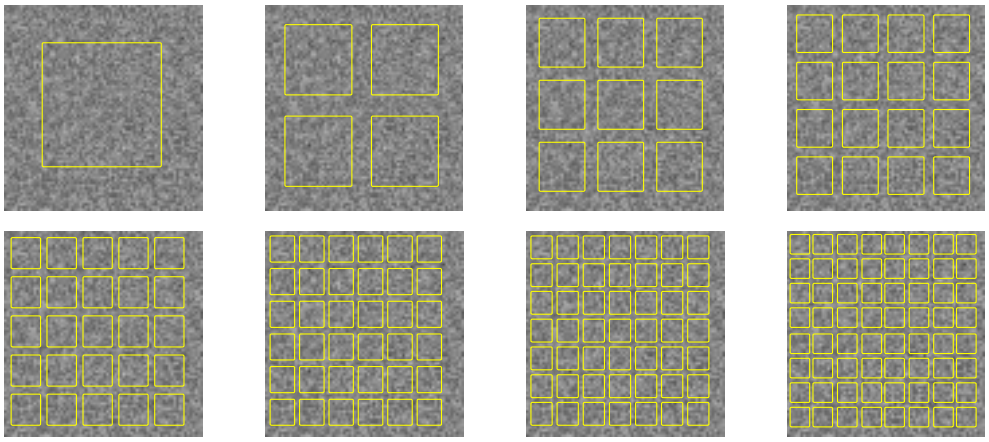


(a) initial (b) intermediate (c) intermediate (d) final

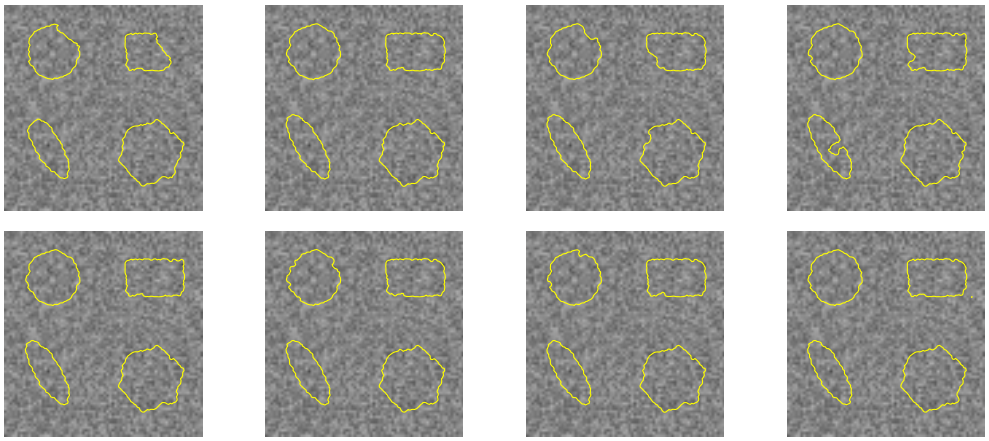
Figure 3.8. Evolution of the curve on a synthetic image; unimodal versus bimodal densities.

stages of the evolution, where the seeds in the background region gradually shrink at each iteration, whereas those in the foreground region grow. The final result shown in Figure 3.8(d) appears to be an accurate segmentation. Similarly, the next synthetic example in Figure 3.10 involves two distributions with the same mean and the same variance, where the foreground distribution is uniform and the background one is bimodal with two Gaussian components. As shown in Figure 3.11, our method can detect the foreground objects without any prior knowledge about the probability densities involved.

We empirically analyze the sensitivity of our segmentation results to initialization. In Figure 3.9, we run our algorithm on the same image as the one generated from unimodal and bimodal densities in Figure 3.7 with different initializations. Figure 3.9(a) shows various initializations with different number of seeds, and Figure 3.9(b) shows the corresponding segmentation results. As the upper row of Figure 3.9(b) shows, the segmentation can be suboptimal if we have a small number of seeds indicating that the segmentations depend on the initializations. However, the lower row of Figure 3.9(b) shows that as long as the number of seeds is large enough, the segmentation result is stable with respect to initializations even for this challenging example. It will be a



(a) initial curves

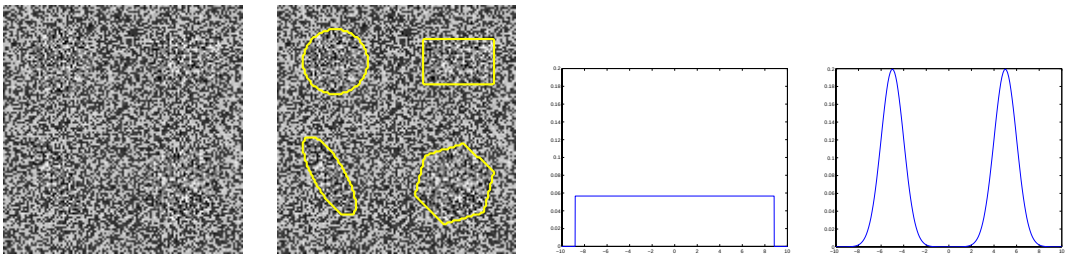


(b) corresponding segmentation results

Figure 3.9. Segmentations of the image in Figure 3.7(a) with various initializations. (a) Eight different initializations with varying number of seeds. (b) Corresponding segmentation results.

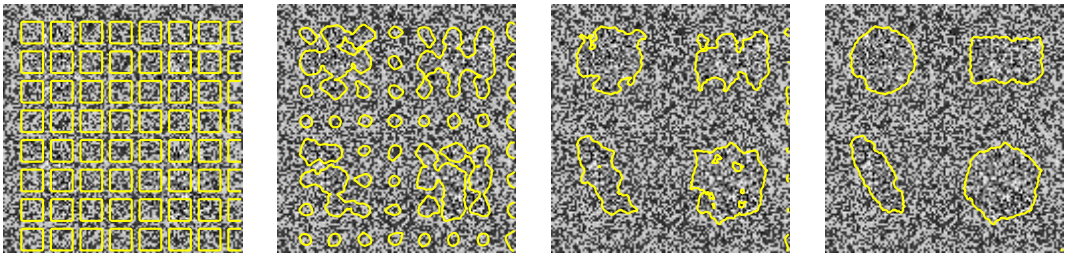
worthwhile future work to analyze the dependence of the curve evolution on the initialization. At this point we can give a rule of thumb for initializations with multiple seeds that the seeds need to cover the entire region such that they intersect with both the foreground and the background with high probability and that the number of seeds needs to be large enough.

Let us now consider the challenging examples in Figure 3.8 and Figure 3.11. If we did not have access to the underlying truth (as shown in Figure 3.7 and Figure 3.10), then based on the data and the results in Figure 3.8 and Figure 3.11, one might naturally ask the question of whether there are really two regions (i.e. foreground and background) here as the segmentations suggest, or whether there is only a single region. This raises the issue of statistical significance of a given result. We can address



(a) image (b) with boundary (c) p_1 (d) p_2

Figure 3.10. Example image with two regions (boundaries marked in (b)), where the foreground has a uniform density p_1, and the background has a bimodal density p_2. The two densities p_1 and p_2 have the same mean and the same variance.

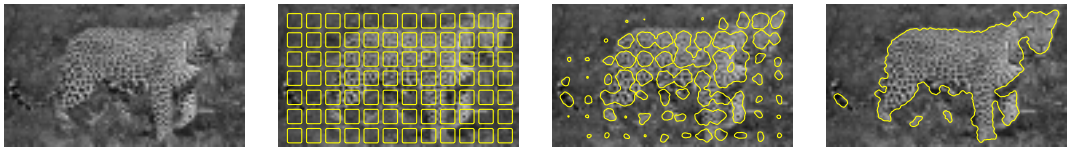


(a) initial (b) intermediate (c) intermediate (d) final

Figure 3.11. Evolution of the curve on a synthetic image; uniform (foreground) versus bimodal (background) densities.

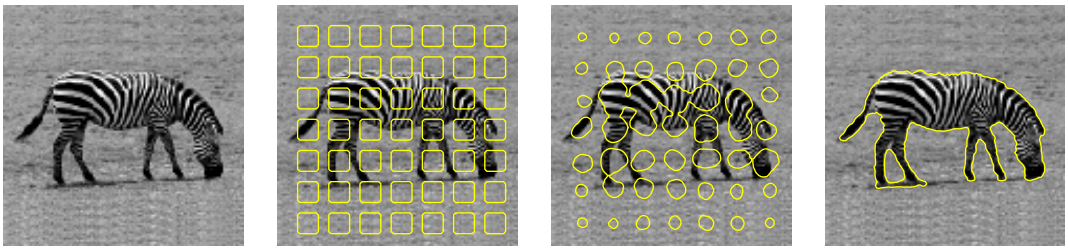
this issue by considering the null hypothesis H_0 that there is only one region versus the alternative hypothesis H_1 that there are two regions. We present the details of this analysis in Appendix B.2.1, where we observe that the key quantity involved here is again the mutual information. Using the mutual information in such a hypothesis test is appealing, due to the equivalence of the test based on mutual information and the likelihood ratio test. Specifically, the log-likelihood ratio $\log \frac{p(\{G(x)|x\in\Omega\}|H_1)}{p(\{G(x)|x\in\Omega\}|H_0)}$ is given by the number of pixels times the mutual information estimate, i.e. $|\Omega|\hat{I}(G(X); L_{\vec{C}}(X))$, which leads to the following interpretations: First, the higher the mutual information, the more different the density estimates $\hat{p}_+$, $\hat{p}_-$ are, and thus the more confidence we have. Second, the larger the number of pixels, the more accurate those density estimates are. Based on these observations, we take $\hat{I}(G(X); L_{\vec{C}}(X))$ as a statistic, and generate samples of this statistic under the null hypothesis H_0. The procedure for generating these samples is described in Appendix B.2.2. Next we compute the sample mean $E[\hat{I}|H_0]$ and the sample variance $Var[\hat{I}|H_0]$ of $\hat{I}(G(X); L_{\vec{C}}(X))$ under H_0. Finally, we evaluate whether the mutual information estimate $\hat{I}_{\text{seg}}(G(X); L_{\vec{C}}(X))$ produced by our segmentation result is a likely outcome under the null hypothesis. For this evaluation,

we simply use the Z-value [27], $Z \triangleq \frac{\hat{I}_{\text{seg}} - E[\hat{I}|H_0]}{\sqrt{Var[\hat{I}|H_0]}}$, which measures the distance between the observed value $\hat{I}_{\text{seg}}$ and the mean under H_0, in terms of the number of standard deviations. Large values indicate that the result is significant, hence the null hypothesis can be rejected[6]. For the result shown in Figure 3.8(d) and Figure 3.11(d) the Z-values are 4.24 and 5.63, respectively. These values are unlikely to occur under the null hypothesis, which thereby indicate that the segmentation results we have are statistically significant.



(a) original (b) initial (c) intermediate (d) final

Figure 3.12. Evolution of the curve on a leopard image.



(a) original (b) initial (c) intermediate (d) final

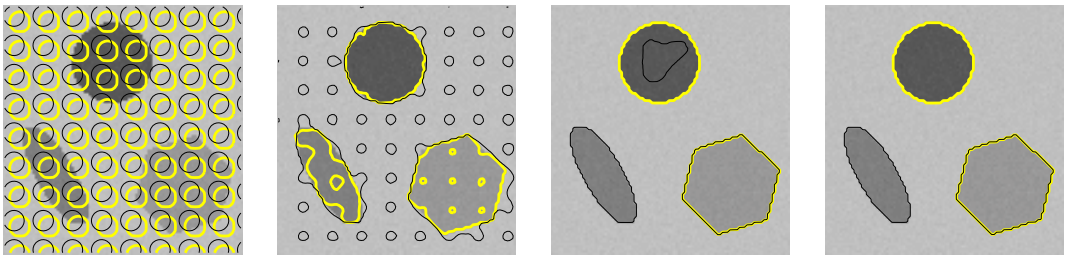
Figure 3.13. Evolution of the curve on a zebra image. (Input image: courtesy of Nikos Paragios)

We now report the result for a leopard image and a zebra image shown in Figure 3.12 and Figure 3.13 respectively. Both of these are challenging segmentation problems, where methods based on single statistics may fail. Figure 3.12(d) shows the segmentation result for the leopard image. The final curve captures the main body of the leopard and some parts of its tail and legs. The parts of the tail and the legs that are missing look similar to the background, which makes a perfect segmentation difficult. Figure 3.13 shows the success of our method in segmenting the zebra image, which is the identical zebra image used in Paragios et al. [51]. Their supervised texture segmentation algorithm requires an image patch taken from the object and an image patch

[6]It is unknown whether the distribution of $\hat{I}$ under the null hypothesis is Gaussian. If the distribution is Gaussian, the Z-value will specify the probability of false alarm, which decreases rapidly as the Z-value increases. Even if the distribution of $\hat{I}$ is not Gaussian, we still expect $Pr(|Z| > \gamma)$ to decrease rapidly as γ increases.

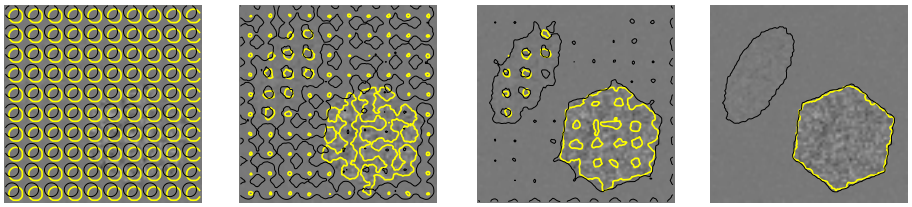
taken from the background in advance as an input to the algorithm. In contrast, the merit of our method is that we do not have to know or choose which feature to use and that the method nonparametrically estimates probability density functions and uses the density estimate as a statistical feature. It is noteworthy that our method, which is unsupervised, can segment this complex image as accurately as their supervised algorithm. Regarding the computational costs, on an Intel Xeon 2.2 GHz cpu, the nonparametric method took 211 seconds for segmenting the zebra image, whose size is 115 by 115.

Although our method can segment textured images without prior training, there are some classes of images where our framework breaks down. For instance, if one region has a texture with a marginal distribution p_1, and the other region has a different texture with the same marginal distribution p_1, then such an image can not be segmented without using a preprocessing such as one based on filter banks.



(a) initial (b) intermediate (c) intermediate (d) final

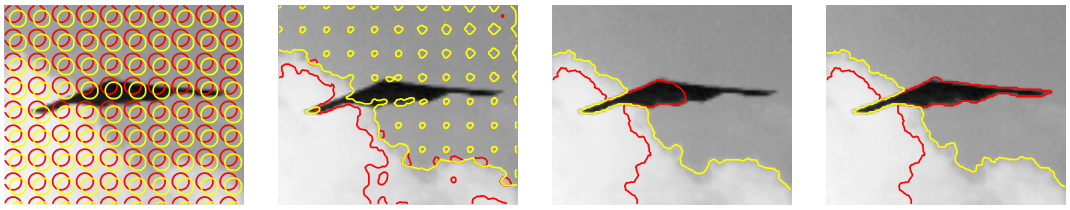
Figure 3.14. Evolution of the curve on a synthetic image; four regions with different mean intensities.



(a) initial (b) intermediate (c) intermediate (d) final

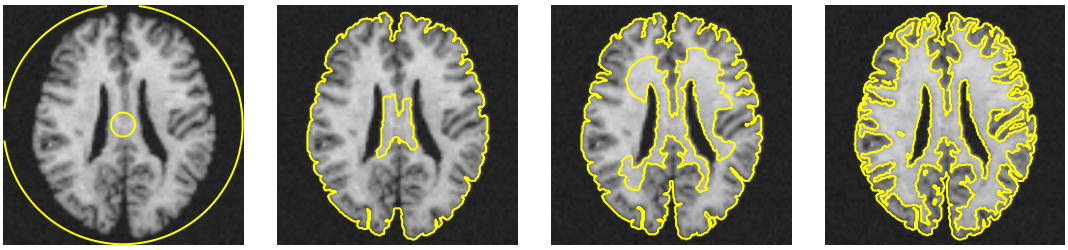
Figure 3.15. Evolution of the curve on a synthetic image; three regions with different mean intensities.

Now we present the results of our information-theoretic, multi-phase segmentation method on synthetic images of geometric objects, as well as real images. The image shown in Figure 3.14(a) contains four regions (circle, ellipse, hexagon, and background) with Gaussian distributions with different means. Hence, in this case we have $m = 2$, $n = 4$. The initial, intermediate, and final stages of our curve evolution algorithm are shown in Figure 3.14, where the four regions $R_{++}, R_{+-}, R_{-+}, R_{--}$ determined by the two curves capture the circle, the background, the hexagon, and the ellipse, respectively. Note that, methods such as that of [74] would also work for this simple example, but



(a) initial (b) intermediate (c) intermediate (d) final

Figure 3.16. Evolution of the curve on an aircraft image.



(a) initial (b) intermediate (c) intermediate (d) final

Figure 3.17. Evolution of the curve on a brain image.

would require the selection of an appropriate statistic (in this case the mean) *a priori*, whereas our method does not. The Mumford Shah-based multi-phase technique of [11], would also work in this case. Figure 3.15(a) contains an example with three regions having Gaussian distributions with different variances, hence $m = 2, n = 3$. In this case, R_{+-}, R_{-+}, and R_{--} capture the background, the hexagon, and the ellipse, respectively, whereas R_{++} shrinks and disappears.

Figure 3.16(a) shows an image of an airplane. The two curves in the final segmentation in Figure 3.16(d) capture the four regions, the airplane, the sky, the white clouds, and the darker clouds.

Figure 3.17(a) shows a brain image, which has three regions, the background, the white matter, and the gray matter. The proposed multiphase segmentation method can capture the white matter, the gray matter, and the background as in Figure 3.17(d). Since each region of this brain image can be distinguished from others by its mean intensity, a three-region segmentation method proposed by Yezzi et al. [74] and the multi-phase segmentation method of Chan and Vese [11] would also work for this image.

■ 3.5 Conclusion

We have developed a new information-theoretic image segmentation method based on nonparametric statistics and curve evolution. We have formulated the segmentation problem as one of maximizing the mutual information between the region labels and

the pixel intensities, subject to curve length constraints. We have derived the curve evolution equations for the optimization problem posed in our framework. Due to the nonparametric aspect of our formulation, the proposed technique can automatically deal with a variety of segmentation problems, in which many currently available curve evolution-based techniques would either completely fail or at least require the *a priori* extraction of representative statistics for each region. We use fast techniques for the implementation of nonparametric estimation, which keep the computational complexity at a reasonable level. Our experimental results have shown the strength of the proposed technique in accurately segmenting real and synthetic images.

Chapter 4

Nonparametric Shape Priors

When segmenting images of low quality or with missing data, prior information about the shape of the object can significantly aid the segmentation process. For instance, when radiologists segment magnetic resonance images of the prostate [67], whose boundary is difficult to find for laymen, they not only observe the intensities but also use their prior knowledge about the anatomical structure of the organ.

The problem in which we are interested is to extract such prior information from available example shapes and use it in segmentation. In particular, we want the prior information in terms of a shape prior distribution such that for a given arbitrary shape we can evaluate the likelihood of observing this shape among shapes of a certain category (e.g. the prostate). However, defining probability densities in the space of shapes is an open and challenging problem.

In this chapter, we propose a nonparametric shape prior model. In particular, we assume that the example shapes are drawn from an unknown shape distribution, and we estimate the underlying shape distribution by extending a Parzen density estimator to the space of shapes. Such density estimates are expressed in terms of distances between shapes. We then incorporate the shape prior distribution into an energy functional for segmentation, and derive the gradient flow for curve evolution. We present some experimental results of segmenting occluded images.

Since our approach is nonparametric, it can deal with a variety of shape densities beyond Gaussian ones. In that sense, it has more modeling capacity than traditional PCA-based approaches. It is also flexible in that it can be combined with a variety of shape distance measures. Also when combined with level set methods, our nonparametric prior can deal with not only 2D shapes but also 3D shapes. In addition to segmentation, these nonparametric shape distributions could be useful for a variety of statistical analysis tasks such as shape classification.

■ 4.1 Problem Statement and Relevant Issues

■ 4.1.1 Motivation for Shape Priors

Let us revisit the segmentation problem. If an image to be segmented is of high contrast, the image intensity data provide a large amount of information about the true boundary.

If the image is of lower contrast, the amount of information the data provide will be smaller. If the image has missing data around a portion of the boundary, the data provide little information about that portion of the boundary. Low contrast images and images with missing data are examples of low quality images, and for such images data alone will not be sufficient for accurate segmentation. Considering that segmentation is equivalent to extracting the pose and the shape of the boundary of the object, prior information on shapes will be helpful in segmentation, if we have any such information.

Now let us consider the case where we know the category of the object in the image. If there is only one possible fixed shape in that category, then we know the exact shape of the object a priori, and the segmentation problem comes down to estimation of pose. However, in general, there is shape variation even within a single category of objects, so that there are considerably more "candidate" shapes in the image than those corresponding simply to variations in pose. Since such candidate shapes may not be equally likely, it is desirable to have a quantitative measure of how good a candidate shape is or how likely such a shape is. In this sense, a probability measure on the set of shapes of a given category is the desirable description of the prior knowledge about shapes of the objects in the category.

Now the question is how to compute such a probability measure on a set of shapes. An intuitive idea is that a shape is more likely if it is similar to the shapes of the same category seen before. This raises the issue of how to define a notion of similarity. Mathematically, this suggests that a measure of distance between two shapes will play an important role in statistical analysis of shapes. In the following section, we state more formally the problem of building shape priors from available example shapes.

■ 4.1.2 Problem of Building a Shape Prior

In the previous chapter, a curve length penalty term $\alpha \oint_C ds$ was used for regularization. The basic idea behind this is that shorter curves are more likely than longer ones. Such a regularization term can be considered as a prior term, more accurately, the negative logarithm of a prior density. This interpretation is motivated by the Bayesian interpretation of the energy functional[1] $E(C)$ for image segmentation.

$$E(C) = -\log p(\text{data}|C) - \log p_C(C) \propto -\log p(C|\text{data}) \tag{4.1}$$

In this respect, the curve length term corresponds to the prior density for the curve $p_C(C) \propto e^{-\alpha \oint_C ds}$.

If we have additional information about the shape of the object to segment, we would like to build a more sophisticated shape prior and use it to guide the evolution of the curve C. In particular, we are interested in the case where we have a set of example shapes of the object class. Suppose that the example shapes are given in terms of n

[1] In statistical physics, the probability density of a certain quantum state is often given in terms of exponential of negative energy e^{-E}. For instance, a probability density for a particle taking an energy E is given by Boltzman distribution [37] $p(E) = \frac{1}{kT} e^{-E/(kT)}$, where k is Boltzman constant and T is the absolute temperature.

curves $C_1, \ldots, C_n$ that delineate the boundaries of the example shapes. The basic idea is that a candidate segmenting curve C will be more likely if it is similar to the example curves. Hence we would like to compare the candidate curve C with the example curves to evaluate how likely the curve C is.

However, when the candidate C and the training curves $C_1, \ldots, C_n$ are not aligned, a direct comparison of C with $C_1, \ldots, C_n$ will include not only the shape difference but also artifacts due to pose difference. In order to deal with this pose issue, we align the curves $C_1, \ldots, C_n$ and C into $\tilde{C}_1, \ldots, \tilde{C}_n$ and $\tilde{C}$, which have the same pose. In this chapter, we denote the aligned curves with tilde, whereas we denote the candidate curve C without tilde. Hence the procedure of computing $p_C(C)$ consists of the following steps:

1. Align $C_1, \ldots, C_n$ into $\tilde{C}_1, \ldots, \tilde{C}_n$
2. Align C w.r.t. $\tilde{C}_1, \ldots, \tilde{C}_n$ into $\tilde{C}$.
3. Evaluate $p_{\tilde{C}}(\tilde{C})$ the prior probability density of $\tilde{C}$ given $\tilde{C}_1, \ldots, \tilde{C}_n$.
4. Relate $p_{\tilde{C}}(\tilde{C})$ to $p_C(C)$.

We now discuss each of the above steps.

Alignment of Training Curves by Similarity Transforms

Here we discuss how to align the n training curves $C_1, \ldots, C_n$. In particular, we use the alignment algorithm presented in Tsai et al. [68], where a similarity transform is applied to each curve such that the transformed curves are well aligned. Let us first define the similarity transform and then provide a criterion for alignment.

The similarity transformation $T[\mathbf{p}]$ with the pose parameter $\mathbf{p}_i = [a\ b\ \theta\ h]$ consists of translation $M(a, b)$, rotation $R(\theta)$, and scaling $H(h)$, and it maps a point $(x, y) \in \mathbb{R}^2$ to $T[\mathbf{p}](x, y)$ as follows:

$$T[\mathbf{p}] \begin{pmatrix} x \\ y \end{pmatrix} = R(\theta) \circ H(h) \circ M(a, b) \begin{pmatrix} x \\ y \end{pmatrix} \tag{4.2}$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} h(x+a) \\ h(y+b) \end{pmatrix} \tag{4.3}$$

We define the transformed curve $T[\mathbf{p}]C$ to be the new curve that is obtained by applying the transformation to every point on the curve. The shape represented by a curve C can also be represented by a binary image $I(x, y)$ whose value is 1 inside C and 0 outside C. The transformation of $I(x, y)$ is defined to be the new image obtained by moving every pixel (x, y) of the image I to a new position $T[\mathbf{p}](x, y)$ making the intensity of $\tilde{I}$ at pixel $T[\mathbf{p}](x, y)$ the same as the intensity of I at pixel (x, y) as illustrated in Figure 4.1. Thus the two images I and $\tilde{I} \triangleq T[\mathbf{p}]I$ are related by

$$I(x, y) = \tilde{I}(T[\mathbf{p}](x, y)), \text{ for all } (x, y) \in \Omega. \tag{4.4}$$

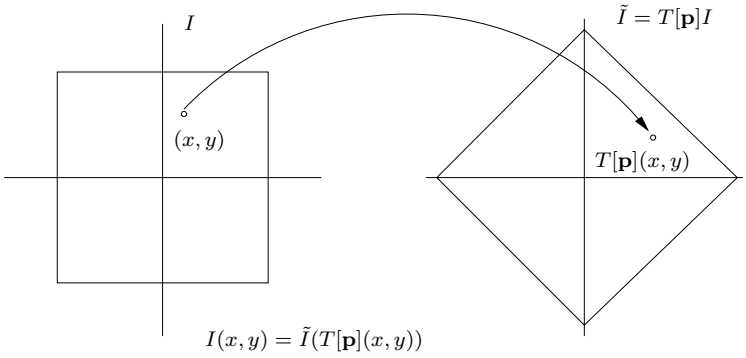


Figure 4.1. Illustration of the similarity transformation $T[\mathbf{p}]I$.

Equivalently, $\tilde{I}$ can be written in terms of I as follows:

$$\tilde{I}(x, y) = I(T^{-1}[\mathbf{p}](x, y)) \tag{4.5}$$

We now provide a criterion for alignment. Given n training curves, we obtain aligned curves $\tilde{C}_1, \ldots, \tilde{C}_n$ by a similarity transformation $\tilde{C}_i = T[\hat{\mathbf{p}}_i]C_i$ with pose estimate $\hat{\mathbf{p}}_i$ for each i. The pose estimates are chosen such that they minimize an energy functional for alignment. The energy functional we use is given by

$$E_{\text{align}}(\mathbf{p}_1, \ldots, \mathbf{p}_n) = \sum_{i=1}^{n} \sum_{j \neq i}^{n} \left\{ \frac{\iint_\Omega (T[\mathbf{p}_i]I^i - T[\mathbf{p}_j]I^j)^2 dxdy}{\iint_\Omega (T[\mathbf{p}_i]I^i + T[\mathbf{p}_j]I^j)^2 dxdy} \right\} \tag{4.6}$$

where I^i is a binary map whose value is 1 inside C_i and 0 outside C_i, and $T[\mathbf{p}]I^i$ is a transformed binary map whose value is 1 inside $T[\mathbf{p}_i]C_i$ and 0 outside $T[\mathbf{p}_j]C_j$. As in (4.5), I^i and $T[\mathbf{p}]I_i$ are related by

$$T[\mathbf{p}_i]I^i(x, y) = I^i[T^{-1}[\mathbf{p}_i](x, y)] \tag{4.7}$$

The numerator in (4.6), which is the area of set-symmetric difference between two interior regions of $T[\mathbf{p}_i]C_i$ and $T[\mathbf{p}_j]C_j$, basically measures the amount of mismatch between $T[\mathbf{p}_i]I^i$ and $T[\mathbf{p}_j]I^j$, and the denominator is present to prevent all the binary images from shrinking to improve the cost function.

To estimate the pose parameters, we fix the pose parameter for the first curve as the one for the identity transform and compute $\mathbf{p}_2, \ldots, \mathbf{p}_n$ by

$$\{\hat{\mathbf{p}}_2, \ldots, \hat{\mathbf{p}}_n\} = \arg \min_{\mathbf{p}_2, \ldots, \mathbf{p}_n} E_{\text{align}}(\mathbf{p}_1, \ldots, \mathbf{p}_n)|_{\mathbf{p}_1 = [0\ 0\ 0\ 1]} \tag{4.8}$$

Alignment of the Candidate Curve

Now we consider the problem of aligning the candidate curve C w.r.t. the n aligned training curves $\tilde{C}_1, \ldots, \tilde{C}_n$. To this end, we estimate a pose parameter $\hat{\mathbf{p}}$ such that

$\tilde{C} = T[\hat{\mathbf{p}}]C$ is well aligned to $\tilde{C}_1, \ldots, \tilde{C}_n$ by minimizing the energy:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} E(\mathbf{p}) = \sum_{i=1}^{n} \left\{ \frac{\int_\Omega (T[\mathbf{p}]I - \tilde{I}^i)^2 dx}{\int_\Omega (T[\mathbf{p}]I + \tilde{I}^i)^2 dx} \right\} \tag{4.9}$$

where I and $\tilde{I}^i$ are binary maps whose values are 1 inside and 0 outside C and $T[\hat{\mathbf{p}}_i]C_i$, respectively.

Evaluating the Shape Density

Now the problem is to estimate how likely the curve $\tilde{C}$ is, given the training curves $\tilde{C}_1, \ldots, \tilde{C}_n$. We consider the case where the n aligned curves are i.i.d. according to a density $p_{\tilde{C}}(\cdot)$. If one knows $p_{\tilde{C}}(\cdot)$ or has an estimate of the density, we can evaluate $p_{\tilde{C}}(\tilde{C})$, the likelihood of observing $\tilde{C}$ within the same class.

In Section 4.2, we address the problem of estimating the density $p_{\tilde{C}}(\cdot)$ from n i.i.d. samples $\tilde{C}_1, \ldots, \tilde{C}_n$. This is a challenging problem in that we are trying to build a probability measure on the shape space, which is an infinite dimensional space.

We can also represent example shapes by signed distance functions $\phi_1, \ldots, \phi_n$ rather than curves. In this case, the problem is to compute $p_{\tilde{\phi}}(\tilde{\phi})$ from n example shapes described by n signed distance functions $\tilde{\phi}_1, \ldots, \tilde{\phi}_n$ under the assumption that $\tilde{\phi}_1, \ldots \tilde{\phi}_n$ are i.i.d. according to $p_{\tilde{\phi}}(\tilde{\phi})$.

Relating $p_{\tilde{C}}(\tilde{C})$ to $p_C(C)$

We would like to relate the density $p_{\tilde{C}}(\tilde{C})$ to the prior density for the candidate curve $p_C(C)$. Since $C = T^{-1}[\mathbf{p}]\tilde{C}$, the two densities are related by

$$\begin{aligned} p_C(C) &= p_C(T^{-1}[\mathbf{p}]\tilde{C}) && (4.10) \\ &= p_{\tilde{C}}(\tilde{C}) p(\mathbf{p}|\tilde{C}). && (4.11) \end{aligned}$$

If the prior information about the pose $p(\mathbf{p}|\tilde{C})$ is available, one can use that information to evaluate $p_C(C)$. In this work, we assume that $p(\mathbf{p}|\tilde{C})$ is uniform, i.e. all pose $\mathbf{p}$ are equally likely. In this case, $p_C(C)$ is simply proportional to $p_{\tilde{C}}(\tilde{C})$, and we have

$$p_C(C) = \gamma p_{\tilde{C}}(\tilde{C}), \tag{4.12}$$

where γ is a normalizing constant.

■ 4.2 Nonparametric Shape Prior

In this section, we address the problem of estimating an unknown shape probability density $p_{\tilde{C}}(\tilde{C})$ or $p_{\tilde{\phi}}(\tilde{\phi})$, which is a probability density over an infinite dimensional space, from example curves $\tilde{C}_1, \ldots, \tilde{C}_n$, or from signed distance functions $\tilde{\phi}_1, \ldots, \tilde{\phi}_n$.

Let us first consider density estimation for a finite dimensional random vector. Suppose that we have n samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^m$ drawn from an m-dimensional density function $p(\mathbf{x})$. The Parzen density estimate is given by:

$$\hat{p}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{k}(\mathbf{x} - \mathbf{x}_i, \Sigma), \tag{4.13}$$

where we use an m-dimensional Gaussian kernel $\mathbf{k}(\mathbf{x}, \Sigma) = N(\mathbf{x}; 0, \Sigma^T \Sigma)$. If the kernel is spherical, i.e. $\Sigma = \sigma I$, the above density estimate becomes

$$\hat{p}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} k(d(\mathbf{x}, \mathbf{x}_i), \sigma), \tag{4.14}$$

where $d(\mathbf{x}, \mathbf{x}_i)$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_i$ in $\mathbb{R}^m$, and $k(x, \sigma)$ is the one dimensional Gaussian kernel $k(x, \sigma) = N(x; 0, \sigma^2)$.

Given a distance measure $d_{\mathcal{C}}(\cdot, \cdot)$ in $\mathcal{C}$, the space of curves, we can extend this Parzen density estimator with a spherical Gaussian kernel to the infinite dimensional space $\mathcal{C}$ as follows:

$$\hat{p}_{\tilde{C}}(\tilde{C}) \quad = \quad \frac{1}{n} \sum_{i=1}^{n} k(d_{\mathcal{C}}(\tilde{C}, \tilde{C}_i), \sigma) \tag{4.15}$$

In this density estimate, the composite of the one dimensional kernel and the distance metric plays the role of an infinite dimensional kernel. For the kernel size σ, we use an ML kernel size with leave-one-out as described in Section 2.4.2.

Similarly, we can build a shape density estimate $\hat{p}_{\tilde{\phi}}(\tilde{\phi})$ from n signed distance functions $\tilde{\phi}_1, \ldots, \tilde{\phi}_n$ as follows:

$$\hat{p}_{\tilde{\phi}}(\tilde{\phi}) = \frac{1}{n} \sum_{i=1}^{n} k(d_{\mathcal{D}}(\tilde{\phi}, \tilde{\phi}_i), \sigma) \tag{4.16}$$

where $d_{\mathcal{D}}(\cdot, \cdot)$ is a metric in $\mathcal{D}$, the space of signed distance functions.

With the nonparametric shape density estimate, we can also estimate information theoretic quantities associated with a random shape such as the entropy of a random shape or the KL divergence between two shape distributions. For details, see Appendix C.

Our nonparametric shape priors in (4.15) and (4.16) can be used with a variety of distance metrics. In the following sections, we consider two specific metrics, namely the template metric and the L_2 distance between signed distance functions. In particular, we claim that the Parzen density estimate with the L_2 distance can be a good approximation of the Parzen density estimate with the geodesic distance in Section 4.2.2. The template metric (described in Section 2.3.2), is given by the area of the set-symmetric difference between interior regions of two shapes. The template metric can be expressed

as a norm of difference between two *binary maps* (1 inside, 0 outside), whereas the second metric we use is a norm of difference between two *signed distance functions*. The key difference between these two metrics is that the set-symmetric difference between binary maps puts equal weight on pixels, whereas the difference between signed distance functions puts variable weight on pixels. We present Parzen density estimates based on these two metrics in the remainder of this section (with template metric in Section 4.2.1 and with L_2 norm in Section 4.2.2), and the corresponding shape-based segmentation algorithms in Section 4.3 (Section 4.3.2 for template metric and Section 4.3.3 for L_2 norm). It is possible to read the sections on the template metric first by skipping the sections on L_2 norm (Section 4.2.2 and Section 4.3.3), which could be read subsequently.

■ 4.2.1 Parzen Density Estimate with the Template Metric

We now consider the Parzen density estimate in (4.15) with a specific metric, namely the template metric $d_T(\tilde{C}, \tilde{C}_i) = \text{Area}(R_{\text{inside } \tilde{C}} \triangle R_{\text{inside } \tilde{C}_i})$ [45], where $\triangle$ denotes set symmetric difference. The density estimate with the template metric is given by

$$\hat{p}_{\tilde{C}}(\tilde{C}) = \frac{1}{n} \sum_{i=1}^{n} k(d_T(\tilde{C}, \tilde{C}_i), \sigma) \tag{4.17}$$

In Section 4.3.2, we will see that the gradient flow $\frac{\partial \tilde{C}}{\partial t}$ for this density estimate is given in closed form.

■ 4.2.2 Parzen Density Estimate on the Space of Signed Distance Functions

We now consider the space $\mathcal{D}$, which is a set of signed distance functions ϕ. We observe that this space is a subset of an infinite dimensional Hilbert space $\mathcal{L}$, which is defined by $\mathcal{L} \triangleq \{\phi | \phi : \Omega \to \mathbb{R}\}$. We can define an inner product in this space as follows:

$$\langle \phi_1, \phi_2 \rangle = \frac{1}{|\Omega|} \int_{\Omega} \phi_1(\mathbf{x}) \phi_2(\mathbf{x}) d\mathbf{x} \tag{4.18}$$

We also have an induced L_2 distance as follows:

$$d_{L_2}(\phi_1, \phi_2) = \sqrt{\langle \phi_1 - \phi_2, \phi_1 - \phi_2 \rangle} \tag{4.19}$$

Since the space $\mathcal{D}$ is embedded in a Hilbert space, a natural metric $d(\phi_1, \phi_2)$ for this space will be a minimum geodesic distance, i.e. the distance of the shortest path from ϕ_1 to ϕ_2 lying in the signed distance function space $\mathcal{D}$. Figure 4.2 provides a conceptual picture of the space $\mathcal{D}$, and the geodesic path connecting two distance functions ϕ_1 and ϕ_2. The direct line (dashed line) connecting ϕ_1 and ϕ_2 gives a shortest path in the Hilbert space and its length corresponds to the L_2 distance $d_{L_2}(\phi_1, \phi_2)$.

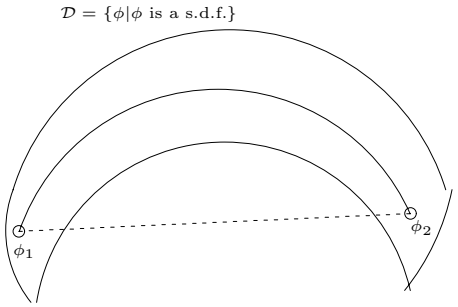


Figure 4.2. Illustration of the space of signed distance functions $\mathcal{D}$ and the geodesic path (solid line) between ϕ_1 and ϕ_2 compared with the shortest path in Hilbert space $\mathcal{L}$ (dashed line) which is off the space $\mathcal{D}$.

If one could compute the minimum geodesic distances $d_{\text{geodesic}}(\cdot, \cdot)$, the corresponding Parzen density estimate from samples $\{\tilde{\phi}_i\}$ would be

$$\hat{p}_{\tilde{\phi}}(\tilde{\phi}) = \frac{1}{n} \sum_i k(d_{\text{geodesic}}(\tilde{\phi}, \tilde{\phi}_i), \sigma). \tag{4.20}$$

However, computing a geodesic distance in an infinite dimensional manifold is a challenging problem. There is some previous work on computing geodesic distances in the space of curves such as Minchor and Mumford [43] and Klassen et al. [38], but there is little work when the shape is represented by signed distance functions.

Instead, we now consider the Parzen density estimate with the L_2 distance in $\mathcal{L}$

$$\hat{p}_{\tilde{\phi}}(\tilde{\phi}) = \frac{1}{n} \sum_i k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma). \tag{4.21}$$

Now we discuss why the above density estimate in (4.21) can be a good approximation of the one with geodesic distance in (4.20). Let us first consider the case where the example shapes are of small variation. Figure 4.3 illustrates this situation. In this case, the part of the manifold supporting the example shapes is approximately flat or linear provided that the manifold does not have too much curvature. This is why methods based on PCA of signed distance functions [40, 68] work reasonably well when there is small shape variation.

For the Parzen density estimate in such a case, we can take advantage of the same phenomenon, namely that the part of the manifold supporting example shapes is approximately flat and that the L_2 distance is close to the geodesic distance. Thus, in this case, the nonparametric density estimate with L_2 distance can be a good approximation of that with the geodesic distance.

Now consider the case where the example shapes have a broad range as illustrated in Figure 4.4. In this case, the part of the manifold supporting the samples is no

longer flat, and PCA is not a valid approach. In contrast, the density estimate with L_2 distance is still a good approximation of (4.20) for the following reasons. When $\tilde{\phi}$ and $\tilde{\phi}_i$ are close enough, the L_2 norm will be a good approximation of the geodesic distance. On the other hand, when $\tilde{\phi}$ and $\tilde{\phi}_j$ are far from each other, there will be an error in approximation of distance, but the overall error in density estimate will be small as long as the kernel size σ is small compared to the distance $d_{L_2}(\tilde{\phi}, \tilde{\phi}_j)$. The kernel size σ will be small provided that we have a sufficiently large number of example shapes.

More precisely, we have the following approximation for small $M\sigma$ and large M:

$$
\begin{aligned}
&\frac{1}{n}\sum_i k(d_{\text{geodesic}}(\tilde{\phi}, \tilde{\phi}_i), \sigma) \\
&= \frac{1}{n}\left(\sum_{\substack{i \\ d_{L_2}(\tilde{\phi}, \tilde{\phi}_i) \le M\sigma}} k(d_{\text{geodesic}}(\tilde{\phi}, \tilde{\phi}_i), \sigma) + \sum_{\substack{i \\ d_{L_2}(\tilde{\phi}, \tilde{\phi}_i) > M\sigma}} k(d_{\text{geodesic}}(\tilde{\phi}, \tilde{\phi}_i), \sigma) \right) \\
&\overset{(1)}{\approx} \frac{1}{n} \sum_{\substack{i \\ d_{L_2}(\tilde{\phi}, \tilde{\phi}_i) \le M\sigma}} k(d_{\text{geodesic}}(\tilde{\phi}, \tilde{\phi}_i), \sigma) \\
&\overset{(2)}{\approx} \frac{1}{n} \sum_{\substack{i \\ d_{L_2}(\tilde{\phi}, \tilde{\phi}_i) \le M\sigma}} k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma) \\
&\overset{(3)}{\approx} \frac{1}{n}\left(\sum_{\substack{i \\ d_{L_2}(\tilde{\phi}, \tilde{\phi}_i) \le M\sigma}} k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma) + \sum_{\substack{i \\ d_{L_2}(\tilde{\phi}, \tilde{\phi}_i) > M\sigma}} k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma) \right) \\
&= \frac{1}{n}\sum_i k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma),
\end{aligned} \tag{4.22}
$$

where

- We can make the approximation (2), provided that $M\sigma$ is small enough such that if $d_{L_2}(\tilde{\phi}, \tilde{\phi}_i) \le M\sigma$, $d_{\text{geodesic}}(\tilde{\phi}, \tilde{\phi}_i) \approx d_{L_2}(\tilde{\phi}, \tilde{\phi}_i)$
- We can make the approximation (1) and (3), provided that M is large enough such that if $d_{L_2}(\tilde{\phi}, \tilde{\phi}_i) > M\sigma$, $k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma) \approx 0$ and $k(d_{\text{geodesic}}(\tilde{\phi}, \tilde{\phi}_i), \sigma) \approx 0$.

These conditions can be satisfied if the kernel size σ is small enough.

A similar argument will hold for the case where the samples form multiple clusters as illustrated in Figure 4.5.

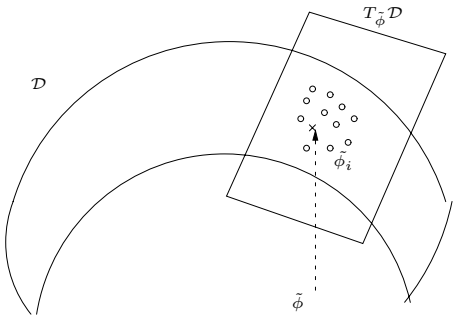


Figure 4.3. Illustration of example shapes in $\mathcal{D}$ with small shape variation.

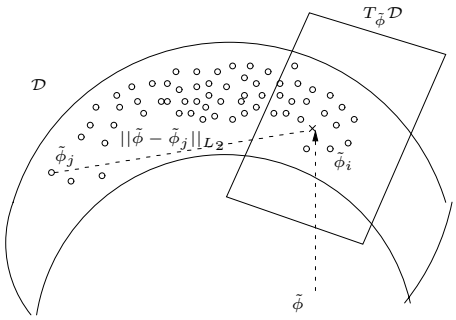


Figure 4.4. Illustration of example shapes in $\mathcal{D}$ with broad range.

■ 4.3 Shape-Based Segmentation

Now we combine the nonparametric shape prior and a data term within a Bayesian framework to form the energy functional for segmentation. The data term[2] we use is the one based on mutual information which we have developed in Chapter 3, and the shape term comes from the nonparametric shape priors introduced in Section 4.2. The task of segmentation is to minimize the energy functional[3].

$$E(C) = -\log p(\text{data}|C) - \log p_C(C) \tag{4.23}$$

We would like to minimize this functional by gradient descent, and the task comes down to computing the gradient flow for the curve C or the corresponding signed distance function ϕ. The gradient flow $\frac{\partial C}{\partial t}$ or $\frac{\partial \phi}{\partial t}$ for the data term is computed as is done in Chapter 3. So we only describe how to compute the gradient flow $\frac{\partial C}{\partial t}$ or $\frac{\partial \phi}{\partial t}$ for maximizing $\log p_C(C)$.

[2]We can also use any other data term such as the one in Mumford-Shah [44].
[3]From now on we drop the hat for simplicity in density estimate $\hat{p}_C(C)$.

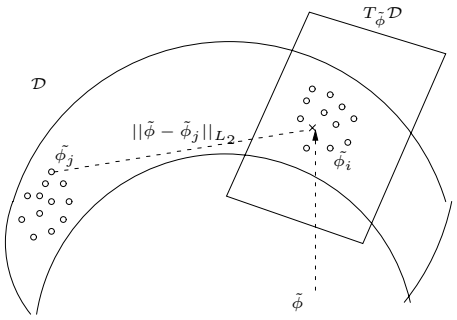


Figure 4.5. Illustration of two clusters of example distance functions in $\mathcal{D}$ and the tangent space at $\tilde{\phi} \in \mathcal{D}$.

However, we cannot compute $\frac{\partial C}{\partial t}$ directly from the shape prior, since as was mentioned in Section 4.1.2 the shape prior

$$\log p_C(C) = \log(\gamma p_{\tilde{C}}(\tilde{C})) \tag{4.24}$$

$$= \log \frac{1}{n} \sum_{i=1}^{n} k(d_{\mathcal{C}}(\tilde{C}, \tilde{C}_i), \sigma) + \log \gamma \tag{4.25}$$

basically compares the aligned curve $\tilde{C} = T[\mathbf{p}]C$ with the training curves $\{\tilde{C}_i\}$ and is given in terms of those aligned curves $\tilde{C}$ and $\{\tilde{C}_i\}$. Hence, we first compute $\frac{\partial \tilde{C}}{\partial t}$ from the shape prior, and then compute $\frac{\partial C}{\partial t}$ from $\frac{\partial \tilde{C}}{\partial t}$.

To this end, we need a pose parameter $\mathbf{p}$ for curve C at each time, and the pose $\mathbf{p}$ should be updated concurrently as the curve C evolves. The updates of C and $\mathbf{p}$ are performed iteratively according to Algorithm 1[4]. All the steps except step 3-(a)-ii and step 3-(c)-i are straightforward. Step 3-(c)-i is discussed in Section 4.1.2, and we discuss step 3-(a)-ii in the following sections.

Similarly, for the shape prior given in terms of signed distance functions $\tilde{\phi}$ and $\{\tilde{\phi}_i\}_{i=1}^n$

$$\log p_{\tilde{\phi}}(\tilde{\phi}) = \log \frac{1}{n} \sum_{i=1}^{n} k(d_{\mathcal{D}}(\tilde{\phi}, \tilde{\phi}_i), \sigma), \tag{4.26}$$

the same algorithm is used for the iterative updates of ϕ and $\mathbf{p}$ with $\log p_{\tilde{C}}(\tilde{C})$ replaced by $\log p_{\tilde{\phi}}(\tilde{\phi})$ and C, $\tilde{C}$, $\frac{\partial C}{\partial t}$, $\frac{\partial \tilde{C}}{\partial t}$ replaced by ϕ, $\tilde{\phi}$, $\frac{\partial \phi}{\partial t}$, $\frac{\partial \tilde{\phi}}{\partial t}$, respectively.

In following sections, we discuss how to compute the gradient flow $\frac{\partial \tilde{C}}{\partial t}$ or $\frac{\partial \tilde{\phi}}{\partial t}$ for maximizing the logarithm of the shape prior probability. We first start with the Parzen density estimate with a generic distance metric and give a sufficient condition so that the

[4] Analysis of the convergence properties of this iterative algorithm is a topic for future work.

1. Evolve the curve C without the shape prior for time $t \in [0, t_0]$
2. For the curve $C|_{t=t_0}$, compute the pose $\mathbf{p}|_{t=t_0}$ by aligning $C|_{t=t_0}$ with respect to $\{\tilde{C}_i\}$
3. Iterate until convergence:
   (a) fix $\mathbf{p}$ and
      i. compute $\tilde{C} = T[\mathbf{p}]C$.
      ii. compute $\frac{\partial \tilde{C}}{\partial t}$ from the shape prior $\log p_{\tilde{C}}(\tilde{C}) = \log \frac{1}{n} \sum_{i=1}^{n} k(d_{\mathcal{C}}(\tilde{C}, \tilde{C}_i), \sigma)$
      iii. compute $\frac{\partial C}{\partial t}$ from $\frac{\partial \tilde{C}}{\partial t}$ by $\frac{\partial C}{\partial t} = T^{-1}[\mathbf{p}] \frac{\partial \tilde{C}}{\partial t}$
   (b) update C by both the data driven force and the shape driven force
   (c) fix C and
      i. compute $\frac{\partial \mathbf{p}}{\partial t}$ using the alignment energy functional in Eqn. (4.9)
      ii. update the pose parameter $\mathbf{p}$ by $\frac{\partial \mathbf{p}}{\partial t}$

Algorithm 1: Iterative algorithm for update the pose estimate $\mathbf{p}$ and the curve C.

gradient flow is computable in Section 4.3.1. In particular, as an example for which the gradient flow is computable, we consider Parzen density estimation with the template metric in Section 4.3.2. Next, in Section 4.3.3, we discuss the case where the metric is the Euclidean distance between two signed distance functions and describe how to evolve the curve or the corresponding level set function in the direction of increasing the shape prior term $\log p_{\tilde{\phi}}(\tilde{\phi})$.

■ 4.3.1 Gradient Flow for the Shape Prior with a Generic Distance Metric

In this section, we derive a gradient flow for the Parzen window shape prior with a general distance measure. It turns out that the gradient flow is given as a weighted average of several directions, where the ith direction is an optimal (gradient) direction that decreases the distance between the ith training shape and the evolving shape.

Let us begin by considering the shape term

$$\log p_{\tilde{C}}(\tilde{C}) = \log \left(\frac{1}{n} \sum_{i} k(d_{\mathcal{C}}(\tilde{C}, \tilde{C}_i), \sigma) \right) \tag{4.27}$$

where $d_{\mathcal{C}}(\tilde{C}, \tilde{C}_i)$ is a generic distance measure between the shape described by the curve $\tilde{C}$ and the ith training shape described by $\tilde{C}_i$. Note that $\tilde{C}$ is function of the time t and $\tilde{C}$ is a shorthand notation for the evolving curve $\tilde{C}(t)$. Now we need to compute a velocity field f in curve evolution $\tilde{C}_t = f\vec{N}$ that increases $\log p_{\tilde{C}}(\tilde{C})$ most rapidly.

The time derivative of $\log p_{\tilde{C}}(\tilde{C})$ is given by

$$\frac{\partial \log p_{\tilde{C}}(\tilde{C})}{\partial t} = \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{n}\sum_i k'(d_{\mathcal{C}}(\tilde{C},\tilde{C}_i),\sigma)\frac{\partial d_{\mathcal{C}}(\tilde{C},\tilde{C}_i)}{\partial t} \tag{4.28}$$

Now suppose that the last term $\frac{\partial d_{\mathcal{C}}(\tilde{C},\tilde{C}_i)}{\partial t}$ is given in the form of $\oint_{\tilde{C}} \left\langle \tilde{C}_t, f_i\vec{N}\right\rangle ds$, i.e. $\tilde{C}_t = -f_i\vec{N}$ decreases $d_{\mathcal{C}}(\tilde{C},\tilde{C}_i)$ most rapidly, then we have

$$\frac{\partial \log p_{\tilde{C}}(\tilde{C})}{\partial t} = \oint_{\tilde{C}} \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{n}\sum_i k'(d_{\mathcal{C}}(\tilde{C},\tilde{C}_i),\sigma)\left\langle \tilde{C}_t, f_i\vec{N}\right\rangle ds \tag{4.29}$$

$$= \oint_{\tilde{C}} \left\langle \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{n}\sum_i k'(d_{\mathcal{C}}(\tilde{C},\tilde{C}_i),\sigma) f_i\vec{N}, \tilde{C}_t\right\rangle ds, \tag{4.30}$$

and we have the following gradient direction that increases $\log p_{\tilde{C}}(\tilde{C})$ most rapidly:

$$\frac{\partial \tilde{C}}{\partial t} = \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{n}\sum_i k'(d_{\mathcal{C}}(\tilde{C},\tilde{C}_i),\sigma)(-f_i)\vec{N} \tag{4.31}$$

In our work, we use a Gaussian kernel $k(x,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{x^2}{2\sigma^2})$, and we have $k'(x,\sigma) = k(x,\sigma)(-\frac{x}{\sigma^2})$. Thus the gradient flow is given by

$$\frac{\partial \tilde{C}}{\partial t} = \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{n}\frac{1}{\sigma^2}\sum_i k(d_{\mathcal{C}}(\tilde{C},\tilde{C}_i),\sigma) d_{\mathcal{C}}(\tilde{C},\tilde{C}_i)(-f_i)\vec{N}, \tag{4.32}$$

which is a linear combination of n terms $\{-f_i\vec{N}\}_{i=1}^n$, where the ith term contributes a force that decreases the distance $d_{\mathcal{C}}(\tilde{C},\tilde{C}_i)$ most rapidly, and the weight for the ith term is given by $k(d_{\mathcal{C}}(\tilde{C},\tilde{C}_i),\sigma)d_{\mathcal{C}}(\tilde{C},\tilde{C}_i)$.

■ 4.3.2 Gradient Flow for the Shape Prior with the Template Metric

As we have seen above, if we can write the term $\frac{\partial d_{\mathcal{C}}(\tilde{C},\tilde{C}_i)}{\partial t}$ in the form of $\oint_{\tilde{C}} \left\langle \tilde{C}_t, f_i\right\rangle ds$, we have the gradient flow for $\log p_{\tilde{C}}(\tilde{C})$ in closed form. The template metric is such an example, and in this section we compute the gradient flow for the shape prior with the template metric introduced in Section 4.2.1.

Consider the template metric $d_T(\tilde{C},\tilde{C}_i) = \text{Area}(R_{\text{inside }\tilde{C}} \triangle R_{\text{inside }\tilde{C}_i})$. This metric can be written in the form of region integrals as follows:

$$d_T(\tilde{C},\tilde{C}_i) = \int_\Omega (1-H(\phi(\mathbf{x})))H(\phi_i(\mathbf{x}))d\mathbf{x} + \int_\Omega H(\phi(\mathbf{x}))(1-H(\phi_i(\mathbf{x})))d\mathbf{x}$$

$$= \int_{R_{\text{inside }\tilde{C}}} H(\phi_i(\mathbf{x}))d\mathbf{x} + \int_{R_{\text{outside }\tilde{C}}} (1-H(\phi_i(\mathbf{x})))d\mathbf{x}, \tag{4.33}$$

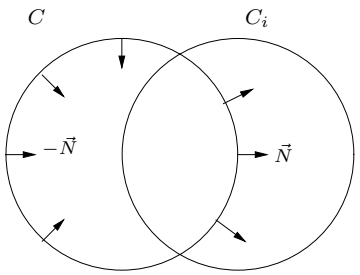


Figure 4.6. Illustration of the shape force that decreases the template metric $d_{\mathcal{C}}(C, C_i) = \text{Area}(R_{\text{inside } C} \triangle R_{\text{inside } C_i})$. $\vec{N}$ is the outward unit normal vector.

where ϕ and $\{\phi_i\}$ are signed distance functions for $\tilde{C}$ and $\{\tilde{C}_i\}$ respectively, and $H(\cdot)$ is the Heaviside function, i.e. $H(\phi)$=1 if $\phi \geq 0$ and $H(\phi) = 0$ if $\phi < 0$. For the region integrals in (4.33), the derivative is well known, which is given by

$$\frac{\partial d_T(\tilde{C}, \tilde{C}_i)}{\partial t} = \oint_{\tilde{C}} \left\langle \tilde{C}_t, (2H(\phi_i(s)) - 1) \right\rangle ds \tag{4.34}$$

By substituting $f_i = (2H(\phi_i(s)) - 1)$ into (4.32), we have the following gradient direction that increases $\log p_{\tilde{C}}(\tilde{C})$ based on the template metric most rapidly:

$$\frac{\partial \tilde{C}}{\partial t} = \frac{1}{p_{\tilde{C}}(\tilde{C})} \frac{1}{n} \frac{1}{\sigma^2} \sum_i k(d_T(\tilde{C}, \tilde{C}_i), \sigma) d_T(\tilde{C}, \tilde{C}_i)(1 - 2H(\phi_i))\vec{N}. \tag{4.35}$$

Figure 4.6 illustrates the i th component of this shape force. Note that $(1 - 2H(\phi_i))$ is 1 inside $\tilde{C}_i$ and -1 outside $\tilde{C}_i$.

Shape Prior with the Square Root of the Template Metric

It is possible to consider variants of the template metric, as alternative distance measures. Here, we consider the square root of the template metric as one such example. Note that, if we interpret the template metric as an L_1 norm of the difference between binary maps, its square root can be viewed as a L_2 norm. We consider the shape prior where the distance metric $d_{\mathcal{C}}(\tilde{C}, \tilde{C}_i)$ is given by the square root of the template metric

$$\hat{p}_{\tilde{C}}(\tilde{C}) = \frac{1}{n} \sum_{i=1}^{n} k(d_{\mathcal{C}}(\tilde{C}, \tilde{C}_i), \sigma) \tag{4.36}$$

$$= \frac{1}{n} \sum_{i=1}^{n} k(\sqrt{d_T(\tilde{C}, \tilde{C}_i)}, \sigma) \tag{4.37}$$

In this case, f_i in (4.32) comes from $\frac{\partial\sqrt{d_T(\tilde{C},\tilde{C}_i)}}{\partial t}$, which is given by

$$\frac{\partial\sqrt{d_T(\tilde{C},\tilde{C}_i)}}{\partial t} = \frac{1}{2\sqrt{d_T(\tilde{C},\tilde{C}_i)}}\frac{\partial d_T(\tilde{C},\tilde{C}_i)}{\partial t} \tag{4.38}$$

$$= \oint_{\tilde{C}} \left\langle \tilde{C}_t, \frac{1}{2\sqrt{d_T(\tilde{C},\tilde{C}_i)}}(2H(\phi_i(s))-1) \right\rangle ds \tag{4.39}$$

Thus substituting $d_{\mathcal{C}}(\tilde{C},\tilde{C}_i) = \sqrt{d_T(\tilde{C},\tilde{C}_i)}$ and $f_i = \frac{1}{2\sqrt{d_T(\tilde{C},\tilde{C}_i)}}(2H(\phi_i(s))-1)$ into (4.32), we have the following gradient direction that increases $\log p_{\tilde{C}}(\tilde{C})$ most rapidly:

$$\frac{\partial\tilde{C}}{\partial t} = \frac{1}{p_{\tilde{C}}(\tilde{C})}\frac{1}{n}\frac{1}{\sigma^2}\sum_i k(\sqrt{d_T(\tilde{C},\tilde{C}_i)},\sigma)\sqrt{d_T(\tilde{C},\tilde{C}_i)}\left(\frac{1}{2\sqrt{d_T(\tilde{C},\tilde{C}_i)}}(1-2H(\phi_i))\right)\vec{N}$$

$$= \frac{1}{2p_{\tilde{C}}(\tilde{C})}\frac{1}{n}\frac{1}{\sigma^2}\sum_i k(\sqrt{d_T(\tilde{C},\tilde{C}_i)},\sigma)(1-2H(\phi_i))\vec{N} \tag{4.40}$$

Note that the gradient flow is given as a linear combination of the force component $(1-2H(\phi_i))\vec{N}$ just as (4.35). The only difference is that the weight in (4.40) is now $k(\sqrt{d_T(\tilde{C},\tilde{C}_i)},\sigma)$. This weight is a monotonic function of the distance metric, $\sqrt{d_T(\tilde{C},\tilde{C}_i)}$, whereas the weight in (4.35) is not a monotonic function of the distance $d_T(\tilde{C},\tilde{C}_i)$.

■ 4.3.3 Approximation of the Gradient Flow for the Shape Prior with the Euclidean Distance

Now we deal with the problem of evolving the level set function $\tilde{\phi}$ given the shape prior $p_{\tilde{\phi}}(\tilde{\phi})$ in (4.21), which is the Parzen density estimate with L2 distance introduced in Section 4.2.2. One natural approach is to evolve $\tilde{\phi}$ in the space $\mathcal{L}$ along the gradient of $\log p_{\tilde{\phi}}(\tilde{\phi})$ w.r.t. $\tilde{\phi}$. In that case the evolving level set function is not necessarily a signed distance function and thus does not remain on the manifold of signed distance functions. Moreover, when the level set function is off the manifold, the evolution of the zero level set can be less stable than the case where the evolving level set function is constrained to be a signed distance function [50]. Hence, it is desirable to project back the evolving level set function to the manifold occasionally or to constrain the evolving level set function to stay on the manifold.

In this section, we start by computing the gradient flow for $\log p_{\tilde{\phi}}(\tilde{\phi})$, which evolves the level set function in the space $\mathcal{L}$ without the constraint that it stays on the manifold. We then modify the evolution equation such that the evolving level set function remains a signed distance function.

Unconstrained Gradient Flow of Level Set Functions

Without the constraint that the evolving level set function stays on the manifold $\mathcal{D}$, we compute the gradient flow for $\log p_{\tilde{\phi}}(\tilde{\phi})$

$$\log p_{\tilde{\phi}}(\tilde{\phi}) = \log \frac{1}{n} \sum_i k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma) \tag{4.41}$$

where $\tilde{\phi}_i$ is the signed distance function for the ith training shape. Note that $\tilde{\phi}$ is a function of the time t and $\tilde{\phi}$ is a shorthand notation for the evolving level set function $\tilde{\phi}(t)$. Using a Gaussian kernel, we have

$$k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \frac{1}{|\Omega|} \int_\Omega (\tilde{\phi}(\mathbf{x}) - \tilde{\phi}_i(\mathbf{x}))^2 d\mathbf{x} \right) \tag{4.42}$$

By differentiating the above expression, we have

$$\begin{aligned} \frac{\partial}{\partial t} k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma) &= k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma) \left[-\frac{1}{2\sigma^2} \frac{1}{|\Omega|} \int_\Omega 2(\tilde{\phi}(\mathbf{x}) - \tilde{\phi}_i(\mathbf{x})) \tilde{\phi}_t(\mathbf{x}) d\mathbf{x} \right] \\ &= \frac{1}{\sigma^2} k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i)) \left\langle \tilde{\phi}_i - \tilde{\phi}, \tilde{\phi}_t \right\rangle \end{aligned} \tag{4.43}$$

Let us now differentiate $\log p_{\tilde{\phi}}(\tilde{\phi})$ in (4.41).

$$\frac{\partial}{\partial t} \log p_{\tilde{\phi}}(\tilde{\phi}) = \frac{1}{p_{\tilde{\phi}}(\tilde{\phi})} \frac{1}{n} \sum_i \frac{\partial}{\partial t} k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma) \tag{4.44}$$

$$= \frac{1}{p_{\tilde{\phi}}(\tilde{\phi})} \frac{1}{\sigma^2} \frac{1}{n} \sum_i k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma) \left\langle \tilde{\phi}_i - \tilde{\phi}, \tilde{\phi}_t \right\rangle \tag{4.45}$$

$$= \left\langle \frac{1}{p_{\tilde{\phi}}(\tilde{\phi})} \frac{1}{\sigma^2} \frac{1}{n} \sum_i k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma)(\tilde{\phi}_i - \tilde{\phi}), \tilde{\phi}_t \right\rangle \tag{4.46}$$

Thus the gradient direction that increases $\log p_{\tilde{\phi}}(\tilde{\phi})$ most rapidly is

$$\frac{\partial \tilde{\phi}}{\partial t} = \frac{1}{p_{\tilde{\phi}}(\tilde{\phi})} \frac{1}{\sigma^2} \frac{1}{n} \sum_i k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma)(\tilde{\phi}_i - \tilde{\phi}) \tag{4.47}$$

This velocity field is given by a weighted average of $\{\tilde{\phi}_i - \tilde{\phi}\}_{i=1}^n$, where $\tilde{\phi}_i - \tilde{\phi}$ is the direction toward the ith training shape $\tilde{\phi}_i$, and the corresponding weight is $k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma)$. Note that the weight for the velocity component $\tilde{\phi}_i - \tilde{\phi}$ increases as $\tilde{\phi}$ gets closer to $\tilde{\phi}_i$. As a result, an example shape that is closer to the current shape becomes more important during the evolution of the shape.

Modifying the Evolution Equation

Now we describe how we modify the evolution equation (4.47) such that the evolving level set function remains a signed distance function. We start by rewriting the update equation (4.47) and defining the velocity field $v(\cdot)$ as follows:

$$\frac{\partial \tilde{\phi}(\mathbf{x},t)}{\partial t} = \frac{1}{p_{\tilde{\phi}}(\tilde{\phi}(\cdot,t))}\frac{1}{\sigma^2}\frac{1}{n}\sum_i k(d_{L_2}(\tilde{\phi}(\cdot,t),\tilde{\phi}_i),\sigma)(\tilde{\phi}_i(\mathbf{x}) - \tilde{\phi}(\mathbf{x},t)) \triangleq v(\mathbf{x}) \quad (4.48)$$

Now we modify the evolution in (4.48) and construct a new velocity field $v_{\text{new}}(\cdot)$ which guarantees that the evolving level set function is a signed distance function. The goal here is to extract relevant information for shape evolution from the velocity field $v(\cdot)$ and to construct $v_{\text{new}}(\cdot)$ such that the resulting trajectory of $\phi(\cdot,t)$ is on the space $\mathcal{D}$.

First we observe that the only components of the velocity field $v(\cdot)$ that directly impact the shape evolution are those defined at the points on the boundary $\tilde{C}(t) = \{\mathbf{x}|\tilde{\phi}(\mathbf{x},t) = 0\}$. In this respect, we take $v_{\text{new}}(\mathbf{x}) = v(\mathbf{x})$ if $\mathbf{x} \in \tilde{C}$. The next key property is that as long as the velocity v_{new} remains constant along the direction normal to the curve $\tilde{C}$, the evolving level set function $\tilde{\phi}(t)$ remains a signed distance function [75]. Since we have defined values of $v_{\text{new}}(\cdot)$ at all the boundary points, we can extend these values in the direction normal to the boundary. Such a procedure is equivalent to setting the velocity $v_{\text{new}}(\mathbf{x})$ at a point $\mathbf{x}$ equal to the boundary velocity $v(\mathbf{x}_{\tilde{C}})$, where $\mathbf{x}_{\tilde{C}}$ is the boundary point closest to the point $\mathbf{x}$.

In summary, we update the level set function $\tilde{\phi}$ by the modified velocity $v_{\text{new}}(\cdot)$ as follows:

$$\frac{\partial \tilde{\phi}(\mathbf{x},t)}{\partial t} = v_{\text{new}}(\mathbf{x}) = v(\mathbf{x}_{\tilde{C}}), \quad (4.49)$$

where $\mathbf{x}_{\tilde{C}}$ is the point on the curve closest to the point $\mathbf{x}$.

This $v_{\text{new}}(\cdot)$ is an approximation of the gradient flow which maximizes the change in the energy. How well this two-step procedure approximates the gradient flow is an issue that deserves further study.

■ 4.4 Experimental Results

Now we present experimental results demonstrating our segmentation method based on nonparametric shape priors. We first show shape-based segmentation results for segmenting occluded objects with various poses. When we say the image is occluded, we assume that the algorithm does not know which part of image is occluded. Next, we present experimental results on segmentation of handwritten digits with missing data, where the algorithm knows which portion of the data is missing. We present the experiments with handwritten digits as an example where the training examples form multiple clusters.

Figure 4.7. Training samples of the aircraft shape before alignment.

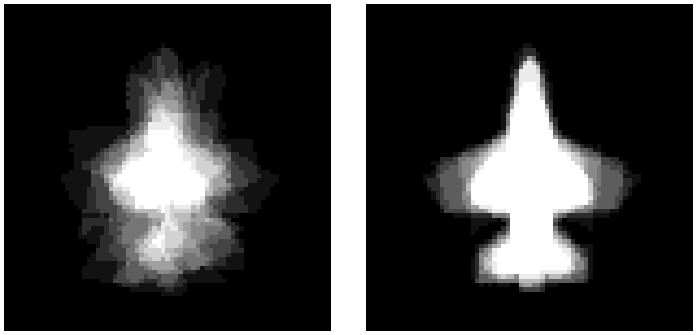
Figure 4.8. Aligned training samples of the aircraft shape.

■ 4.4.1 Segmentation of Occluded Objects with Various Poses

In this section, we demonstrate our shape-based segmentation algorithm with the segmentation of synthetic aircraft images. As example shapes of this class, we have a set of 11 binary images displayed in Figure 4.7, whose boundaries give the training curves $C_1, \ldots, C_n$ $(n = 11)$. Figure 4.8 shows the training shapes after alignment, hence the boundaries of these binary images correspond to the aligned training curves $\tilde{C}_1, \ldots, \tilde{C}_n$. Figure 4.9(a) and Figure 4.9(b) contain overlaid images of the training samples, showing the amount of overlap among training shapes before and after alignment respectively, and providing a picture of the shape variability.

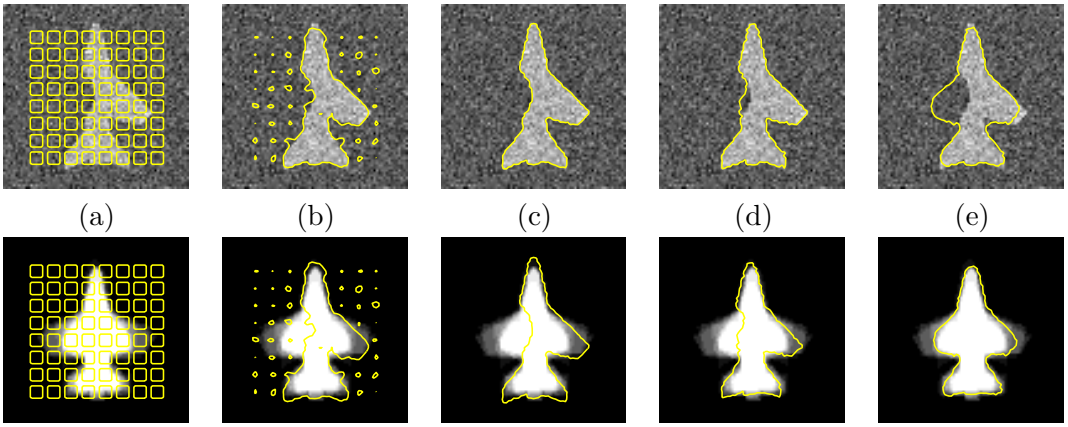
We now present segmentation results on the image of an aircraft whose shape was not included in the training set. In particular, Figure 4.10 shows the noisy aircraft test image with an occluded left-wing as well as its segmentation using the Parzen shape prior with L_2 distance between signed distance functions. The first row, (a)–(e), shows the evolving curve C on top of the image to be segmented, and the second row, (f)–(j), shows the transformed curve $\tilde{C} = T[\mathbf{p}]C$ on top of the aligned training shapes shown in Figure 4.9(b). In our shape-based segmentation, we evolve the curve as is given in Algorithm 1. First, the curve evolves without shape prior (using curve length regularization term) as shown in (a)–(c), which corresponds to the step 1 of Algorithm 1. After the curve finds all the portions of the object boundary except those occluded as shown in (c)[5], the shape force is turned on, and both the data force and shape force are applied during the stages (c)–(e). This procedure is more desirable than turning on the shape force from the start, since during the initial stages of the curve evolution, the pose estimate may not be accurate and in that case the shape force might deform the curve with the wrong pose estimate. Note that while the shape force is turned off, we need no pose estimates and we have $\tilde{C} = C$. It took 200 iterations to reach the final segmentation in (e). At the final segmentation the shape force and data force are in equilibrium. For instance the data force at the boundary of the left wing will try to shrink the left wing to match the given data, whereas the shape force tries to expand the left wing to increase the shape prior.

[5] At stage (c), the curve has converged with data force and curve shortening term. Such convergence is detected automatically and then the shape force is turned on.



(a) (b)

Figure 4.9. Overlay of training samples of the aircraft shape (a) before alignment (b) after alignment. The images (a) and (b) are generated by taking an average of the binary images in Figure 4.7 and Figure 4.8, respectively.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 4.10. Segmentation of an occluded aircraft image using Parzen shape prior with L_2 distance between signed distance functions. The first row,(a)–(e), shows the evolution of the curve C on top of the occluded image. The second row, (f)–(j), shows the aligned curve $\tilde{C}$ on top of the image shown in Figure 4.9(b). Images in the same column correspond to the same step during the iterative curve evolution process.

In these experiments, we have an issue of how to balance the data force and the shape force. We balance the two forces in such a way that the maximum value of one force over the boundary points is equivalent to the maximum of the other force, in order to prevent one force from dominating the other.

Figure 4.11 shows the same object with a different pose (rotated). Up to the stage shown in (c), the curve evolves without a shape force. Then the shape force is turned on and the pose parameter $\mathbf{p}$ is updated as is shown in (i) and (j) while the curve evolves as in (d) and (e). Figure 4.12 shows a segmentation example involving a rotated and scaled version of the object in Figure 4.10, and Figure 4.13 shows a segmentation example

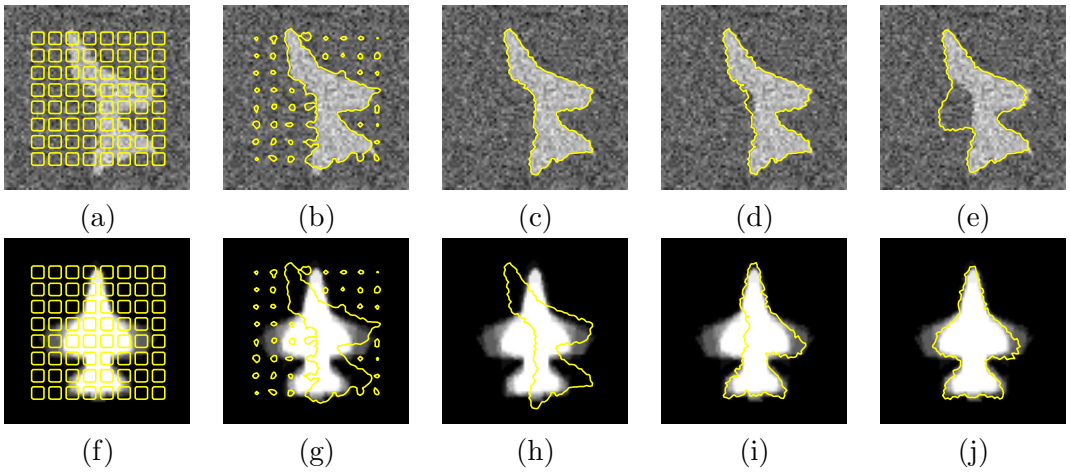


Figure 4.11. Segmentation of an occluded aircraft image (rotated) using Parzen shape prior with L_2 distance between signed distance functions. The first row,(a)–(e), shows the evolution of the curve C on top of the occluded image. The second row, (f)–(j), shows the aligned curve $\tilde{C}$ on top of the image shown in Figure 4.9(b).

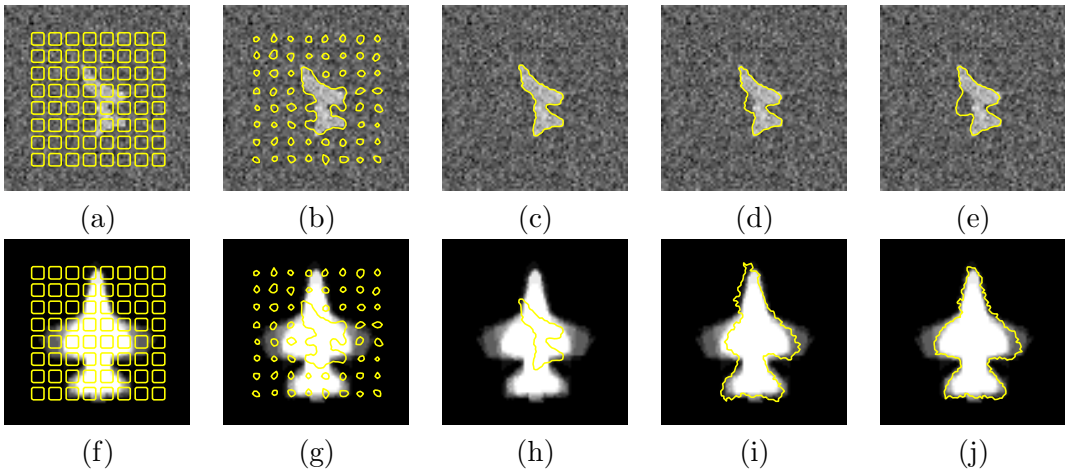


Figure 4.12. Segmentation of an occluded aircraft image (rotated, scaled) using Parzen shape prior with L_2 distance between signed distance functions. The first row,(a)–(e), shows the evolution of the curve C on top of the occluded image. The second row, (f)–(j), shows the aligned curve $\tilde{C}$ on top of the image shown in Figure 4.9(b).

involving a rotated, scaled and translated version of the same object in Figure 4.10. In all of these examples, we have reasonable segmentation despite the occlusions. These results demonstrate that our segmentation algorithm can locate an object with an arbitrary pose, when we have prior knowledge about the shape of the object.

Figure 4.14–Figure 4.17 show segmentation results using shape priors with the template metric. Let us briefly compare the case with the template metric with the case

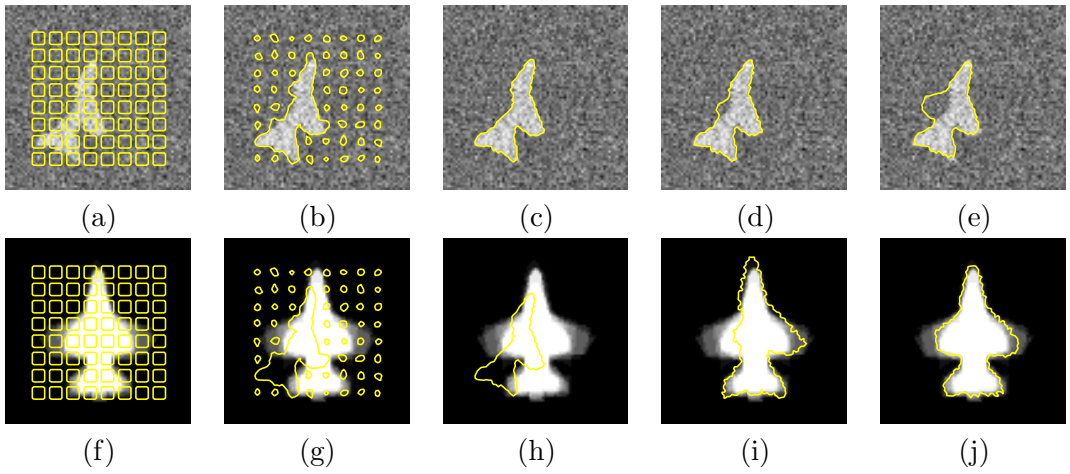


Figure 4.13. Segmentation of an occluded aircraft image (rotated, scaled, translated) using Parzen shape prior with L_2 distance between signed distance functions. The first row,(a)–(e), shows the evolution of the curve C on top of the occluded image. The second row, (f)–(j), shows the aligned curve $\tilde{C}$ on top of the image shown in Figure 4.9(b).

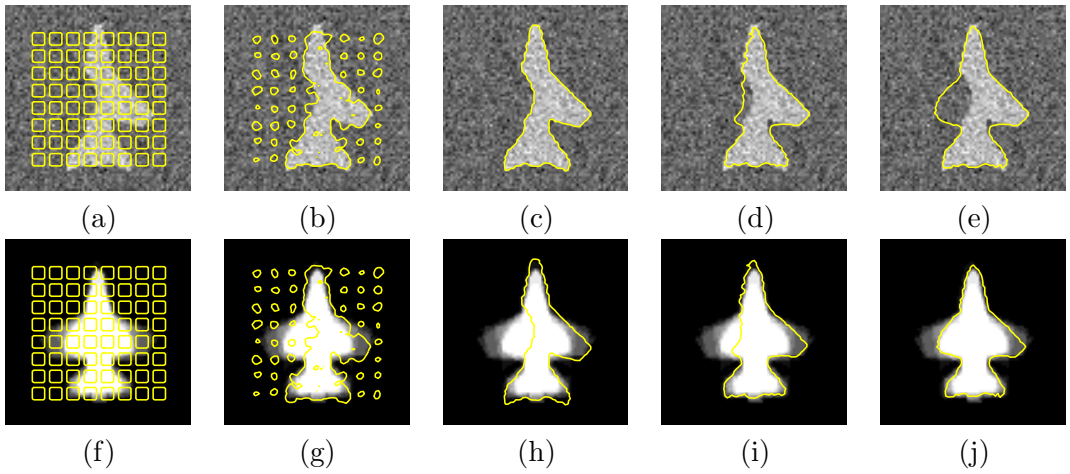


Figure 4.14. Segmentation of an occluded aircraft image using Parzen shape prior with the template metric. The first row,(a)–(e), shows the evolution of the curve C on top of the occluded image. The second row, (f)–(j), shows the aligned curve $\tilde{C}$ on top of the image shown in Figure 4.9(b).

with the L_2 distance between signed distance functions. With the L_2 distance, the ith term of the shape force is given by $\tilde{\phi}_i - \tilde{\phi}$, which is large when there is a significant gap between the current shape and the i th training shape. For instance, in Figure 4.10(d), the boundary of the right wing is already captured, thus the value of $\tilde{\phi}$ in that portion of the curve will be close to the values of signed distance functions for the aligned training shapes. As a result, the shape force component $\tilde{\phi}_i - \tilde{\phi}$ in the portion of the right wing will be of small value. In contrast, on the portion of the curve that just started to

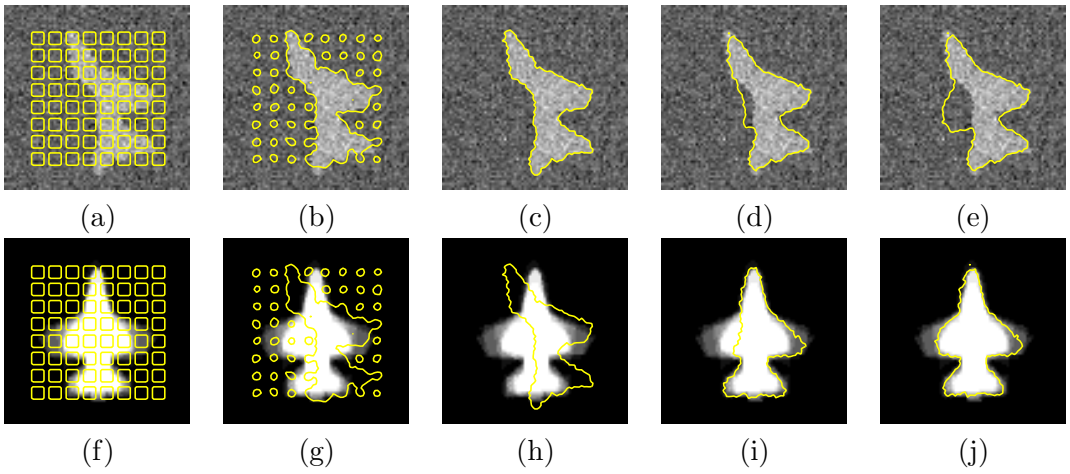


Figure 4.15. Segmentation of an occluded aircraft image (rotated) using Parzen shape prior with the template metric. The first row,(a)–(e), shows the evolution of the curve C on top of the occluded image. The second row, (f)–(j), shows the aligned curve $\tilde{C}$ on top of the image shown in Figure 4.9(b).

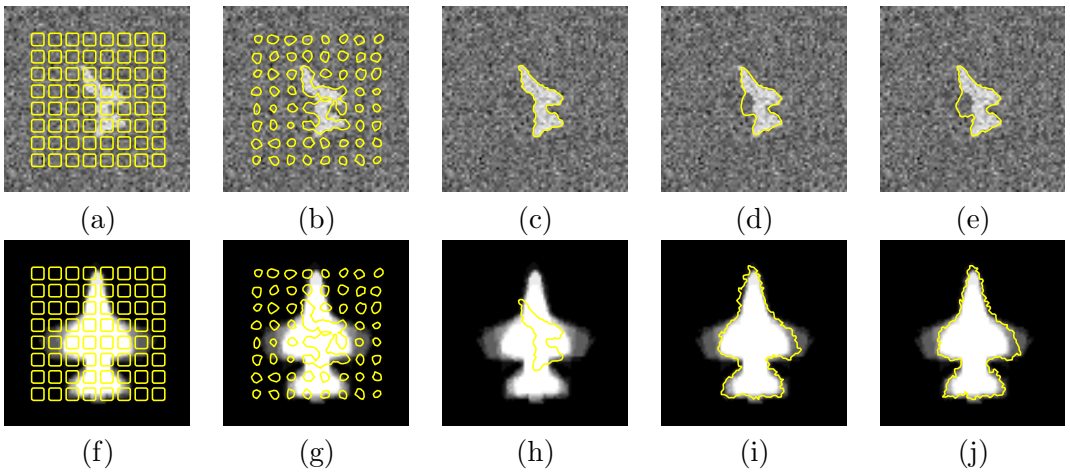


Figure 4.16. Segmentation of an occluded aircraft image (rotated, scaled) using Parzen shape prior with the template metric. The first row,(a)–(e), shows the evolution of the curve C on top of the occluded image. The second row, (f)–(j), shows the aligned curve $\tilde{C}$ on top of the image shown in Figure 4.9(b).

capture the left wing, there is still a significant gap between $\tilde{\phi}_i$ and $\tilde{\phi}$, thus the shape force at the boundary of the left wing will be of large value. Such shape force diminishes as the curve gets closer to the boundaries of training shapes. On the other hand, with the template metric, the shape force component $(1 - 2H(\tilde{\phi}_i))\vec{N}$ is of unit magnitude at all points on the boundary. For instance, in Figure 4.14(d) the portions of the curve for left wing will move in outward normal direction whose speed is less dependent on the gap between the current portion of the left wing and that of the training shapes.

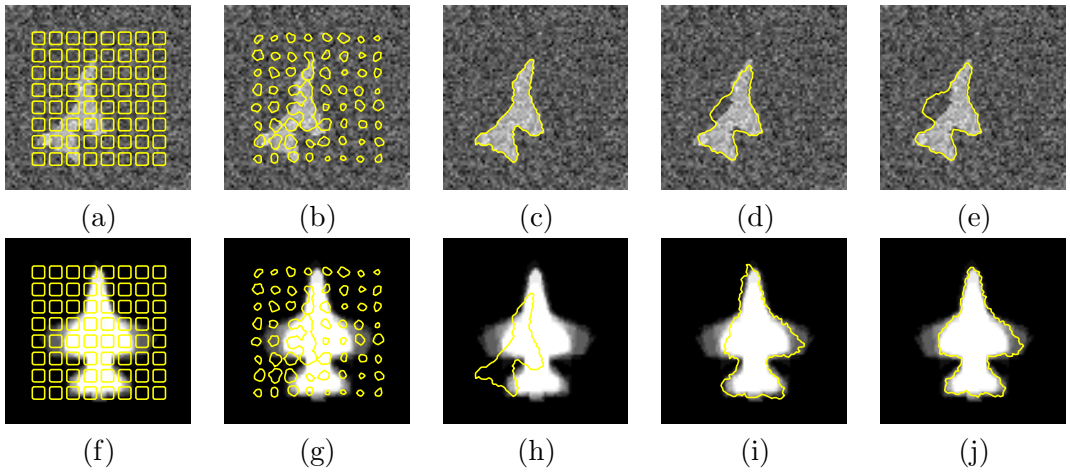


Figure 4.17. Segmentation of an occluded aircraft image (rotated, scaled, translated) using Parzen shape prior with the template metric. The first row,(a)–(e), shows the evolution of the curve C on top of the occluded image. The second row, (f)–(j), shows the aligned curve $\tilde{C}$ on top of the image shown in Figure 4.9(b).

With the template metric, the ith shape force component at a particular point on the boundary is either outward normal or inward normal depending on whether the point is inside or outside the ith training shape. Thus the evolution of the curve by the shape force will arrive at a stationary point when the curve is inside about the half of the training shapes and outside the remaining ones. In this sense, the shape force with template metric evolves the curve toward a local maximum of the shape prior, which in this case can be interpreted as a sort of median of neighboring training shapes.

In contrast, the shape force due to L_2 norm will evolve the curve toward a local maximum of the shape prior, which is approximately a weighted average of the neighboring training shapes. Let us consider the Eqn.(4.47). Although the actual shape force is modified version (Section 4.3.3) of the Eqn.(4.47), this equation gives a useful interpretation of a local maximum of the shape prior as follows. At the local maximum of $p(\tilde{\phi})$, the gradient flow will be zero.

$$\frac{\partial \tilde{\phi}}{\partial t} = \frac{1}{p_{\tilde{\phi}}(\tilde{\phi})} \frac{1}{\sigma^2} \frac{1}{n} \sum_i k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma)(\tilde{\phi}_i - \tilde{\phi}) \tag{4.50}$$

$$= \frac{1}{\sigma^2} \sum_i \lambda(\tilde{\phi}, \tilde{\phi}_i)(\tilde{\phi}_i - \tilde{\phi}) \tag{4.51}$$

$$= 0 \tag{4.52}$$

where $\lambda(\tilde{\phi}, \tilde{\phi}_i) = \frac{k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma)}{n p_{\tilde{\phi}}(\tilde{\phi})}$ and $\sum_i \lambda(\tilde{\phi}, \tilde{\phi}_i) = 1$. Hence the local maximum is given as

$$\tilde{\phi} = \sum_i \lambda(\tilde{\phi}, \tilde{\phi}_i) \tilde{\phi}_i \tag{4.53}$$

The nonlinear weight function $\lambda(\tilde{\phi}, \tilde{\phi}_i)$ will be negligible if $\tilde{\phi}_i$ is not in the local neighborhood (or within the same mode) of $\tilde{\phi}$ and the kernel size is small enough compared to $d_{L_2}(\tilde{\phi}, \tilde{\phi}_i)$. Thus the local maximum is approximately given as a weighted average of neighboring training shapes. Note that the weight function behaves as a selection function of the local neighbors or samples in the same cluster (or in the same mode). As the kernel size gets smaller, the neighboring samples contributing to the shape of local maximum are more localized, thus the part of the manifold that supports such neighboring samples will be more linear or flat. In short, the final segmentation is obtained at the equilibrium of the data force and shape force, where the shape force tries to evolve the current shape toward the weighted average of the local neighboring training shapes (in the case of L_2 distance) or toward the median of the training shapes (in the case of template metric).

If one knows the location of the occluder, we can obtain better segmentation results by avoiding the conflict between the data force and shape force simply by turning off the data force around the occluded region. Our current work involves estimation of the location of the occluder when the algorithm starts with no such knowledge. The idea is to first obtain shape-based segmentation results as in Figure 4.10(e) – Figure 4.16(e), and then use the statistics of the segmented regions to detect portions that are likely to be occluded. Given such occluder detections the segmentations can be refined further by running the algorithm with the data force turned off on the occluded region.

■ 4.4.2 Segmentation of Handwritten Digit Images

We now consider the case where we have multiple classes of example shapes. In particular, we consider the handwritten digits as such a case, where there are 10 classes of handwritten digits, i.e. 0, 1, ..., 9. Figure 4.18 shows the training shapes, where we have 10 sample images for each digit. In this experiment, the training shapes and test shapes are already aligned, so we fix the pose parameters $\mathbf{p}_i$ for the training curves and the pose parameter $\mathbf{p}$ for the evolving curve to be [0 0 0 1], the one for the identity transform. Hence $C_i = \tilde{C}_i$ and $C = \tilde{C}$, and we just use C_i and C to denote aligned curves. Let L denote the class label for digit taking values in $\{0, 1, \ldots, 9\}$. For each digit $l \in \{0, 1, \ldots, 9\}$, we can estimate the prior density $p(C|L = l)$ from the example shapes of the digit l.

Now we consider the problem of segmenting a handwritten digit image with missing data as shown in Figure 4.19. The gray region indicates where we do not have observations. In this experiments, we assume that the algorithm knows which pixels are missing. Since the curve evolution inside the region of missing data will not change the data-based energy term, the data driven force in that region would be zero. Hence, when we evolve the curve, the portion of the curve in the region of missing data will be evolved only by shape force whereas the other portion of the curve will be evolved by both the data force and the shape force.

We consider two cases for segmenting the digit image. The first is the case in which we know the class label *a priori* or can extract the class label by classifying

the test image based on the observed shape. In such a case, we can use the shape prior $p(C|L = l)$ conditioned on the known or estimated class label for shape-based segmentation. We then consider the second case in which we do not have the information about the class label and just use the marginal shape prior $p(C)$ for segmentation.

Segmentation of Handwritten Digit Images Aided by Digit Classification

We present examples where we extract the class label L and use the label to select a relevant shape prior $p(C|L = l)$ for segmenting a handwritten digit with missing data. To this end, we segment the test digit image without using shape priors until the curve captures all the portions of the object except the parts of the missing data and perform a classification based on the partial knowledge on the shape of the object. Once the class label is estimated, we use a shape prior conditioned on the class label.

We estimate the shape prior $p(C|L = l)$ for each $l = 0, 1, \ldots, 9$ from training shapes of that particular class. These conditional shape densities are used for both classification and segmentation. When estimating the shape prior, we use L_2 distance between signed distance functions for the experimental results in Figure 4.20 and Figure 4.21 and the template metric for the experimental results in Figure 4.22.

The results in Figure 4.20 and Figure 4.21 are obtained by using the shape priors with the L_2 distances but with different kernel sizes. In Figure 4.20, the kernel size is chosen to be the ML kernel size σ_{ML}, whereas in Figure 4.21 the kernel size is reduced to $0.5\sigma_{ML}$.

Figure 4.20(a) and Figure 4.21(a) show the images to be segmented with initial curves overlaid, and Figure 4.20(b) and Figure 4.21(b) show the intermediate stages of the curve evolution, where the curves capture all the portion of the object except the occluded parts. Up to the stages shown in (b), the curves are evolved without shape priors but with curve length regularization term. At that stage, we evaluate $p(C|L = l)$ for each l, where we use the density estimates with L_2 distance in Eqn.(4.21). Figure 4.20(c) and Figure 4.21(c) show bar graphs of $p(C|L = l)$ for $l = 0, 1, \ldots, 9$. Choosing the label which maximizes the likelihood $p(C|L = l)$ gives a label estimate. In Figure 4.20(c), we have a misclassification for the digit 8 estimating the class label to be "3", whereas in Figure 4.21(c), the ML labels were all correct. In general the probability of misclassification is nonzero, and if the label is misclassified the segmentation result will also be affected as is shown in the segmentation of digit 8 using $p(C|L = 3)$ as a shape prior in Figure 4.20(d). Once the class label is obtained, the curves in stage (b) are then evolved further with the conditional shape prior $p(C|L = l)$ with l being the ML label, and we have the final segmentation results in Figure 4.20(d) and Figure 4.21(d). Each of the final segmentation results is a typical shape in the sense that it is of high $p(C|L = l)$.

We can see that the likelihoods in Figure 4.21 are more distinguishing than those in Figure 4.20. This suggests that the ML kernel size may over-smooth the shape density estimate. Such over-smoothing may occur when the samples are sparse relative to the shape variation, i.e. shape variation is over wide range but we do not have enough

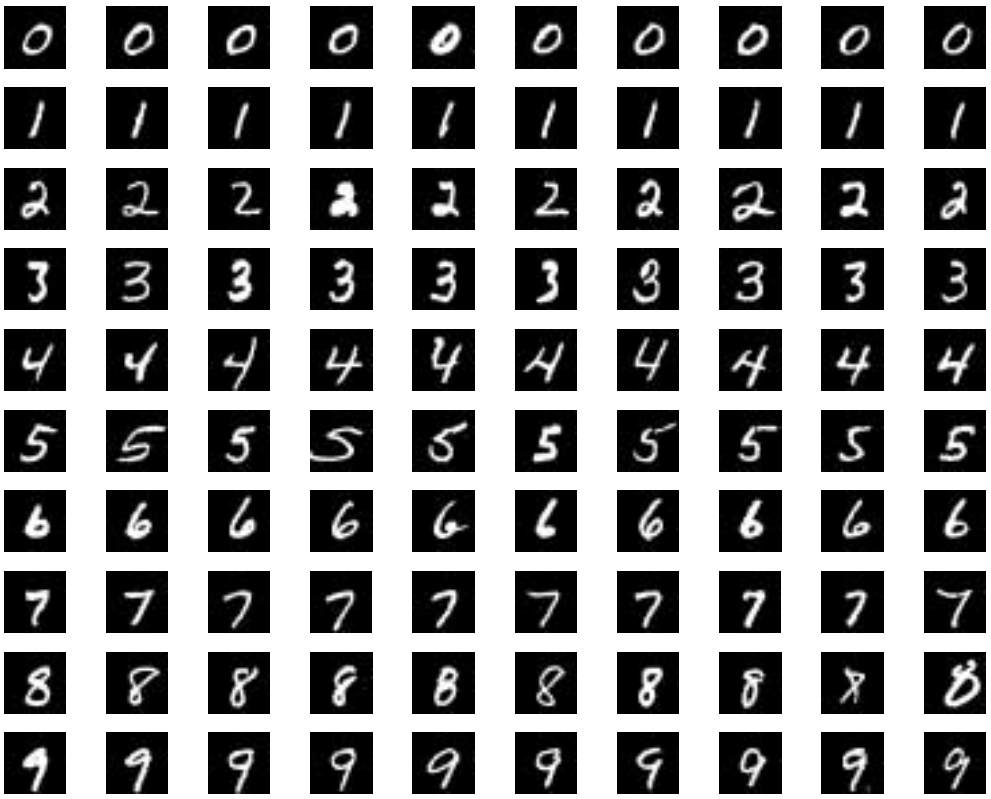


Figure 4.18. Training data for handwritten digits; Courtesy of Erik Miller

samples to model such shape variability. Regarding the segmentation results, the final curves in Figure 4.21(d) look better than those in Figure 4.20(d), e.g. digits 2 and 6. This again shows the effect of kernel size on the performance of the shape priors, and we think this issue deserves further research.

Figure 4.22 shows the results of the similar experiments, where we use the template metric for density estimation. The kernel size is chosen to be the ML kernel size. Comparing with Figure 4.20(c) and Figure 4.22(c), it seems that the template metric is more robust in classification than the L_2 distance. Comparing and analyzing the performance of these two classifiers deserve further study.

Segmentation of Handwritten Digit Images with Unlabeled Shape Prior

Now we consider the problem of segmenting the handwritten digit images with unknown class labels. This time, we just use the marginal shape prior $p(C) = \sum_l p(C|L = l)p(L = l)$ assuming that all the digit classes are equiprobable. Such a shape prior will have a broad range of support, and it will be multi-modal.

The marginal shape prior can be estimated by using a single kernel size σ for all the training samples $\mathcal{C}_i$ as in $p(C) = \frac{1}{n}\sum k(d(C, C_i), \sigma)$ or we can choose the kernel

Figure 4.19. Handwritten digits with missing data; each of these examples is not included in the training set in Figure 4.18. The parts of missing data are displayed in gray

size for each cluster of samples differently, which corresponds to estimating the lth mode $p(C|L=l)$ with its own kernel size σ_l and obtaining the marginal shape prior as $p(C) = \sum_l p(C|L=l)p(L=l)$. We first consider the case with the single kernel size.

Again we first evolve the curve without using the shape force until the curve captures all the portions of the object except the parts of missing data. After that, the shape force is turned on. We then take advantage of the weight factor $k(d_{L_2}(\tilde{\phi}, \tilde{\phi}_i), \sigma)$ in (4.47), which puts more emphasis on training shapes that are closer to $\tilde{\phi}$. Let us consider the ratio of such weights

$$\frac{k(d_i, \sigma)}{k(d_j, \sigma)} = \exp(-\frac{d_i^2 - d_j^2}{2\sigma^2}) \tag{4.54}$$

where $d_i = d_{L_2}(\tilde{\phi}, \tilde{\phi}_i)$ and $d_j = d_{L_2}(\tilde{\phi}, \tilde{\phi}_j)$. If $d_i < d_j$, the ratio increases as the kernel size σ decreases, indicating that the weights are more sensitive to differences in distances d_i's as the kernel size becomes smaller. As was discussed before, this weight function plays a role of emphasizing nearby training shapes in shape-based curve evolution. Choice of kernel size is an open issue and its choice depends on the application at hand [60]. Our choice of kernel size is $\sigma = \delta\sigma_{ML}$, a scaled version of the ML kernel size with scale parameter δ. This choice is based on the ML kernel size, since it can be automatically estimated from the data, and the scale parameter δ is chosen to be 0.1 or 0.2 in this application, in order to prevent over-smoothing across multiple clusters of samples.

Figure 4.23(a) shows the same test images as in Figure 4.21(a). Figure 4.23(b) shows segmentations without the shape prior. Then the shape force for the marginal shape prior is turned on and we have the final segmentation results in Figure 4.23(c). In order to check the class label of the final segmentation results in Figure 4.23(c), Figure 4.24(a) and Figure 4.24(b) show the bar graphs of likelihoods with the L_2 distance and the template metric respectively.

With the small kernel size, the weights in the shape force become very discriminating and thus behave as a built-in classifier attracting the segmenting curve to a correct mode of the multi-modal density. With highly discriminating weights, the shape force is dominated by the attraction force toward the training shape which is closest to the evolving curve.

Figure 4.25 shows the result for a similar experiment, when the shape prior is given in terms of the template metric. In this case, the ratio of the two weights is given by:

$$\frac{k(d_i,\sigma)d_i}{k(d_j,\sigma)d_j} = \frac{d_i}{d_j}\exp(-\frac{d_i^2-d_j^2}{2\sigma^2}) \tag{4.55}$$

where $d_i = d_T(C, C_i)$ and $d_j = d_T(C, C_j)$. Again, when $d_i < d_j$, the ratio increases as the kernel size σ decreases. The result in Figure 4.25 obtained for the kernel size $0.2\sigma_{ML}$ is as good as the result in Figure 4.22, where the label information is extracted by a classifier.

We now the consider the case in which we choose the kernel size for each cluster of samples differently. We choose the kernel size σ_l for each clusters of samples $l = 0, 1, \ldots, 9$. Again we use $\sigma_l = \delta\sigma_{l,ML}$, a scaled version of ML kernel sizes for each cluster $\sigma_{l,ML}$ with scaling factor δ. We tested with the scaling factor $\delta \in \{0.1, 0.2, \ldots, 1.0\}$ and show the best results in Figure 4.27 and Figure 4.29. We also show the classification results for these final segmentations in Figure 4.28 and Figure 4.30. For the shape prior with the L_2 distance, the scaling parameter 0.3 worked best. For the shape prior with the template metric, the scaling parameters $0.3, 0.4, 0.5$ worked equally well. Figure 4.29(c) contains results with a scaling parameter of 0.5.

It is expected that the ML kernel size σ_{ML} computed using samples of all the digits would be larger than ML kernel sizes $\sigma_{l,ML}$ computed using individual class of digits. We show the ratios $\frac{\sigma_{l,ML}}{\sigma_{ML}}$ in Table 4.1. The training data for the digits such as 0 and 1 have small shape variation and thus have smaller $\sigma_{l,ML}$ than other classes. The issue of kernel sizes is a topic for further analysis.

These experimental examples in Figure 4.23–Figure 4.30 demonstrate the power of our nonparametric shape prior framework in handling complicated, multi-modal densities in the space of shapes.

To conclude, our shape prior has a capacity to model the distributions of multi-class shapes. Consequently, the second segmentation method with the unlabeled shape prior is sufficient without any need for additional classifiers to select a conditional shape prior.

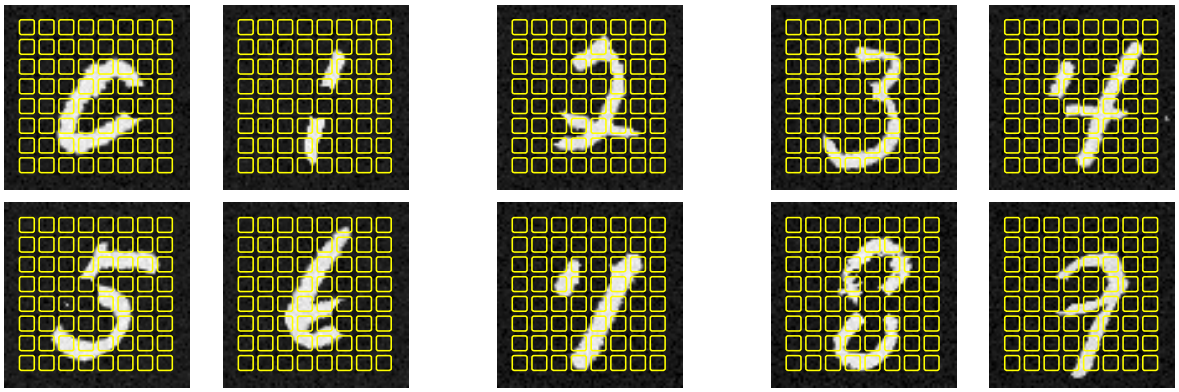
class label l	0	1	2	3	4
ratio $\frac{\sigma_{l,ML}}{\sigma_{ML}}$	0.2884	0.1845	0.7145	0.5532	0.7190
class label l	5	6	7	8	9
ratio $\frac{\sigma_{l,ML}}{\sigma_{ML}}$	0.6166	0.4530	0.5425	0.6990	0.4613

(a) with the L_2 distance

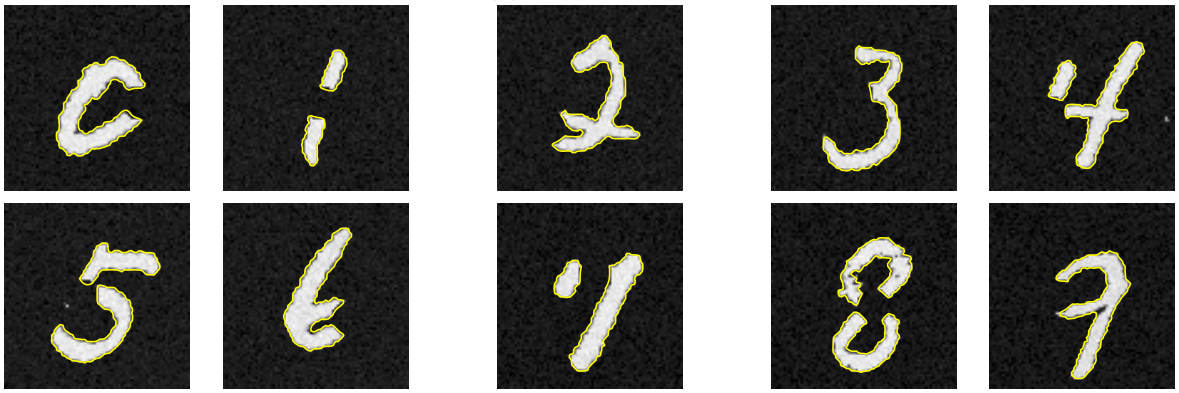
class label l	0	1	2	3	4
ratio $\frac{\sigma_{l,ML}}{\sigma_{ML}}$	0.3230	0.1107	0.6624	0.5405	0.4975
class label l	5	6	7	8	9
ratio $\frac{\sigma_{l,ML}}{\sigma_{ML}}$	0.6790	0.4589	0.3333	0.8343	0.5119

(b) with the template metric

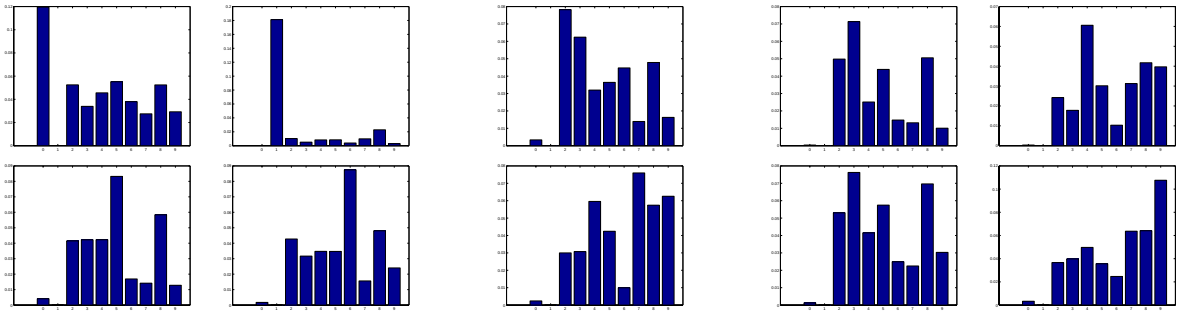
Table 4.1. The ratio $\frac{\sigma_{l,ML}}{\sigma_{ML}}$ for the density estimate: (a) with the L_2 distance; (b) with the template metric.



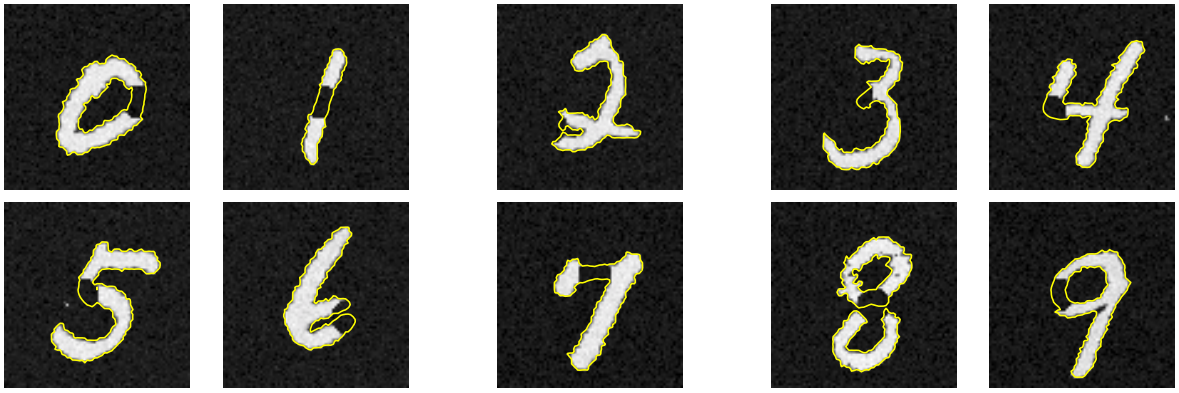
(a) initial curves



(b) intermediate curves

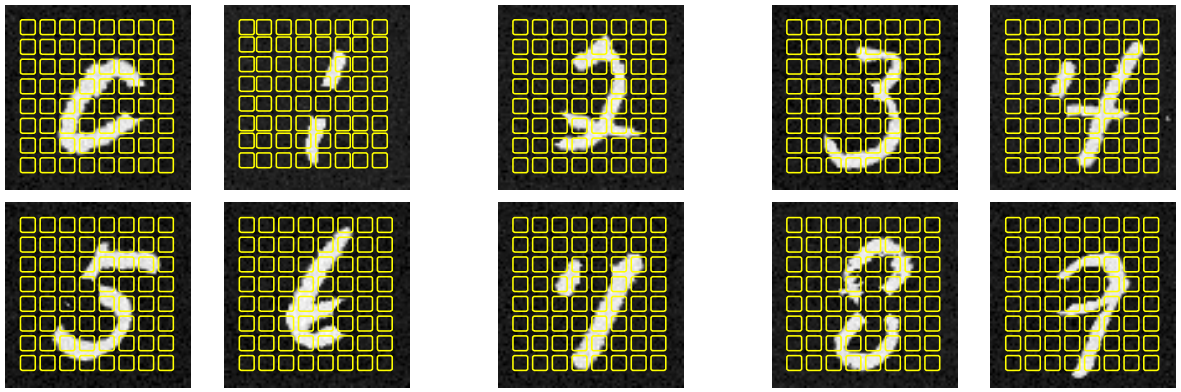


(c) likelihoods

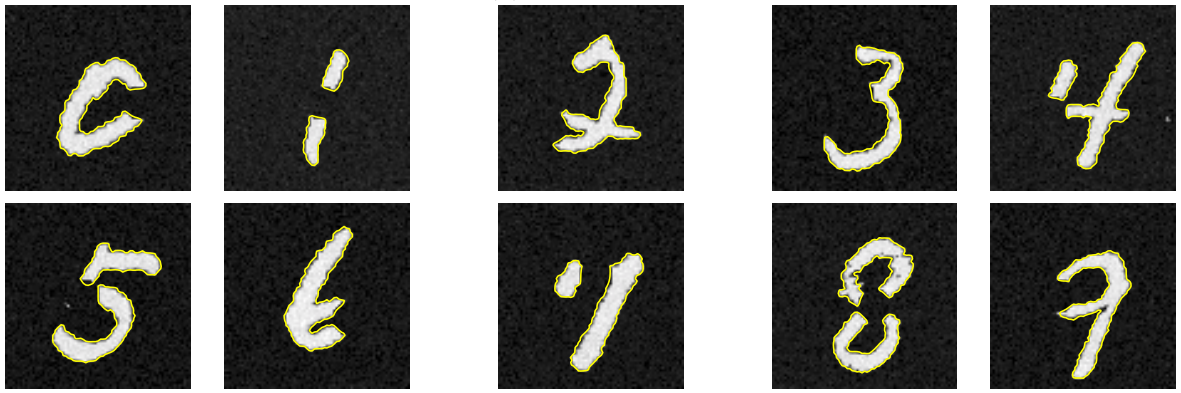


(d) final curves

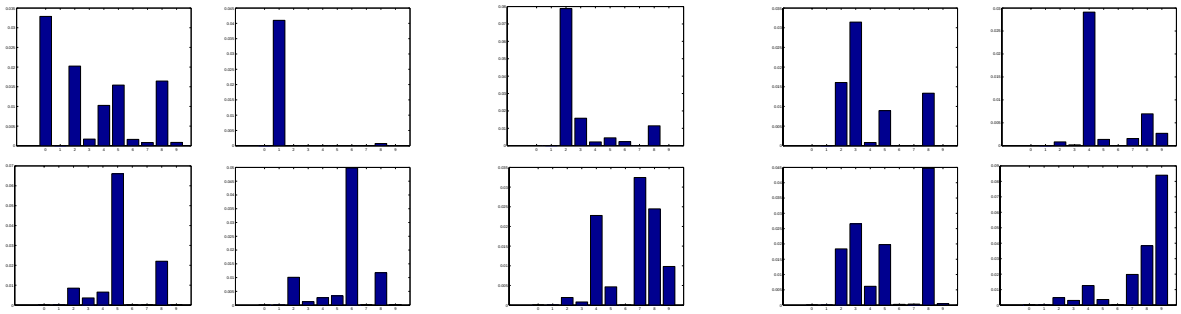
Figure 4.20. Segmentation of handwritten digits with missing data using a classifier to determine the shape prior to employ. The L_2 metric is used for shape priors. The kernel size is chosen to be $\sigma = \sigma_{ML}$.



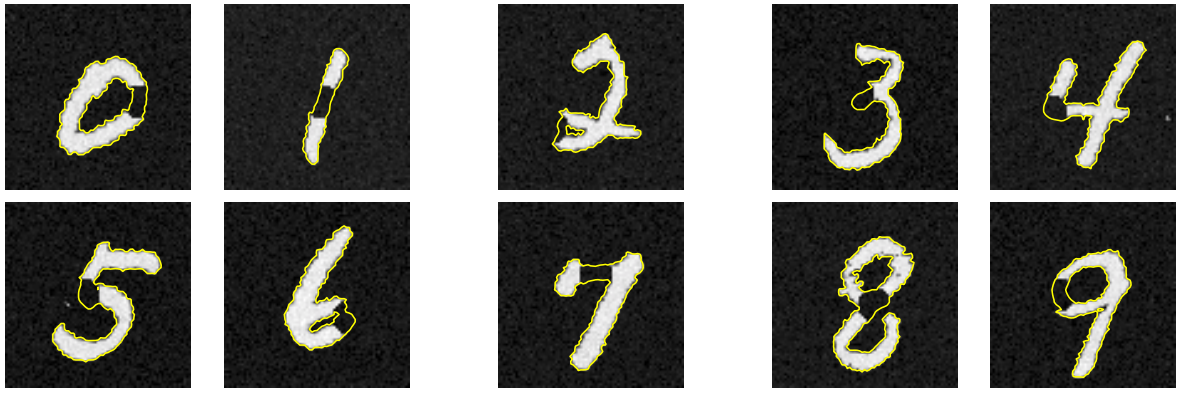
(a) initial curves



(b) intermediate curves

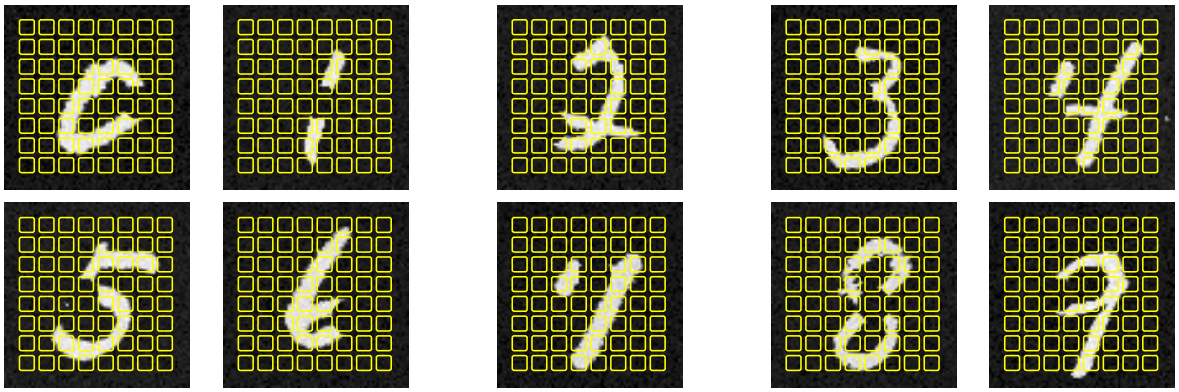


(c) likelihoods



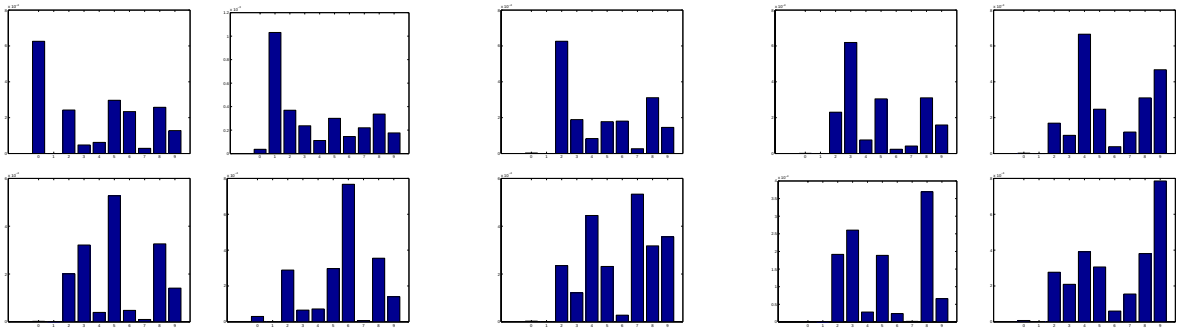
(d) final curves

Figure 4.21. Segmentation of handwritten digits with missing data using a classifier to determine the shape prior to employ. The L_2 metric is used for shape priors. The kernel size is $0.5\sigma_{ML}$.

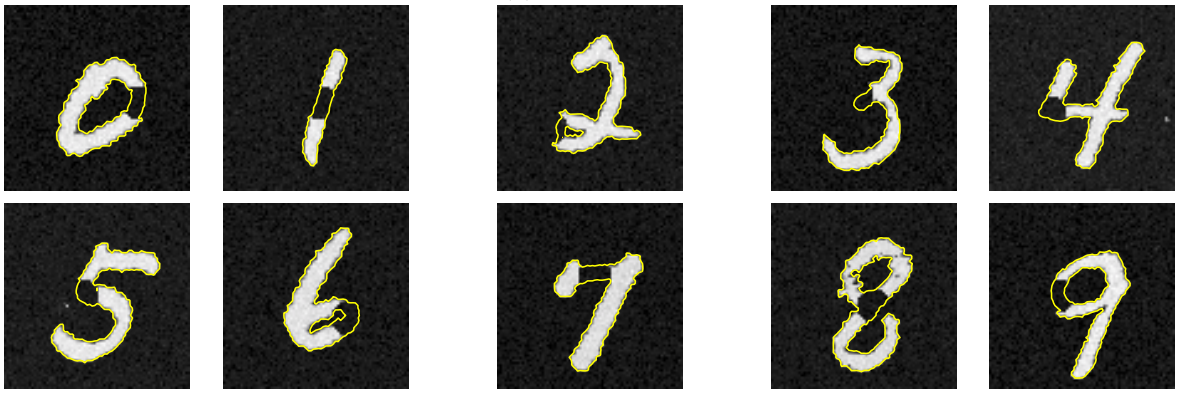


(a) initial curves

(b) intermediate curves

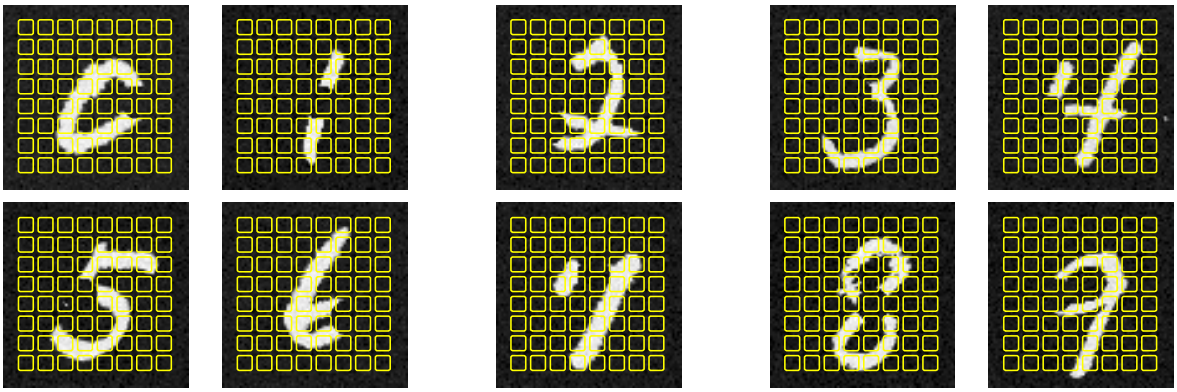


(c) likelihoods

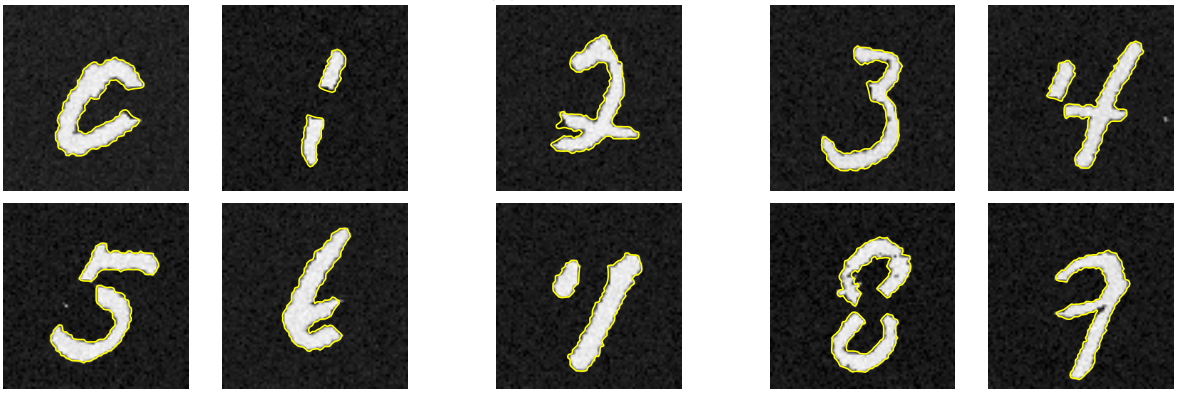


(d) final curves

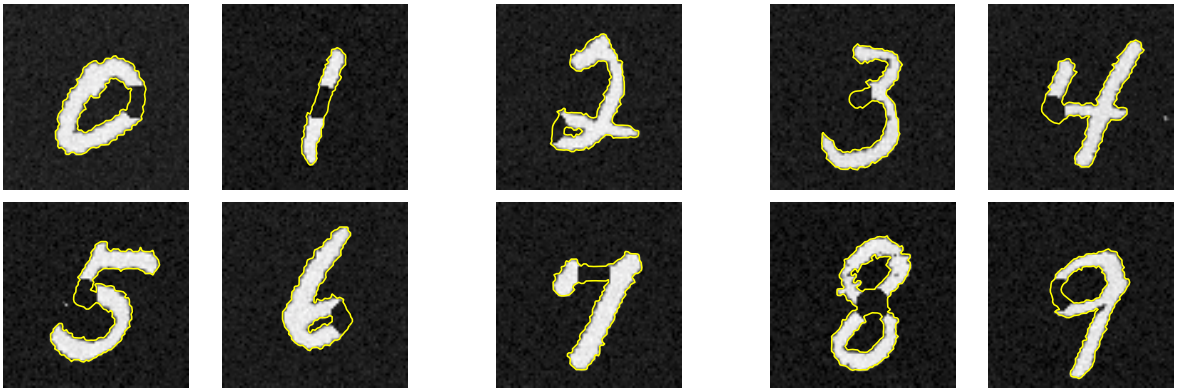
Figure 4.22. Segmentation of handwritten digits with missing data using a classifier to determine the shape prior to employ. The template metric is used for shape priors. The kernel size is σ_{ML}.



(a) initial curves



(b) intermediate curves



(c) final curves

Figure 4.23. Segmentation of handwritten digit with missing data using an unlabeled prior density $p(C)$. The L_2 metric is used for shape priors. The kernel size is $0.1\sigma_{ML}$.

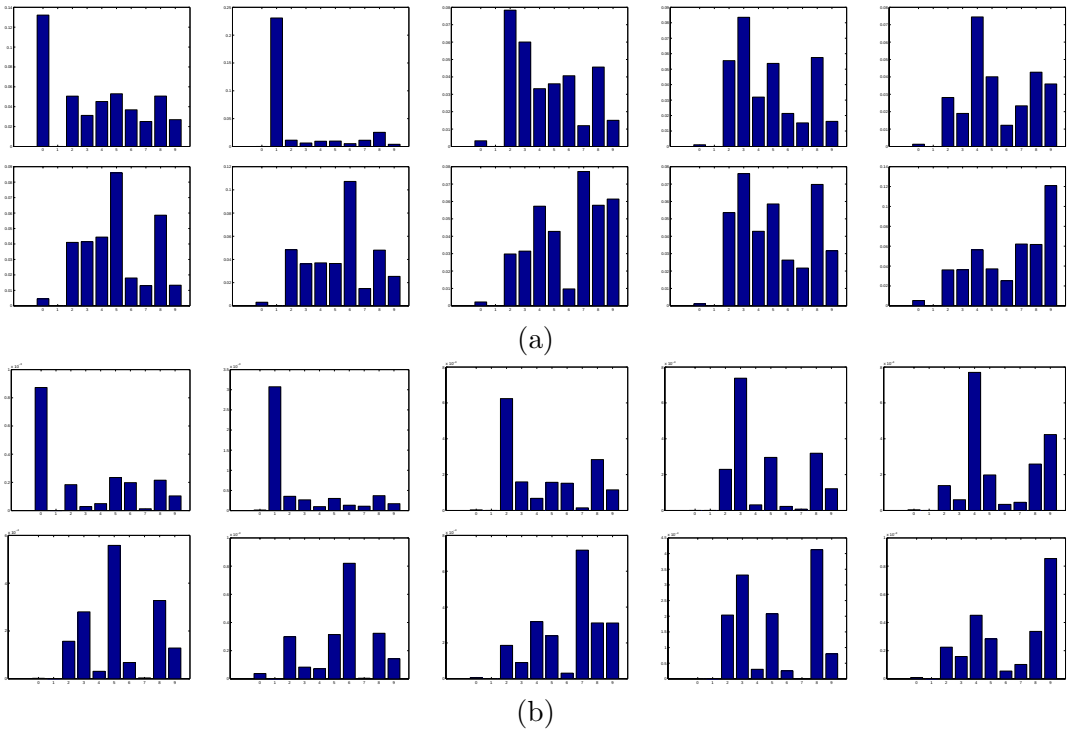
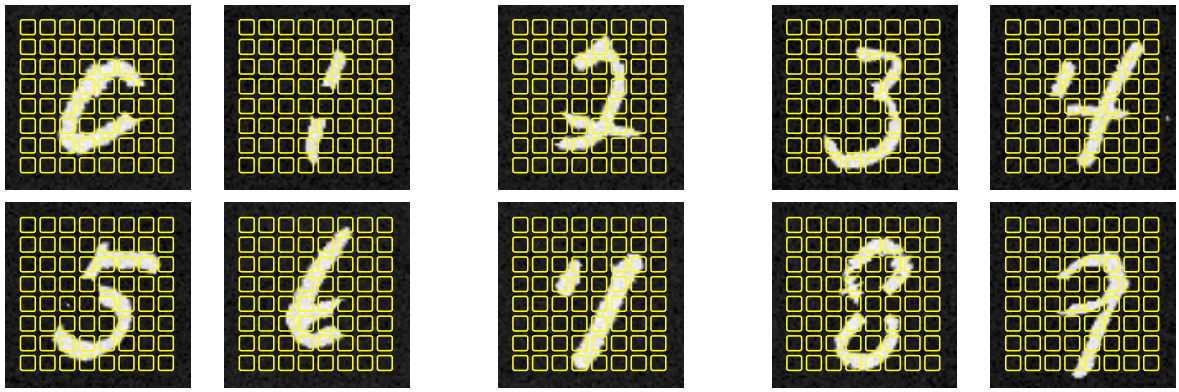
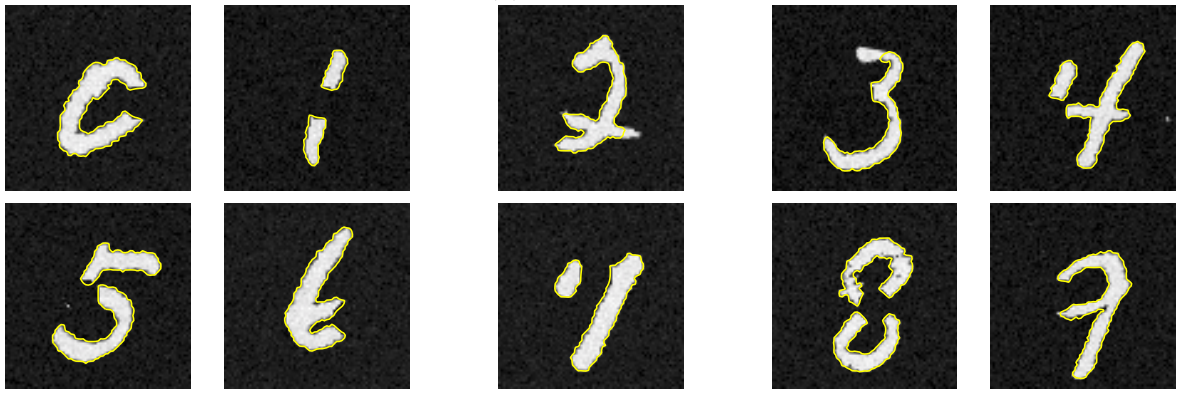


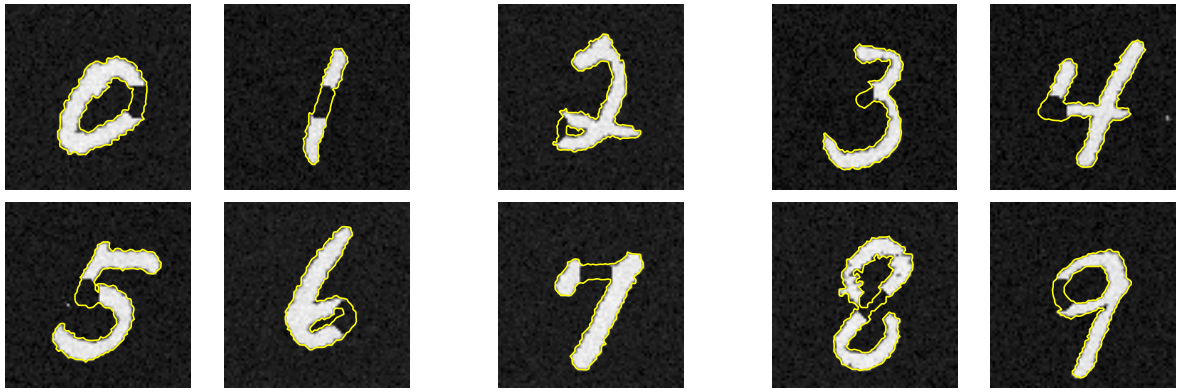
Figure 4.24. Bar graphs of likelihoods for classification of the final segmentation result in Figure 4.23. (a) likelihoods $p(C|L = l)$ with the L_2 distance with the ML kernel size. (b) likelihoods $p(C|L = l)$ with the template metric with the ML kernel size.



(a) initial curves



(b) intermediate curves



(c) final curves

Figure 4.25. Segmentation of handwritten digit with missing data using an unlabeled prior density $p(C)$ with template metric. The kernel size is $0.2\sigma_{ML}$.

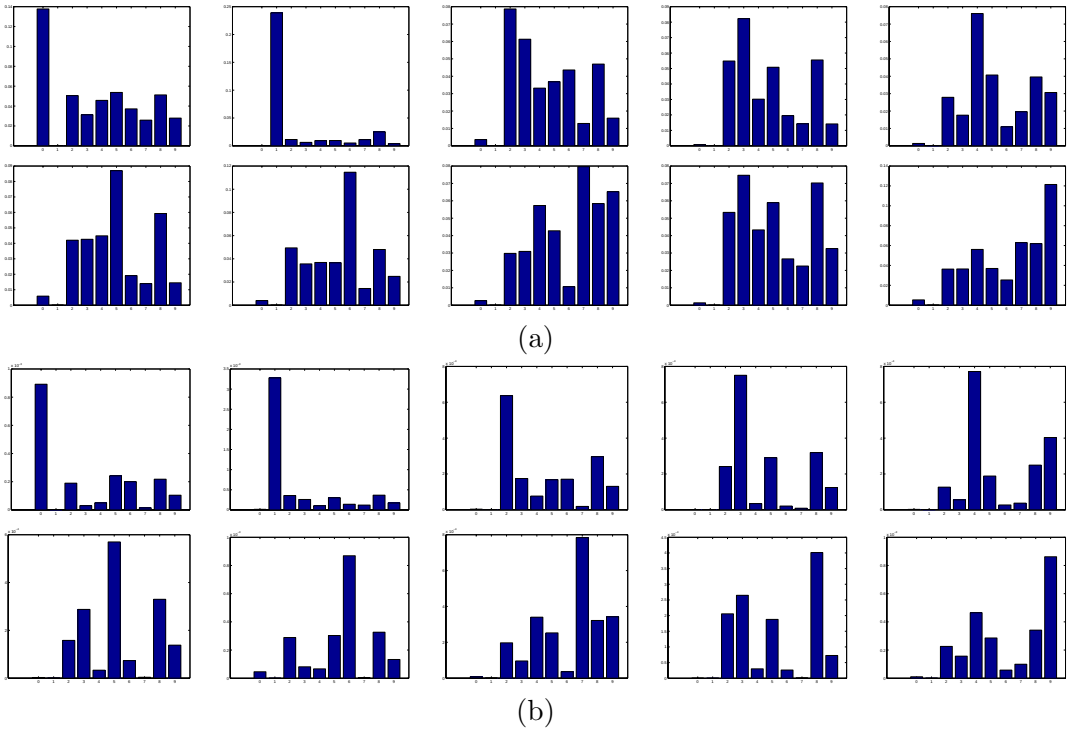
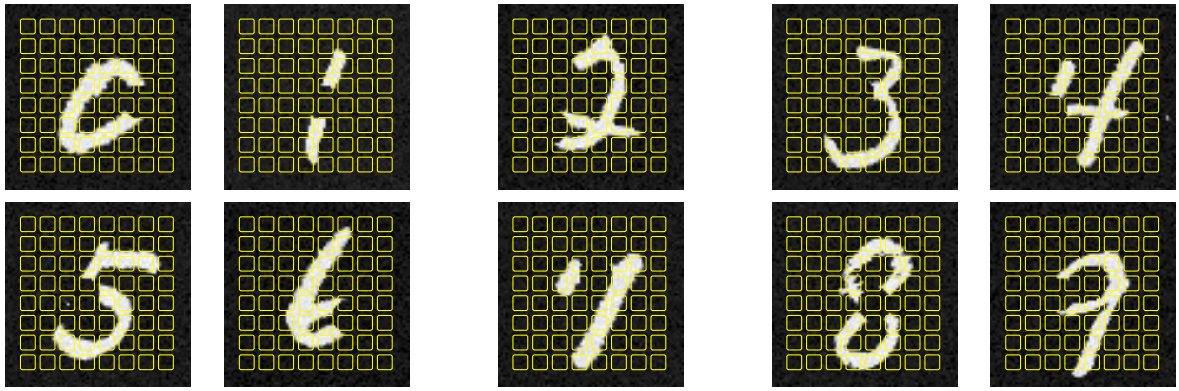
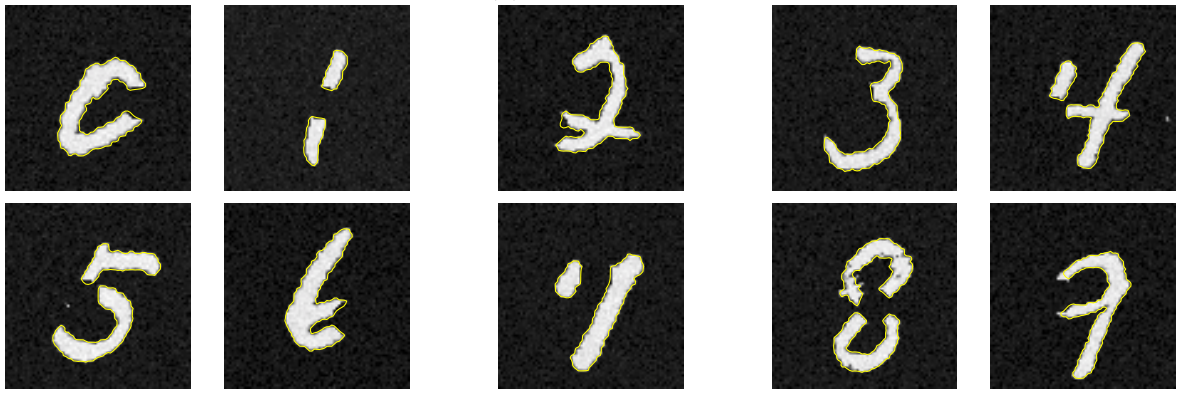


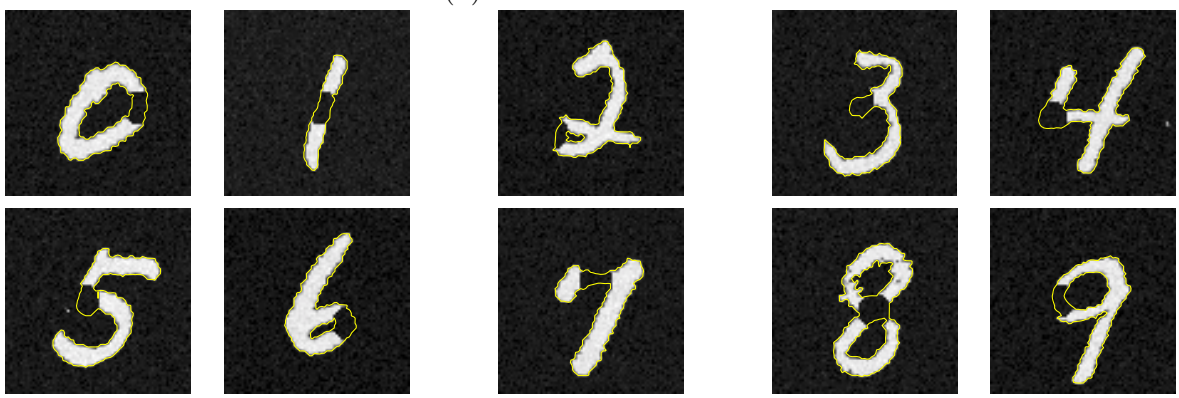
Figure 4.26. Bar graphs of likelihoods for classification of the final segmentation result in Figure 4.25. (a) likelihoods $p(C|L = l)$ with the L_2 distance with the ML kernel size $\sigma_{l,ML}$. (b) likelihoods $p(C|L = l)$ with the template metric with the ML kernel size $\sigma_{l,ML}$.



(a) initial curves



(b) intermediate curves



(c) final curves

Figure 4.27. Segmentation of handwritten digit with missing data using an unlabeled prior density $p(C)$ with the L_2 distance with different kernel sizes for each mode, where the kernel size for lth mode is given by $\sigma_l = 0.3\sigma_{l,ML}$.

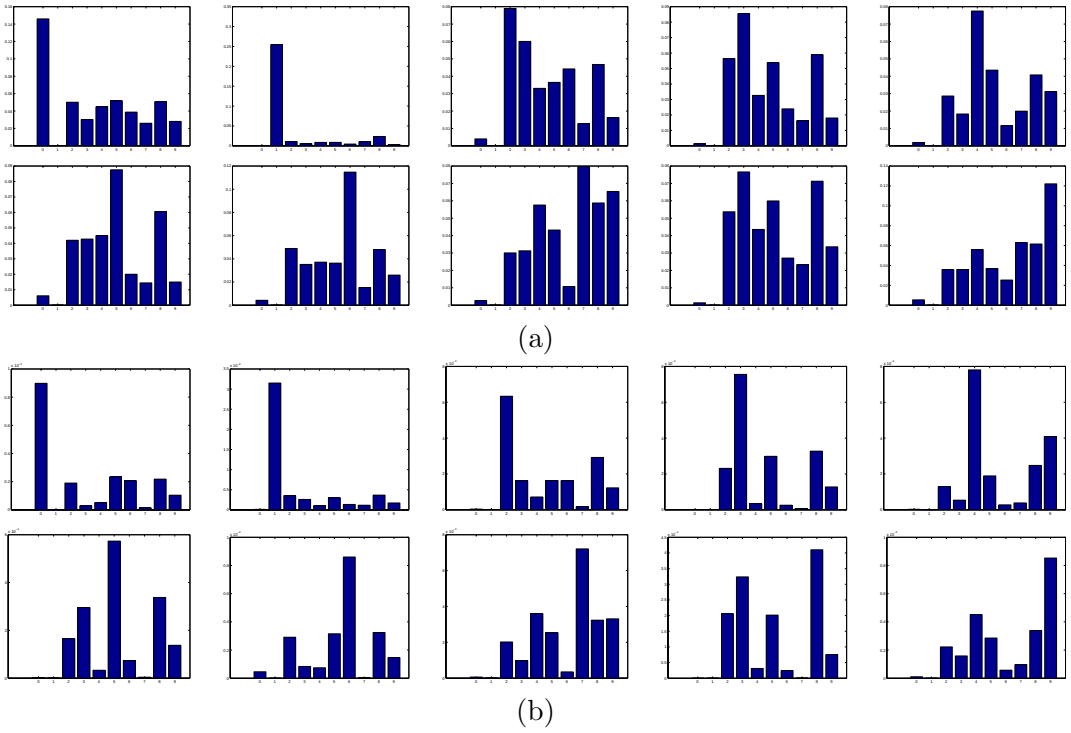
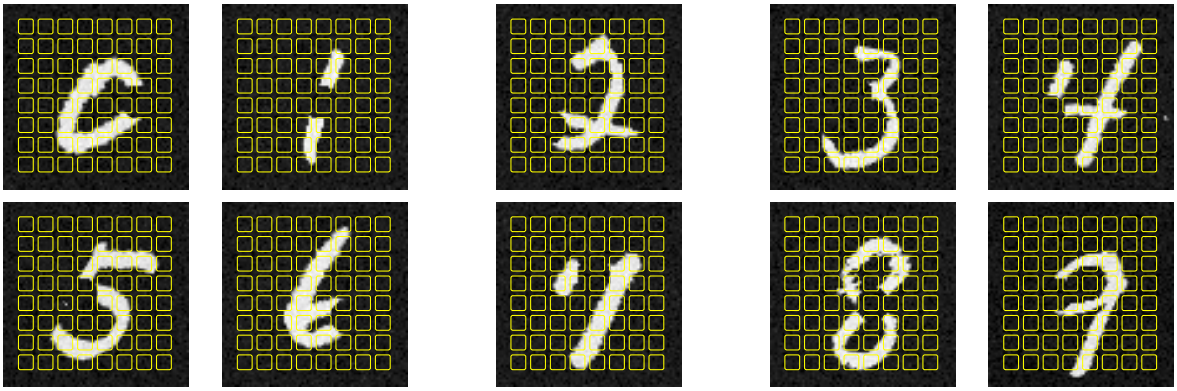
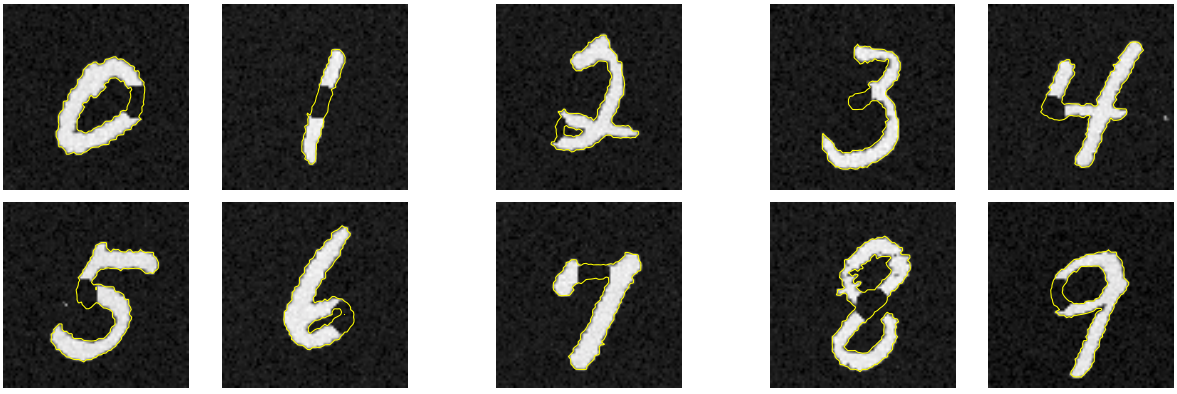


Figure 4.28. Bar graphs of likelihoods for classification of the final segmentation result in Figure 4.27. (a) likelihoods $p(C|L = l)$ with the L_2 distance with the ML kernel size $\sigma_{l,ML}$. (b) likelihoods $p(C|L = l)$ with the template metric with the ML kernel size $\sigma_{l,ML}$.



(a) initial curves

(b) intermediate curves



(c) final curves

Figure 4.29. Segmentation of handwritten digit with missing data using an unlabeled prior density $p(C)$ with the template metric with different kernel sizes for each mode, where the kernel size for lth mode is given by $\sigma_l = 0.5\sigma_{l,ML}$.

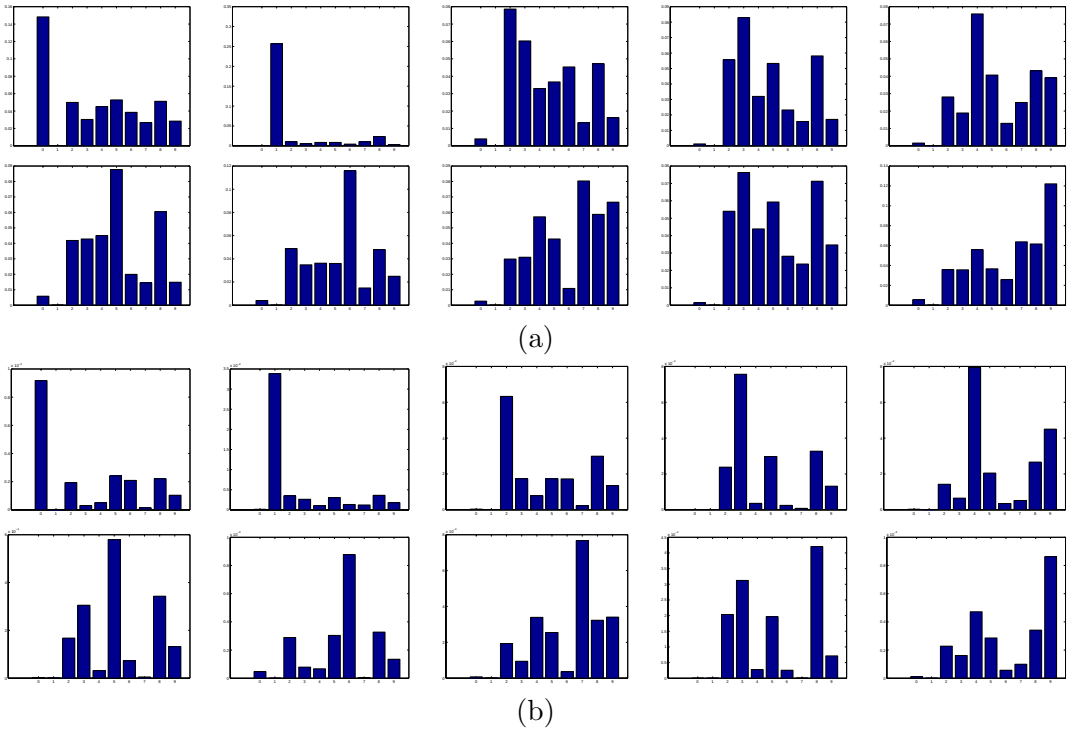


Figure 4.30. Bar graphs of likelihoods for classification of the final segmentation result in Figure 4.29. (a) likelihoods $p(C|L = l)$ with the L_2 distance with the ML kernel size $\sigma_{l,ML}$. (b) likelihoods $p(C|L = l)$ with the template metric with the ML kernel size $\sigma_{l,ML}$.

Chapter 5

Contributions and Suggestions

In this thesis, we have developed nonparametric statistical methods for image segmentation and shape analysis. We have proposed a nonparametric information-theoretic image segmentation method that can segment a large class of images. We have also proposed nonparametric shape priors for modeling shape distributions and developed a shape-based image segmentation method. We summarize the contributions of this thesis in Section 5.1 and provide several related topics for future research in Section 5.2.

■ 5.1 Summary and Contributions

The major contributions of this thesis are summarized as follows.

A Nonparametric Information-Theoretic Method for Image Segmentation

We have developed a new information-theoretic image segmentation method based on nonparametric statistics and curve evolution. We have formulated the segmentation problem as one of maximizing the mutual information between the region labels and the pixel intensities, subject to curve length constraints. We have considered a class of images where each region has a distinct intensity distribution. For such class of images, the MI has the property that it is maximized if and only if the label gives the correct segmentation. We have derived the curve evolution equations for the optimization problem posed in our framework. The resulting curve evolution equation can be interpreted as a nonparametric region competition based on a likelihood ratio test. Due to the nonparametric aspect of our formulation, the proposed technique can automatically deal with a variety of segmentation problems, in which many currently available curve evolution-based techniques would either completely fail or at least require the *a priori* extraction of representative statistics for each region. We have also extended our method to problems involving more than two regions, where we evolve multiple curves by a nonparametric region competition. Our experimental results have shown the strength of the proposed technique in accurately segmenting real and synthetic images.

Nonparametric Shape Priors

We have addressed the problem of estimating shape prior densities from example shapes and developed a shape-based segmentation method. In particular, we have developed a framework for estimating shape priors from training shapes in a nonparametric way, and based on such nonparametric shape priors, we have formulated the shape-based segmentation problem as a maximum a posteriori estimation problem. Evaluation of the nonparametric shape prior for a candidate curve for segmentation is given in terms of distances between the candidate curve and the training curves. In this thesis, we considered the template metric and the L_2 distance between signed distance functions, but other metrics can also be used for nonparametric shape priors. We have derived a curve evolution equation (or level set evolution equation) based on the nonparametric shape priors. The curve evolution due to the shape prior is given as a weighted average of forces, where each force tries to warp the current shape toward an example shape such that the metric between the two shapes is decreased. We have presented experimental results of segmenting partially-occluded images. We have considered the case in which the training shapes form multiple clusters. Our nonparametric shape priors model such shape distributions successfully without requiring prior knowledge on the number of clusters, whereas applying existing PCA-based approaches in such scenarios for shape-based segmentation would require such prior knowledge.

■ 5.2 Open Problems and Suggestions for Future Research

We suggest several ways to extend the framework developed in this thesis. In particular, our nonparametric information theoretic segmentation method can be extended to the case of vector-valued images by using multi-dimensional density estimates. The mutual information measure we proposed can be a useful stopping criterion in hierarchical segmentation. We discuss some open issues with the problem of segmenting images with spatially varying distribution.

The nonparametric shape priors can be extended by incorporating more sophisticated distance metric such as the distance of the geodesic path connecting two signed distance functions. We also briefly mention several open problems associated with the geodesic distances.

Segmentation of Vector-Valued Images and Textured Images

Our nonparametric method was developed for segmentation of gray-level images, where we used a one-dimensional Parzen density estimator. This nonparametric method can be extended to the case of vector-valued images such as color images by using multi-dimensional Parzen density estimators. For this case, the energy functional and the gradient flow will remain the same except that the density estimates are now multi-dimensional.

Our method can also be extended for textured image segmentation[1]. Previous methods for textured image segmentation [51] often use filter banks in order to extract features to distinguish different textures. The above extension to vector-valued images can also be applied to outputs of filter banks, thereby producing a segmentation scheme for textured images.

Hierarchical Segmentation

We proposed a nonparametric segmentation method for multi-region segmentation using multiple curves. An alternative approach to multi-region segmentation is hierarchical segmentation, which first segments the image into two broad regions and further segments each region if the region has subregions to segment. For instance, a human face can be first segmented out of an image then subregions such as eyes, a nose, and a mouth can be segmented subsequently. One merit of hierarchical segmentation is that it gives a tree-like structure that describes the relation between parts and whole specifying the super-region to which each subregion belongs.

The mutual information measure described in Chapter 3 can be useful in deciding whether each region should be further segmented or not, since the MI can measure the likelihood that a region has distinct subregions (discussed in Section 3.4 and Section B.2.1). When a region is further segmented into subregions, the MI can be used as a criterion for deciding whether it is an over-segmentation or not. This MI criterion could also be used together with a model order selection criterion such as the Akaike Information Criterion [3] to decide when to stop the hierarchical segmentation process.

Segmentation of Images with Spatially Varying Distribution

Region-based segmentation methods are robust to high-frequency noise since they do not rely on edge functions or image gradients. However, a low frequency artifact such as illumination variation over image regions can affect the region statistics and hence the segmentation results. Hence, an interesting direction for future work would be to consider the problem of segmenting images whose intensity distribution is spatially-varying. In order to model an image with a spatially-varying intensity distribution, we need to model intensities in each region as a non-stationary random-field (stochastic process). Extension of our MI-based method to deal explicitly with such distributions is an open research problem.

Parzen Shape Density Estimates with Other Distance Metrics

In Chapter 4, we formulated the Parzen shape density estimates with the L_2 distance between signed distance functions and the template metric. We also derived the corresponding gradient flow for curve evolution.

[1]Note that although the method we developed can segment some textured images (as demonstrated in Chapter 3), it does so by exploiting the first order pdfs of intensities rather than the textures.

Similarly, we can formulate the Parzen shape density estimates in terms of other metrics such as the Hausdorff metric or the transport metric (Monge-Kantorovich) mentioned in Section 2.3.2. The Hausdorff metric is easy to evaluate, but it is not differentiable w.r.t. shapes since it is an L_∞ type metric. Recently, Charpiat et al. [12] have proposed an approximation of the Hausdorff metric in order to make it differentiable and used a gradient of the approximate Hausdorff metric to warp one shape into another shape. Since the gradient flow for the Parzen shape density estimate is given in terms of linear combination of shape forces for such warpings, the curve evolution for Parzen shape density with the approximate Hausdorff metric can be implemented using their methods.

Unlike the aforementioned metrics, the evaluation of transport metric is not straightforward, since it involves an optimization of a work functional over mass preserving mappings. A relevant problem is to develop an efficient numerical scheme for evaluating the transport metric and a curve evolution method for warping shapes based on the transport metric.

Open Problems about Geodesic Distance between Signed Distance Functions

In Chapter 4, we introduced the space of signed distance functions $\mathcal{D}$ as a subspace of the infinite dimensional Hilbert space $\mathcal{L}$ and mentioned the notion of the geodesic distance between two signed distance functions ϕ_1 and ϕ_2. There are several related open problems.

The Space of Signed Distance Functions

In Section 4.2.2, we assumed that the space of signed distance functions or the manifold of signed distance functions is not too curved and that the manifold is locally flat. We can formulate this assumption in the form of the following conjecture. To verify this conjecture is a problem worthwhile for future research.

Conjecture:
For any constant $c > 1$, there exists $\delta(c)$ such that if two signed distance functions ϕ_1 and ϕ_2 satisfy $d_{L_2}(\phi_1, \phi_2) < \delta(c)$, we have

$$\frac{d_{\text{geodesic}}(\phi_1, \phi_2)}{d_{L_2}(\phi_1, \phi_2)} < c. \tag{5.1}$$

In the above conjecture, the ratio $\frac{d_{\text{geodesic}}(\phi_1,\phi_2)}{d_{L_2}(\phi_1,\phi_2)}$ is a degree of mismatch between the geodesic distance and the L_2 distance, and the upper bound $c > 1$ on the ratio $\frac{d_{\text{geodesic}}(\phi_1,\phi_2)}{d_{L_2}(\phi_1,\phi_2)}$ is a degree of flatness of the part of manifold with diameter $\delta(c)$. This conjecture implies that for any desired level of flatness c, we can find a diameter $\delta(c)$ such that any open ball in the manifold with diameter $\delta(c)$ has the desired level of flatness.

Computing the Geodesic Distances

One way to compute the distance of the minimum geodesic path $\{\phi(t)|t \in [0,T]\}$ connecting two signed distance functions ϕ_0 and ϕ_T is to obtain a finite number of samples $\{\phi_0, \phi_1 \ldots, \phi_n = \phi_T\}$ along the geodesic path and approximating the geodesic distance $d_{\text{geodesic}}(\phi_0, \phi_T)$ by

$$\hat{d}_{\text{geodesic}}(\phi_0, \phi_T) = \sum_{i=0}^{n-1} d_{L_2}(\phi_i, \phi_{i+1}) \tag{5.2}$$

If the above conjecture is true and the samples are dense enough satisfying $d_{L_2}(\phi_i, \phi_{i+1}) \leq \delta(c)$, the geodesic distance is bounded as follows

$$\hat{d}_{\text{geodesic}}(\phi_0, \phi_T) \leq d_{\text{geodesic}}(\phi_0, \phi_T) \leq c \cdot \hat{d}_{\text{geodesic}}(\phi_0, \phi_T) \tag{5.3}$$

Note that finding such samples on the geodesic path is an open problem.

Computing the Gradient Flow for Geodesic Distance

Finally, computing gradient flow $\frac{\partial \phi}{\partial t}$ for minimizing $d_{\text{geodesic}}(\phi, \phi_1)$ for a fixed ϕ_1 is an important open problem. If one can compute the gradient flow, we can warp a signed distance function ϕ toward ϕ_1 along the minimum geodesic path. Furthermore, the gradient flow for the Parzen shape density estimate will be given as a linear combination of the forces that warp ϕ toward ϕ_i along the minimum geodesic path.

Appendix A

First Variation of Simple Region Integrals

We consider the gradient flow of $\vec{C}$ with respect to a simple region integral whose integrand f is a function that does *not depend on* $\vec{C}$.

$$E(\vec{C}) = \int_R f(\mathbf{x})\, d\mathbf{x} \; = \oint_{\vec{C}} \langle F, \vec{N} \rangle \, ds$$

where $\vec{N}$ denotes the unit normal of $\vec{C}$, ds is the Euclidean arclength element, and $F(\mathbf{x})$ is a vector field chosen so that $\nabla \cdot F(\mathbf{x}) = f(\mathbf{x})$. The equivalence between the region integral based on f and the contour integral based on F follows from the divergence theorem.

We start by considering a fixed parameterization $p \in [0,1]$ of the curve $\vec{C}$ which does not vary as the curve evolves in time t so that (p,t) comprise independent variables. By a change of variables, we may rewrite E as follows

$$E(\vec{C}) = \int_0^1 \langle F, J\vec{C}_p \rangle \, dp,$$

where $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ denotes a -90° rotation matrix. Differentiating with respect to t yields

$$\frac{dE}{dt} = \int_0^1 \left\langle \frac{dF}{d\mathbf{x}}\vec{C}_t, J\vec{C}_p \right\rangle + \left\langle F, J\vec{C}_{pt} \right\rangle \, dp = \int_0^1 \left\langle \frac{dF}{d\mathbf{x}}\vec{C}_t, J\vec{C}_p \right\rangle - \left\langle \frac{dF}{d\mathbf{x}}\vec{C}_p, J\vec{C}_t \right\rangle \, dp$$

where the equality follows via integration by parts and where $\frac{dF}{d\mathbf{x}}$ denotes the Jacobian matrix of F with respect to $\mathbf{x}$. Rearranging terms leads to

$$\frac{dE}{dt} = \int_0^1 \left\langle \vec{C}_t, \left[\left(J^T \frac{dF}{d\mathbf{x}} \right)^T - \left(J^T \frac{dF}{d\mathbf{x}} \right) \right] \vec{C}_p \right\rangle \, dp = \oint_{\vec{C}} \left\langle \vec{C}_t, f\vec{N} \right\rangle \, ds$$

from which the form of the gradient flow for $\vec{C}$ (the negative of the gradient so that the region integral decreases most rapidly) is revealed to be

$$\frac{\partial \vec{C}}{\partial t} = -f\vec{N}.$$

Thus the flow depends only upon f (as we would expect), not upon our particular choice for F.

Appendix B

Proofs and Derivations for Chapter 3

B.1 Proof of the Statement about Mutual Information from Section 3.1

In this appendix, we prove a statement from Section 3.1, namely that the mutual information $I(G(X); L_{\vec{C}}(X))$ is maximized if and only if $\vec{C}$ is the correct segmentation, i.e. if $R_+ = R_1$, $R_- = R_2$ (or equivalently $R_+ = R_2$, $R_- = R_1$). We remind the readers that this analysis makes use of the knowledge of R_1, R_2, p_1, p_2 so that we can compute the MI. Since $I(G(X); X)$ is independent of the label $L_{\vec{C}}(\cdot)$, it is sufficient to show that

$$I(G(X); L_{\vec{C}}(X)) \leq I(G(X); X) \tag{B.1}$$

and that equality holds if and only if $R_+ = R_1$, $R_- = R_2$ (or equivalently $R_+ = R_2$, $R_- = R_1$).

Proof. The inequality is basically the data processing inequality [16]. We will follow the proof in [16].

By using the chain rule, we can expand the mutual information between $G(X)$ and $\{X, L_{\vec{C}}(X)\}$, namely $I(G(X); X, L_{\vec{C}}(X))$ in the following two different ways:

$$\begin{aligned} I(G(X); X, L_{\vec{C}}(X)) &= I(G(X); L_{\vec{C}}(X)) + I(G(X); X | L_{\vec{C}}(X)) & \text{(B.2)} \\ &= I(G(X); X) + I(G(X); L_{\vec{C}}(X) | X) & \text{(B.3)} \end{aligned}$$

Note that given $X = x$, $L_{\vec{C}}(X)$ is just a constant $L_{\vec{C}}(x)$. Thus $G(X)$ and $L_{\vec{C}}(X)$ are conditionally independent given X, and we have $I(G(X); L_{\vec{C}}(X)|X) = 0$. Since $I(G(X); X|L_{\vec{C}}(X)) \geq 0$, we have

$$I(G(X); X) \geq I(G(X); L_{\vec{C}}(X)). \tag{B.4}$$

The equality holds if and only if $I(G(X); X|L_{\vec{C}}(X)) = 0$, i.e. $G(X)$ and X are conditionally independent given $L_{\vec{C}}(X)$. Now it suffices to show that $p_{G(X)|L_{\vec{C}}(X)} = p_{G(X)|X,L_{\vec{C}}(X)}$ if and only if $R_+ = R_1$, $R_- = R_2$ (or equivalently $R_+ = R_2$, $R_- = R_1$). The remainder of the proof is based on the fact that $p_{G(X)|L_{\vec{C}}(X)}$ is not homogeneous, (i.e. it is a

mixture of p_1 and p_2) unless $L_{\vec{C}}(\cdot)$ gives a correct segmentation, whereas $p_{G(X)|X,L_{\vec{C}}(X)}$ is always homogeneous.

Note that the conditional densities $p_{G(X)|L_{\vec{C}}(X)=L_+}$ and $p_{G(X)|L_{\vec{C}}(X)=L_-}$ are mixtures of p_1 and p_2 as given in (3.6) and (3.7):

$$p_{G(X)|L_{\vec{C}}(X)=L_+} = \frac{|R_+ \cap R_1|}{|R_+|}p_1 + \frac{|R_+ \cap R_2|}{|R_+|}p_2 \tag{B.5}$$

$$p_{G(X)|L_{\vec{C}}(X)=L_-} = \frac{|R_- \cap R_1|}{|R_-|}p_1 + \frac{|R_- \cap R_2|}{|R_-|}p_2 \tag{B.6}$$

On the other hand, the conditional density $p_{G(X)|X=x,L_{\vec{C}}(X)=L_{\vec{C}}(x)}$ is p_1 if $x \in R_1$ and p_2 if $x \in R_2$.

Suppose that $R_+ = R_1$, $R_- = R_2$. Then (B.5) and (B.6) give us that $p_{G(X)|L_{\vec{C}}(X)=L_+} = p_1$ and $p_{G(X)|L_{\vec{C}}(X)=L_-} = p_2$. Similarly, if $R_+ = R_2$, $R_- = R_1$, then $p_{G(X)|L_{\vec{C}}(X)=L_+} = p_2$ and $p_{G(X)|L_{\vec{C}}(X)=L_-} = p_1$. In either case, we have $p_{G(X)|L_{\vec{C}}(X)} = p_{G(X)|X,L_{\vec{C}}(X)}$.

However, unless $R_+ = R_1$, $R_- = R_2$ (or equivalently $R_+ = R_2$, $R_- = R_1$), at least one of $p_{G(X)|L_{\vec{C}}(X)=L_+}$ and $p_{G(X)|L_{\vec{C}}(X)=L_-}$ is a mixture of p_1 and p_2, thus $p_{G(X)|L_{\vec{C}}(X)} \neq p_{G(X)|X,L_{\vec{C}}(X)}$.

Therefore, $p_{G(X)|L_{\vec{C}}(X)} = p_{G(X)|X,L_{\vec{C}}(X)}$ if and only if $R_+ = R_1$, $R_- = R_2$ (or equivalently $R_+ = R_2$, $R_- = R_1$), and this completes the proof.

■

Remark: The inequality (B.1) is also true for the case where $L_{\mathbf{C}}(\cdot)$ is an n-ary label, and the equality holds if and only if $p_{G(X)|L_{\mathbf{C}}(X)} = p_{G(X)|X,L_{\mathbf{C}}(X)}$. Consequently, the equality holds if the label $L_{\mathbf{C}}(\cdot)$ gives a correct segmentation. Now we prove that the equality does not hold if the label gives an incorrect segmentation. Since $p_{G(X)|X,L_{\mathbf{C}}(X)}$ is always homogeneous, the equality holds only if $p_{G(X)|L_{\mathbf{C}}(X)}$ is homogeneous. However, if the segmentation is incorrect, $p_{G(X)|L_{\mathbf{C}}(X)=L_{s(i)}}$ is a mixture for at least one $L_{s(i)}$ thus $p_{G(X)|L_{\mathbf{C}}(X)} \neq p_{G(X)|X,L_{\mathbf{C}}(X)}$. This proves the same fact for the n-ary label case.

■ B.2 Statistical Interpretation and Analysis

■ B.2.1 MI as a Confidence Measure

We express the question of whether the image has only a single region or two regions as the following hypothesis testing problem:

$$H_0 : p_1(x) = p_2(x) \quad \text{(single region)} \tag{B.7}$$

$$H_1 : p_1(x) \neq p_2(x) \quad \text{(two regions)} \tag{B.8}$$

Under the null hypothesis H_0, the data $\{G(x)|x \in \Omega\}$ have a single unknown density $p_1 = p_2$, and in this case $p_{G(X)} = p_1 = p_2$, whose estimate is $\hat{p}_{G(X)}$. Thus the log-

likelihood is given by

$$\log p(\{G(x)|x \in \Omega\}|H_0) = \int_\Omega \log \hat{p}_{G(X)}(G(x))dx \quad \text{(B.9)}$$

$$= -|\Omega|\hat{h}(G(X)) \quad \text{(B.10)}$$

Under the alternative hypothesis, the data have two unknown densities p_1 and p_2, and their estimates are $\hat{p}_+$ and $\hat{p}_-$. Thus (3.11) gives the negative of the log-likelihood of the data under H_1. Therefore, we have the log-likelihood ratio in terms of the data size and the mutual information estimate as follows:

$$\log \frac{p(\{G(x)|x \in \Omega\}|H_1)}{p(\{G(x)|x \in \Omega\}|H_0)} = -|\Omega|\hat{h}(G(X)|L_{\vec{C}}(X)) + |\Omega|\hat{h}(G(X)) \quad \text{(B.11)}$$

$$= |\Omega|\hat{I}(G(X); L_{\vec{C}}(X)) \quad \text{(B.12)}$$

This gives a quantitative measure of the belief that H_1 is true.

■ B.2.2 Computing the Z-value

To evaluate the significance of a segmentation result (indicating the existence of two regions in the image), we need to generate samples of the statistic under the null hypothesis that there is a single region. We obtain such samples through random permutations of the binary label. More formally, we define the permutation of the binary labels $L_{\pi,\vec{C}}(\cdot)$ induced by a permutation of the pixels $\pi : \Omega \to \Omega$ as follows:

$$L_{\pi,\vec{C}}(x) \triangleq L_{\vec{C}}(\pi(x)).$$

In a similar way to [23], we perform the following procedure:

- Repeat M times (with index $m = 1$ to M):
  - sample a random permutation π_m from a uniform distribution over the set of all permutations,
  - compute the MI statistic $I_m = \hat{I}(G(X); L_{\pi_m,\vec{C}}(X))$
- compute sample mean and sample variance of $\{I_1, \ldots, I_M\}$.

These sample mean and sample variance are used as estimates of $E[\hat{I}|H_0]$ and $Var[\hat{I}|H_0]$.

■ B.3 Gradient Flows for "Nested" Region Integrals

In Section 3.2.2, we stated the gradient flow for a general nested region integral. In this section, we provide a derivation of the gradient flow (via the first variation) of a curve $\vec{C}$ with respect to an energy integral E over the curve's interior (the region denoted by R). Alternative derivations for this type of region integrals can be found in [21, 30, 63].

We consider a general class of region-based energy functionals E where the integrand f depends upon another family of region integrals $\varepsilon(\mathbf{x}, t)$ of over R. Note that the "nested" region integrals $\varepsilon(\mathbf{x}, t)$ depend on t, since R (the interior of $\vec{C}$) changes as the curve evolves over time. More precisely, we assume as in (3.15)

$$E(\vec{C}) = \int_R f(\varepsilon(\mathbf{x}, t))\, d\mathbf{x} \quad \text{where} \quad \varepsilon(\mathbf{x}, t) = \int_R g(\mathbf{x}, \hat{\mathbf{x}})\, d\hat{\mathbf{x}} \tag{B.13}$$

We start out by using the divergence theorem to rewrite the integral as a contour integral. To do so, we note that there exists a vector field $F(\mathbf{x}, t)$ such that $\nabla \cdot F(\mathbf{x}, t) = f(\varepsilon(\mathbf{x}, t))$ where $\nabla\cdot$ denotes the divergence operator (involving partial derivatives with respect to $\mathbf{x}$ only, not t).

$$E(\vec{C}) = \oint_{\vec{C}} \langle F(\mathbf{x}, t), \vec{N} \rangle\, ds, = \int_0^1 \langle F, J\vec{C}_p \rangle\, dp. \tag{B.14}$$

Note that $p \in [0, 1]$ is a fixed parameterization of $\vec{C}$ which is independent of t (unlike the arclength parameter s). We now differentiate this expression with respect to t in order to determine the form of the gradient flow for $\vec{C}$. In the mathematical development below, $\frac{\partial F}{\partial \mathbf{x}}$ will denote the Jacobian matrix of F with respect to $\mathbf{x}$, while F_t will denote the partial derivative of F with respect to t.

$$\begin{aligned}
\frac{dE}{dt} &= \int_0^1 \left\langle \frac{\partial F}{\partial \mathbf{x}} \vec{C}_t, J\vec{C}_p \right\rangle + \left\langle F, J\vec{C}_{pt} \right\rangle + \left\langle F_t, J\vec{C}_p \right\rangle\, dp && \text{(B.15)} \\
&= \oint_{\vec{C}} \left\langle \vec{C}_t, (f \circ \varepsilon)\vec{N} \right\rangle + \langle F_t, \vec{N} \rangle\, ds && \text{(B.16)} \\
&= \oint_{\vec{C}} \left\langle \vec{C}_t, (f \circ \varepsilon)\vec{N} \right\rangle ds + \int_R \frac{\partial (f \circ \varepsilon)}{\partial t}\, d\mathbf{x} && \text{(B.17)} \\
&= \oint_{\vec{C}} \left\langle \vec{C}_t, (f \circ \varepsilon)\vec{N} \right\rangle ds + \int_R f'\big(\varepsilon(\mathbf{x}, t)\big)\varepsilon_t(\mathbf{x}, t)\, d\mathbf{x} && \text{(B.18)}
\end{aligned}$$

To further manipulate the second term, we note that ε_t appears in the integrand and that $\varepsilon(\mathbf{x}, t)$ *does have* the form of a simple region integral for each $\mathbf{x}$, whose integrand $g(\cdot, \cdot)$ *does not depend on* $\vec{C}$ given its arguments. As such, we may write ε_t as follows:

$$\varepsilon_t(\mathbf{x}, t) = \oint_{\vec{C}} \left\langle \vec{C}_t, g(\mathbf{x}, \vec{C}(s))\vec{N} \right\rangle ds \tag{B.19}$$

Plugging this into the above expression for $\frac{dE}{dt}$ yields

$$\begin{aligned}
\frac{dE}{dt} &= \oint_{\vec{C}} \left\langle \vec{C}_t, (f \circ \varepsilon)\vec{N} \right\rangle ds + \int_R f'\big(\varepsilon(\mathbf{x}, t)\big) \left[\oint_{\vec{C}} \left\langle \vec{C}_t, g(\mathbf{x}, \vec{C})\vec{N} \right\rangle ds \right] d\mathbf{x} && \text{(B.20)} \\
&= \oint_{\vec{C}} \left\langle \vec{C}_t, (f \circ \varepsilon)\vec{N} \right\rangle ds + \int_R \oint_{\vec{C}} \left\langle \vec{C}_t, f'\big(\varepsilon(\mathbf{x}, t)\big) g(\mathbf{x}, \vec{C})\vec{N} \right\rangle ds\, d\mathbf{x} && \text{(B.21)} \\
&= \oint_{\vec{C}} \left\langle \vec{C}_t, \left[f \circ \varepsilon + \int_R f'\big(\varepsilon(\mathbf{x}, t)\big) g(\mathbf{x}, \vec{C})\, d\mathbf{x} \right] \vec{N} \right\rangle ds && \text{(B.22)}
\end{aligned}$$

revealing the following gradient flow for $\vec{C}$ (where t is omitted as an argument for simplicity):

$$\frac{\partial \vec{C}}{\partial t} = -\left[f\big(\varepsilon(\vec{C})\big) + \int_R f'\big(\varepsilon(\mathbf{x})\big) g(\mathbf{x}, \vec{C})\, d\mathbf{x} \right] \vec{N}, \tag{B.23}$$

which is the result we stated in (3.16).

■ B.4 Derivation of the Curve Evolution Formula

This section presents the derivation of the curve evolution formula (3.18) given in Section 3.2.3. We begin by rewriting the energy functional (3.9) as follows:

$$E(\vec{C}) = -|\Omega|\hat{h}(G(X)) + E_+(\vec{C}) + E_-(\vec{C}) + \alpha \oint_{\vec{C}} ds, \tag{B.24}$$

where the components $E_+(\vec{C})$ and $E_-(\vec{C})$ are given by

$$\begin{aligned} E_+(\vec{C}) &= |\Omega| Pr(L_{\vec{C}}(X) = L_+) \hat{h}(G(X) | L_{\vec{C}}(X) = L_+) && \text{(B.25)} \\ &= -\int_{R_+} \log\left(\frac{1}{|R_+|} \int_{R_+} K(G(\mathbf{x}) - G(\hat{\mathbf{x}})) d\hat{\mathbf{x}} \right) d\mathbf{x} && \text{(B.26)} \\ E_-(\vec{C}) &= |\Omega| Pr(L_{\vec{C}}(X) = L_-) \hat{h}(G(X) | L_{\vec{C}}(X) = L_-) && \text{(B.27)} \\ &= -\int_{R_-} \log\left(\frac{1}{|R_-|} \int_{R_-} K(G(\mathbf{x}) - G(\hat{\mathbf{x}})) d\hat{\mathbf{x}} \right) d\mathbf{x}. && \text{(B.28)} \end{aligned}$$

We now proceed with a calculation of the gradient flow for E_+ noting that the flow for E_- will have a similar form (but with an opposite sign). Since $\frac{1}{|R_+|}$ in (B.26) also depends on the curve, we start by breaking E_+ into two integrals:

$$\begin{aligned} E_+ &= -(E_+^1 + E_+^2) && \text{(B.29)} \\ E_+^1 &= -\int_{R_+} \log |R_+| d\mathbf{x} = -|R_+| \log |R_+| && \text{(B.30)} \\ E_+^2 &= \int_{R_+} \overbrace{\log}^{f(\cdot)} \left(\overbrace{\underbrace{\int_{R_+} \overbrace{K(G(\mathbf{x}) - G(\hat{\mathbf{x}}))}^{g(\mathbf{x},\hat{\mathbf{x}})} d\hat{\mathbf{x}}}_{\varepsilon(\mathbf{x},t)}}^{f(\varepsilon(\mathbf{x},t))} \right) d\mathbf{x}, && \text{(B.31)} \end{aligned}$$

where the second integral E_+^2 exhibits the structure of the general nested form given in (3.15) (with the integrand $f()$, the nested integral $\varepsilon()$, and the nested integrand $g()$ labeled accordingly). Using (3.16), the gradient flow for E_+^2, which we denote by $\nabla_C E_+^2$,

is given by

$$\nabla_C E_+^2 = -\Bigg[\overbrace{\log\Big(\int_{R_+} K\big(G(\vec{C})-G(\hat{\mathbf{x}})\big)\,d\hat{\mathbf{x}}\Big)}^{f(\varepsilon(\vec{C}))} + \int_{R_+} \underbrace{\frac{1}{\Big(\int_{R_+} K\big(G(\mathbf{x})-G(\hat{\mathbf{x}})\big)\,d\hat{\mathbf{x}}\Big)}}_{f'(\varepsilon(\mathbf{x}))} \overbrace{K\big(G(\mathbf{x})-G(\vec{C})\big)}^{g(\mathbf{x},\vec{C})}\,d\mathbf{x}\Bigg]\vec{N} \quad \text{(B.32)}$$

$$= -\left[\log|R_+| + \log\hat{p}_+(G(\vec{C})) + \frac{1}{|R_+|}\int_{R_+}\frac{K\big(G(\mathbf{x})-G(\vec{C})\big)}{\hat{p}_+(G(\mathbf{x}))}\right]\vec{N}, \quad \text{(B.33)}$$

while the gradient flow for E_+^1 is given by

$$\nabla_C E_+^1 = -(\nabla_C|R_+|)\log|R_+| - \nabla_C|R_+| = (1+\log|R_+|)\vec{N}. \quad \text{(B.34)}$$

Adding these gradients yields

$$\nabla_C E_+ = -(\nabla_C E_+^1 + \nabla_C E_+^2) = \left[-1+\log\hat{p}_+(G(\vec{C})) + \frac{1}{|R_+|}\int_{R_+}\frac{K\big(G(\mathbf{x})-G(\vec{C})\big)}{\hat{p}_+(G(\mathbf{x}))}\right]\vec{N}. \quad \text{(B.35)}$$

The gradient for E_- has a similar structure (but with an opposite sign since the outward normal with respect to R_- is given by $-\vec{N}$ rather than $\vec{N}$)

$$\nabla_C E_- = -\left[-1+\log\hat{p}_-(G(\vec{C})) + \frac{1}{|R_-|}\int_{R_-}\frac{K\big(G(\mathbf{x})-G(\vec{C})\big)}{\hat{p}_-(G(\mathbf{x}))}\right]\vec{N}. \quad \text{(B.36)}$$

Finally, the overall gradient flow for $E(\vec{C})$ of (3.9) is obtained as follows:

$$\frac{\partial\vec{C}}{\partial t} = \left[\log\frac{\hat{p}_+(G(\vec{C}))}{\hat{p}_-(G(\vec{C}))} + \frac{1}{|R_+|}\int_{R_+}\frac{K\big(G(\mathbf{x})-G(\vec{C})\big)}{\hat{p}_+(G(\mathbf{x}))}\,d\mathbf{x} - \frac{1}{|R_-|}\int_{R_-}\frac{K\big(G(\mathbf{x})-G(\vec{C})\big)}{\hat{p}_-(G(\mathbf{x}))}\,d\mathbf{x}\right]\vec{N} - \alpha\kappa\vec{N}. \quad \text{(B.37)}$$

■ B.5 Approximations of the Second and Third Terms

In Section 3.4, we have empirically observed that the second and third terms in the curve evolution expression in (3.18) have a limited range. Here we show that under

certain assumptions, the values of these terms approach 1. In particular, provided that $|R_+ \cap R_1| >> 1$ and $|R_+ \cap R_2| >> 1$, we have

$$\frac{1}{|R_+|}\int_{R_+} \frac{K\big(G(\mathbf{x}) - G(\vec{C})\big)}{\hat{p}_+(G(\mathbf{x}))}\, d\mathbf{x} \quad \approx \quad 1. \tag{B.38}$$

Similarly, provided that $|R_- \cap R_1| >> 1$ and $|R_- \cap R_2| >> 1$, we have

$$\frac{1}{|R_-|}\int_{R_-} \frac{K\big(G(\mathbf{x}) - G(\vec{C})\big)}{\hat{p}_-(G(\mathbf{x}))}\, d\mathbf{x} \quad \approx \quad 1. \tag{B.39}$$

Derivation

Let $\lambda = \frac{|R_+ \cap R_1|}{|R_+|}$, then $\hat{p}_+ \approx \lambda p_1 + (1-\lambda)p_2$.

Now the approximation is as follows:

$$\begin{aligned}
&\frac{1}{|R_+|}\int_{R_+} \frac{K\big(G(\mathbf{x}) - G(\vec{C})\big)}{\hat{p}_+(G(\mathbf{x}))}\, d\mathbf{x} \\
&= \frac{|R_+ \cap R_1|}{|R_+|}\left[\frac{1}{|R_+ \cap R_1|}\int_{R_+\cap R_1} \frac{K\big(G(\mathbf{x}) - G(\vec{C})\big)}{\hat{p}_+(G(\mathbf{x}))}\right] \\
&\quad + \frac{|R_+ \cap R_2|}{|R_+|}\left[\frac{1}{|R_+ \cap R_2|}\int_{R_+\cap R_2} \frac{K\big(G(\mathbf{x}) - G(\vec{C})\big)}{\hat{p}_+(G(\mathbf{x}))}\right] && \text{(B.40)} \\
&\approx \lambda E_{p_1}\left[\frac{K\big(Y - G(\vec{C})\big)}{\hat{p}_+(Y)}\right] + (1-\lambda)E_{p_2}\left[\frac{K\big(Y - G(\vec{C})\big)}{\hat{p}_+(Y)}\right] && \text{(B.41)} \\
&\approx \lambda\int \frac{p_1(y)K(y - G(\vec{C}))}{\lambda p_1(y) + (1-\lambda)p_2(y)}dy + (1-\lambda)\int \frac{p_2(y)K(y - G(\vec{C}))}{\lambda p_1(y) + (1-\lambda)p_2(y)}dy && \text{(B.42)} \\
&= \int K(y - G(\vec{C}))dy && \text{(B.43)} \\
&= 1 && \text{(B.44)}
\end{aligned}$$

The derivation of (B.39) is similar to that of (B.38).

Appendix C

Information Theoretic Quantities Associated with the Shape Distribution

With the nonparametric shape density estimate, we can estimate information theoretic quantities associated with a random shape such as the entropy of a random shape or KL divergence between two shape distributions.

The entropy of a shape distribution $p_S(\tilde{\phi})$ is given by

$$h(p_S(\tilde{\phi})) = -\int_{\mathcal{S}} p_S(\tilde{\phi}) \log p_S(\tilde{\phi}) d\tilde{\phi} \tag{C.1}$$

where the support of integral is the shape space $\mathcal{S}$.

The entropy can be estimated in terms of nonparametric density estimates.

$$\begin{aligned} h(p_S(\tilde{\phi})) &= -E_p[\log p_S(\tilde{\Phi})] && \text{(C.2)} \\ &\approx -\frac{1}{n}\sum_i \log p_S(\tilde{\phi}_i) && \text{(C.3)} \\ &\approx -\frac{1}{n}\sum_i \log \frac{1}{n}\sum_j k(d_{\mathcal{D}}(\tilde{\phi}_i, \tilde{\phi}_j), \sigma) && \text{(C.4)} \end{aligned}$$

Suppose we have two shape distributions p_1 and p_2, and sample shapes $\tilde{\phi}_1, \ldots, \tilde{\phi}_n$ drawn from p_1 and sample shapes $\tilde{\psi}_1, \ldots, \tilde{\psi}_m$ drawn from p_2. We can estimate KL

divergence between the two shape distributions as follows:

$$
\begin{aligned}
D(p_1||p_2) &= \int_{\mathcal{S}} p_1(\tilde{\phi}) \frac{\log p_1(\tilde{\phi})}{\log p_2(\tilde{\phi})} d\tilde{\phi} && \text{(C.5)} \\
&= E_{p_1}[\log \frac{p_1(\tilde{\Phi})}{p_2(\tilde{\Phi})}] && \text{(C.6)} \\
&\approx \frac{1}{n} \sum_i \log \frac{p_1(\phi_i)}{p_2(\phi_i)} && \text{(C.7)} \\
&\approx \frac{1}{n} \sum_i \log \frac{\frac{1}{n}\sum_j k(d_{\mathcal{D}}(\tilde{\phi}_i, \tilde{\phi}_j), \sigma_1)}{\frac{1}{m}\sum_j k(d_{\mathcal{D}}(\tilde{\phi}_i, \tilde{\psi}_j), \sigma_2)} && \text{(C.8)}
\end{aligned}
$$

where the two kernel sizes σ_1 and σ_2 are computed from $\{\tilde{\phi}_i\}$ and $\{\tilde{\psi}_i\}$ respectively.

Bibliography

[1] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *J. Comput. Phys.*, 118:269–277, 1995.

[2] Ibrahim A. Ahmad and Pi-Erh Lin. A nonparametric estimation of the entropy for absolutely continuous distributions. *IEEE Transactions on Information Theory*, pages 372–375, May 1976.

[3] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, pages 716–723, December 1974.

[4] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson. Image segmentation using active contours: Calculus of variation or shape optimization ? *SIAM Journal on Applied Mathematics*, 63(6):2128–2154, 2003.

[5] J. Beirlant, E. J. Dudewicz, L. Györfi, and E. C. van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of Math. Stat. Sci.*, 6(1):17–39, 1997.

[6] T. Cacoullos. Estimation of a multivariate density. *Annals of the Institute of Statistical Mathematics*, 18:179–190, 1965.

[7] J. F. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[8] V. Caselles, F. Catte, T. Col, and F. Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66:1–31, 1993.

[9] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[10] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2):266–277, February 2001.

[11] Tony F. Chan and Luminita A. Vese. An efficient variational multiphase motion for the mumford-shah segmentation model. In *Proc. Asilomar Conf. on Signals, Systems, and Computers*, pages 490–494, 2000.

[12] Guillaume Charpiat, Olivier Faugeras, and Renaud Keriven. Approximations of shape metrics and application to shape warping and empirical shape statistics. Technical report, INRIA, May 2003.

[13] D. L. Chopp. Computing minimal surfaces via level set curvature flow. *J. Comput. Phys.*, 106:77–91, 1993.

[14] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models— their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[15] T. F. Cootes, C. J. Twining, and C. J. Taylor. Diffeomorphic statistical shape models. In *Proc. British Machine Vision Conference*, volume 1, pages 447–456, 2004.

[16] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[17] Daniel Cremers, Timo Kohlberger, and Christoph Schnörr. Nonlinear shape statistics in mumford-shah based segmentation. In *Proc. European Conference in Computer Vision*, pages 93–108, 2002.

[18] Daniel Cremers and Stefano Soatto. A pseudo-distance for shape priors in level set segmentation. In *IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, 2003.

[19] Rhodri H. Davies, Tim F. Cootes, and Chris. J. Taylor. A minimum description length approach to statistical shape modelling. In *Proc. Information Processing in Medical Imaging*, pages 50–63, 2001.

[20] Rhodri H. Davies, Carole J. Twining, Tim F. Cootes, John C. Waterton, and Chris. J. Taylor. A minimum description length approach to statistical shape modeling. *IEEE Trans. on Medical Imaging*, 21(5):525–537, 2002.

[21] M. C. Delfour and J.P. Zolesio. *Shapes and Geometries: Analysis, Differential Calculus, and Optimization*. SIAM, 2001.

[22] Luc Devroye, László Györfi, and Gábor Lugosi. *A Prababilisitic Theory of Pattern Recognition*. Springer, 1996.

[23] Polina Golland and Bruce Fischl. Permutation tests for classification: Towards statistical significance in image-based studies. In *Proc. Information Processing in Medical Imaging*, volume 2732 of *LNCS*, pages 330–341, 2003.

[24] M Grayson. The heat equation shrinks embedded plane curves to round points. *Journal of Differential Geometry*, 26:285–314, 1987.

[25] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica*, 6:229–269, 1997.

[26] Leslie Greengard and John Strain. The fast Gauss transform. *SIAM J. Sci. Stat. Comput.*, 12(1):79–94, 1991.

[27] Robert V. Hogg and Elliot A. Tanis. *Probability and Statistical Inference*. Prentice Hall, 2000.

[28] Alex Ihler. Maximally informative subspaces: Nonparametric estimation for dynamical systems. Master's thesis, Massachusetts Institute of Technology, 2000.

[29] David W. Jacobs. Robust and efficient detection of salient convex groups. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(1):23–37, 1996.

[30] Stephanie Jehan-Besson and Michel Marlaud. Dream^{2}s: Deformable regions driven by an eulerian accurate minimization method for image ad video segmentation. *International Journal of Computer Vision*, 53:45–70, 2003.

[31] Harry Joe. On the estimation of entropy and other functionals of a multivariate density. *Ann. Inst. Statist. Math.*, 41:683–697, 1989.

[32] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[33] D. G. Kendall. Shape manifolds, procrustean metrics and complex projective spaces. *Bull. of London Math. Soc.*, 16:81–121, 1984.

[34] Junmo Kim, John W. Fisher, Müjdat Çetin, Anthony Yezzi, Jr., and Alan S. Willsky. Incorporating complex statistical information in active contour-based image segmentation. In *Proc. IEEE Conf. on Image Processing*, volume 2, pages 655–658, 2003.

[35] Junmo Kim, John W. Fisher, Anthony Yezzi, Jr., Müjdat Çetin, and Alan S. Willsky. Nonparametric methods for image segmentation using information theory and curve evolution. In *Proc. IEEE Conf. on Image Processing*, volume 3, pages 797–800, 2002.

[36] Junmo Kim, John W. Fisher, Anthony Yezzi, Jr., Müjdat Çetin, and Alan S. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *IEEE Trans. on Image Processing*, to appear.

[37] Charles Kittel and Herbert Kroemer. *Thermal Physics*. W. H. Freeman and Company, 1980.

[38] Eric Klassen, Anuj Srivastava, Washington Mio, and Shantanu H. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(3):372–383, March 2004.

[39] Y. Leclerc. Constructing stable descriptions for image partitioning. *International Journal of Computer Vision*, 3:73–102, 1989.

[40] Michael E. Leventon, W. Eric L. Grimson, and Olivier Faugeras. Statistical shape influence in geodesic active contours. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 316–323. IEEE, 2000.

[41] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.

[42] D. Marr and E.C. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London, B*, 207:187–217, 1980.

[43] Peter W. Minchor and David Mumford. Riemannian geometries on spaces of plane curves. *J. Eur. Math. Soc. (JEMS)*, to appear.

[44] David Mumford. The problem of robust shape descriptors. In *Proceedings of the IEEE First International Conference on Computer Vision*, pages 602–606, 1987.

[45] David Mumford. Mathematical theories of shape: Do they model perception? In *SPIE Proceedings Vol. 1570 Geometric Methods in Computer Vision*, pages 2–10, 1991.

[46] David Mumford and Jayant Shah. Boundary detection by minimizing functionals. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 22–26, 1985.

[47] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications in Pure and Applied Mathematics*, 42(4):577–685, 1989.

[48] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[49] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

[50] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

[51] Nikos Paragios and Richid Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *Int. J. Computer Vision*, 2002.

[52] Nikos Paragios and Mikael Rousson. Shape priors for level set representations. In *Proc. European Conference in Computer Vision*, 2002.

[53] Emanuel Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[54] Emanuel Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[55] S. Rachev. The Monge-Kantorovich mass transfer problem and sampling theory. *Theory of Probability and Applications*, 29:647–676, 1985.

[56] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27:832–837, 1956.

[57] P. Salembier and F. Marqués. Region-based representations of image and video: Segmentation tools for multimedia services. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(8):1147–1169, 1999.

[58] J. A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Science*. Cambridge University Press, 1996.

[59] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[60] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[61] C. G. Small. *The Statistical Theory of Shapes*. Springer-Verlag, 1996.

[62] S. Soatto and Anthony Yezzi, Jr. Deformotion, deforming motion, shape average and the joint registration ad segmentation of images. In *Proceedings of the 7th European Conference on Computer Vision*, pages 32–47, May 2002.

[63] J. Sokolowski and Jean-Paul Jolesio. *Introduction to Shape Optimization: Shape Sensitivity Analysis*. Springer-Verlag, 1992.

[64] J. Strain. The fast Gauss transform with variable scales. *siamSSC*, 12(5):1131–1139, 1991.

[65] Walter A. Strauss. *Partial Differential Equations: An Introduction*. Wiley, 1992.

[66] Andy Tsai. *Curve Evolution and Estimation-Theoretic Techniques for Image Processing*. PhD thesis, Massachusetts Institute of Technology, 2000.

[67] Andy Tsai, Anthony Yezzi, Jr., William Wells, Clare Tempany, Dewey Tucker, Ayres Fan, W. Eric Grimson, and Alan Willsky. Model-based curve evolution technique for image segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 463–468, 2001.

[68] Andy Tsai, Anthony Yezzi, Jr., William Wells, Clare Tempany, Dewey Tucker, Ayres Fan, W. Eric Grimson, and Alan Willsky. A shape-based apprach to the segmentation of medical imagery using level sets. *IEEE Trans. on Medical Imaging*, 22(2):137–154, February 2003.

[69] J. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Trans. on Automatic Control*, 40:1528–1538, 1995.

[70] Shimon Ullman. *High-level Vision*. The MIT Press, 1996.

[71] Luminita A. Vese and Tony F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.

[72] Paul Viola. *Alignment by Maximization of Mutual Information*. PhD thesis, Massachusetts Institute of Technology, 1995.

[73] Paul Viola and William M. Wells, III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[74] Anthony Yezzi, Jr., Andy Tsai, and Alan Willsky. A statistical approach to snakes for bimodal and trimodal imagery. In *Int. Conf. on Computer Vision*, pages 898–903, 1999.

[75] Hong-Kai Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127:179–195, 1996.

[76] Song-Chun Zhu. Embedding geestalt laws in markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(11):1170–1187, November 1999.

[77] Song Chun Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(9):884–900, September 1996.