Bayesian Methods for Segmentation of Objects from Multimodal and Complex Shape Densities using Statistical Shape Priors

by

Ertunç Erdil

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of
Doctor of Philosophy

Sabancı University
December 2017

Bayesian Methods for Segmentation of Objects from Multimodal and Complex Shape Densities using Statistical Shape Priors

APPROVED BY

Assoc. Prof. Dr. Müjdat ÇETİN .....
(Thesis Supervisor)

Assoc. Prof. Dr. Devrim ÜNAY .....
(Thesis Co-supervisor)

Assoc. Prof. Dr. Selim BALCISOY .....

Assist. Prof. Dr. Sinan YILDIRIM .....

Assoc. Prof. Dr. Tolga TAŞDİZEN .....

Assist. Prof. Dr. Ender KONUKOĞLU .....

DATE OF APPROVAL: 13.12.2017

Acknowledgments

I feel very lucky to have many great people to acknowledge. The accomplishments in this dissertation would not be possible without their support and guidance.

First, and foremost, I was very fortunate to have Dr. Mujdat Cetin as my advisor. Many pages of acknowledgement would not be enough to express my sincere gratitude to him. He has tremendous contribution in the technical content of this dissertation and my technical skills. More importantly, he has helped me to gain a new perspective and vision which I will use in my future career. I was also very fortunate to have Dr. Devrim Unay as my co-advisor. I thank to him for giving me the opportunity to work in dendritic spine project. I want to thank to other members of my dissertation committee, Dr. Tolga Tasdizen for his valuable feedback and discussion in every stage of this dissertation, Dr. Sinan Yildirim for helping me to learn and develop MCMC methods, Dr. Selim Balcisoy and Dr. Ender Konukoglu for a careful evaluation of my work and their useful feedback.

It was a pleasure to me being a member of SPIS lab. I am really thankful to my friends in SPIS lab for the great times we spent together. I am also indebted to all of my friends for their endless support during the course of my work. I thank TUBITAK for providing financial support to my Ph.D.

Finally, I am grateful to my family for their encouragement, support and pure love.

BAYESIAN METHODS FOR SEGMENTATION OF OBJECTS FROM MULTIMODAL AND COMPLEX SHAPE DENSITIES USING STATISTICAL SHAPE PRIORS

Ertunç Erdil

Computer Science, Ph.D. Thesis, 2017

Thesis Supervisor: Assoc. Prof. Müjdat ÇETİN

Thesis Co-supervisor: Assoc. Prof. Devrim ÜNAY

Keywords: shape prior, kernel density estimation, level set, Markov chain Monte Carlo, image segmentation, multimodal shape density

Abstract

In many image segmentation problems involving limited and low-quality data, employing statistical prior information about the shapes of the objects to be segmented can significantly improve the segmentation result. However, defining probability densities in the space of shapes is an open and challenging problem, especially if the object to be segmented comes from a shape density involving multiple modes (classes).

In the literature, there are some techniques that exploit nonparametric shape priors to learn multimodal prior densities from a training set. These methods solve the problem of segmenting objects of limited and low-quality to some extent by performing maximum *a posteriori* (MAP) estimation. However, these methods assume that the boundaries found by using the observed data can provide at least a good initialization for MAP estimation so that convergence to a desired mode of the posterior density is achieved. There are two major problems with this assumption that we focus in this thesis. First, as the data provide less information, these approaches can get stuck at a local optimum which may not be the desired solution. Second, even though a good initialization directs the segmenting curve to a local optimum

solution that looks like the desired segmentation, it does not provide a picture of other probable solutions, potentially from different modes of the posterior density, based on the data and the priors.

In this thesis, we propose methods for segmentation of objects that come from multimodal posterior densities and suffer from severe noise, occlusion and missing data. The first framework that we propose represents the segmentation problem in terms of the joint posterior density of shapes and features. We incorporate the learned joint shape and feature prior distribution into a maximum *a posteriori* estimation framework for segmentation. In our second proposed framework, we approach the segmentation problem from the approximate Bayesian inference perspective. We propose two different Markov chain Monte Carlo (MCMC) sampling based image segmentation approaches that generates samples from the posterior density. As a final contribution of this thesis, we propose a new shape model that learns binary shape distributions by exploiting local shape priors and the Boltzmann machine. Although the proposed generative shape model has not been used in the context of object segmentation in this thesis, it has great potential to be used for this purpose. The source code of the methods introduced in this thesis will be available in `https://github.com/eerdil`.

ÇOK DORUKLU VE KARMAŞIK ŞEKİL DAĞILIMLARINDAN GELEN NESNELERİN İSTATİSTİKSEL ŞEKİL ÖN BİLGİSİ KULLANARAK BÖLÜTLENMESİ İÇİN BAYESÇİ YAKLAŞIMLAR

Ertunç Erdil

Bilgisayar Bilimleri, Doktora Tezi, 2017

Tez danışmanı: Assoc. Prof. Müjdat ÇETİN

Tez eş-danışmanı: Assoc. Prof. Devrim ÜNAY

Anahtar Kelimeler: şekil ön bilgisi, çekirdek yoğunluk kestiricisi, Markov zinciri Monte Carlo, imge bölütleme, çok doruklu şekil dağılımları

Özet

Sınırlı ve düşük kaliteli görüntüler ieren bir çok bölütleme probleminde bölütlenecek nesne ile ilgili istatistiksel şekil ön bilgisini kullanmak bölütleme sonuçlarını önemli derecede iyileştirmektedir. Ancak, şekil uzayında olasılık yeğinlik fonksiyonunun tanımlanması, özellikle şekil çok doruklu bir şekil yeğinlik fonksiyonundan geliyorsa, zorlu ve araştırmaya açık bir problemdir.

Literatürde parametrik olmayan şekil ön bilgisinden yararlanarak bir eğitim kümesinden şekil önsel dağılımını öğrenen yöntemler bulunmaktadır. Bu yöntemler, sınırlı ve düşük kaliteli veride bulunan nesneleri sonsal dağılımın en büyüğü kestirimi yöntemi ile bölütler. Ancak bu yöntemler, veriden gelen bilgi ile bulunan bölütleme sınırlarının, sonsal dağılımın en büyüğü kestirimi sonsal dağılımın istenilen doruğuna yakınsayacak şekilde iyi bir ilklendirme olduğu kabullenmesini yapar. Bu kabullenme ile ilgili iki temel problem vardır. Birinci problem, veri kötüleştikçe bu yöntemlerin istenen çözüm olmama ihtimali olan bir yerel en iyi çözümünde takılı kalmasıdır. İkinci problem, ilklendirmenin iyi olduğu durumda istenilen yerel en iyi çözüme gidilse bile, sonsal dağılımın farklı doruklarındaki diğer olası çözümler ile ilgili bir bilgi vermemesidir.

Bu tezde, çok doruklu sonsal dağılımlardan gelen şekillerin verinin yeterince iyi olmadığı durumlarda bölütlenmesi için yöntemler önermekteyiz. Önerdiğimiz ilk yöntem bölütleme problemini şekil ve öz nitelik ortak sonsal dağılımı olarak temsil eder. Bir eğitim veri kümesinden öğrenilen ortak şekil ve öz nitelik önsel dağılımı kullanılarak sonsal dağılımın en büyüğü kestirimi yöntemi ile bölütleme sonucu elde edilir. İkinci olarak bölütleme problemine Bayesçi çıkarım bakış açısından bakmaktayız. Bu tezde Markov zinciri Monte Carlo örneklemesi tabanlı, sonsal dağılımdan örnekler üreten iki farklı yöntem önermekteyiz. Bu tezdeki son katkı olarak ikili şekil dağılımlarını, yerel şekil ön bilgisi ve Boltzmann makinasından yararlanarak öğrenen yeni bir şekil modeli önermekteyiz. Bu tezde, üretici modeller bölütleme problemi için kullanılmamış olsa da bu amaçla kullanılabilmeleri mümkündür. Bu tezde tanıtılan yöntemlerin kaynak kodları `https://github.com/eerdil` adresinde erişime açık olacaktır.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image segmentation can be defined as the process of grouping meaningful regions in a given image and it is one of the most fundamental problems in image processing and computer vision. The output of segmentation can be used for various applications ranging from object detection to medical image analysis. In this thesis, we consider challenging problems in which the observed image data alone are insufficient for effective segmentation and need to be supplemented with other pieces of statistical information about the shapes or other features of the objects to be segmented. With this perspective, we develop new Bayesian methods for segmentation of objects from multimodal and complex shape densities using statistical shape and feature priors.

1.1 Recent work on image segmentation

There have been significant efforts for developing general purpose segmentation algorithms in the literature. Some of these attempts are based on edge detection [4] [5] [6], graph theory [7] [8] and active contours [9]. Edge detection based methods start segmentation by detecting edges. In general, detected edges include those of fragmented and redundant ones. Edge detection based approaches then convert these types of edges to form a closed contour which is expected to be the ultimate segmentation. Graph theory based segmentation approaches form a weighted graph where nodes of the graph correspond to image pixels and weights correspond to the likelihood of the pixels at both ends being in the same region. Once the graph is constructed, the segmentation is found by finding a cut that minimizes the cost of

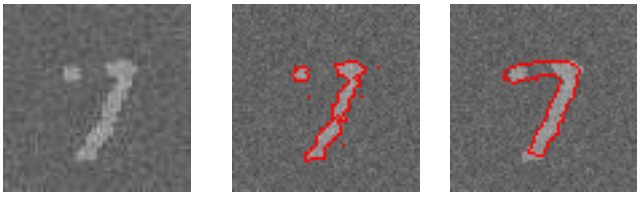
the cut. A common choice for the cost function is the sum of the weights on the cut. As the optimization process is generally NP-hard, approximation to the optimal solution is preferred rather than the analytic solution [8].

In this thesis, we focus on active contour based image segmentation methods. The idea of active contours was first proposed by Kass et al. [9]. The initial approach represents the boundary between regions as a closed contour which is evolved until it converges to the boundary of the desired region. The segmentation problem is often represented as an optimization problem where a cost function that depends on the evolving contour is minimized to obtain the ultimate segmentation. Active contour based methods have two major advantages over the edge detection and graph theory based methods. First, active contour based methods do not require an explicit effort to sustain a closed curve. Second, optimization of a cost function can be performed in polynomial time. Active contour based methods have become more popular after level set methods have been introduced by Osher and Sethian [10] [11]. By using level set methods, boundaries with complex geometries and topological changes can be handled during the curve evolution process in a natural way and automatically. The initial active contour based approach of Kass et al. [9] uses a simple assumption about the curve length as a prior. Later, more complicated shape priors have been proposed in the active contour framework such as the ones in [1, 12–17]. In [15] and [16], shape variability is captured using PCA on signed distance functions of level sets. However, these techniques work well only when the shape variation is small due to their use of PCA. Therefore, they cannot handle multimodal shape densities. In order to learn multimodal shape densities, Kim et al. [1] and Cremers et al. [17] proposed nonparametric density estimation based shape priors using level sets to handle multimodal shape densities. Various extensions and applications of these methods that exploit nonparametric shape priors can be found in [2, 18–21].

1.2 Motivation for and highlights of the proposed methods

In this thesis, we propose novel active contour based image segmentation methods that exploits nonparametric shape priors. Let us consider the problem of seg-

menting a noisy and partially occluded digit shown in Figure 1.1(a). Segmentation of such images that suffer from missing data, occlusion, or severe noise requires a prior knowledge about the shape to be segmented. Otherwise, when there is no prior shape information, segmentation based on the information obtained from data results in a segmentation similar to the one shown in Figure 1.1(b). Given some shape samples from each digit class, segmentation approaches that uses nonparametric shape priors can learn the prior shape distribution from the training data and incorporate this information into the segmentation process together with the information that comes from data. Those approaches produce successful segmentation results when the information provided by data is limited. The segmentation result of an approach that exploits nonparametric shape priors produces segmentations similar to the one shown in Figure 1.1(c)

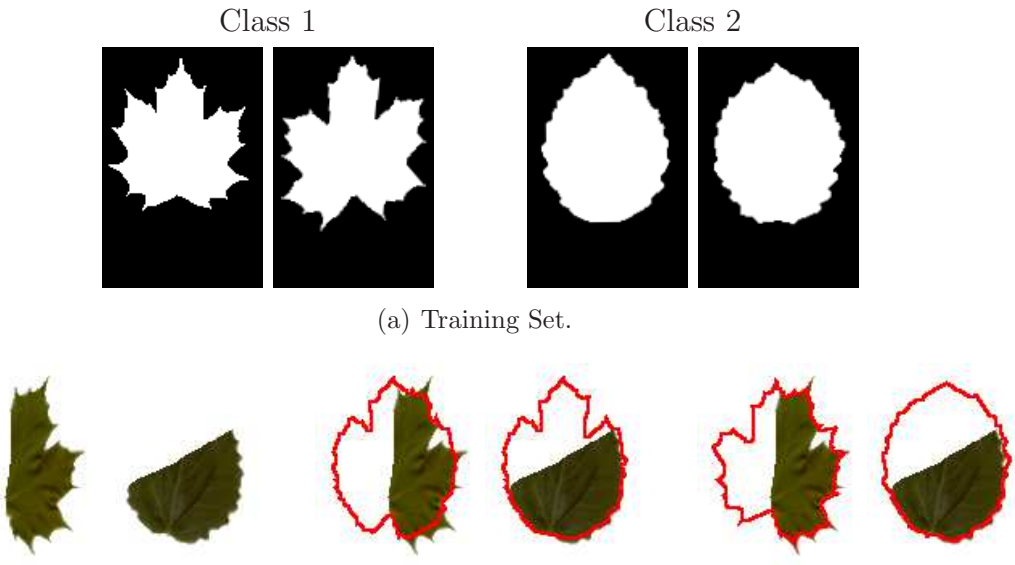


(a) Test image. (b) Data driven segmentation. (c) Segmentation using nonparametric shape priors.

Figure 1.1: A toy example that shows advantages of using nonparametric shape priors.

The state-of-the-art segmentation methods that use nonparametric shape priors produce poor segmentation results when the information obtained from data is less and shapes in different classes are similar to each other in terms of a particular distance metric. Let us consider the visual example shown in Figure 1.2. In this example, given the training set shown in Figure 1.2(a), a nonparametric shape priors based approach produces segmentation results from the same class for test images from different classes (see Figure 1.2(c)). In this example, using only shape priors is not sufficient to produce correct segmentations. This motivates us to develop a segmentation algorithm that exploits some class-related features together with the nonparametric shape priors to achieve better segmentation results. The proposed

approach generates segmentations from correct classes for each test image as shown in Figure 1.2(d).



(a) Training Set.

(b) Test Image.

(c) Segmentation using nonparametric shape priors.

(d) Proposed approach.

Figure 1.2: The first motivating example.

In the literature, the segmentation methods that use nonparametric shape priors represent the segmentation problem in Bayesian framework and perform maximum *a posteriori* estimation on the resulting posterior density. In other words, these approaches return a single segmentation solution at a local optimum. This does not provide a measure of the degree of confidence in that result, neither does it provide a picture of other probable solutions based on the data and priors. With a statistical view, addressing these issues would involve the problem of characterizing the posterior densities of the shapes of the objects to be segmented. This motivates us to develop Markov chain Monte Carlo (MCMC) sampling-based image segmentation algorithms that use nonparametric shape priors. Our sampling-based segmentation approaches can generate multiple solutions from different modes of the posterior density. Going back to the segmentation problem shown in Figure 1.1(a), our sampling-based approaches produce segmentations from different digit classes as shown in Figure 1.3.

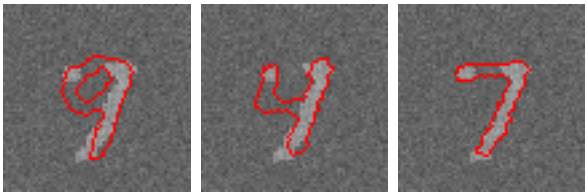


Figure 1.3: The second motivating example.

1.3 Contributions of this thesis

In this section, we briefly decribe the contributions of this thesis:

- We propose a novel segmentation algorithm that exploits nonparametric shape and feature priors for object segmentation where the object to be segmented comes from a multimodal shape density. Unlike the state-of-the-art methods that perform segmentation using nonparametric shape density estimation, we exploit learned discriminative class-dependent features extracted from specific parts of the scene and incorporate the joint shape and feature density into the segmentation process.

  This work has been done in collaboration with M. Usman Ghani, Lavdie Rada, A. Ozgur Argunsah, Devrim Unay, Tolga Tasdizen and Mujdat Cetin.

- We propose a novel Markov chain Monte Carlo shape sampling approach for image segmentation using nonparametric shape priors. To the best of our knowledge, this is the first MCMC sampling-based approach that exploits nonparametric shape priors and level sets.

  This work has been done in collaboration with Sinan Yildirim, Tolga Tasdizen and Mujdat Cetin.

- We propose a novel pseudo-marginal Markov chain Monte Carlo shape sampling approach for image segmentation. To the best of our knowledge, pseudo-marginal sampling has not been used in the literature for image segmentation before. Moreover, unlike the existing MCMC sampling-based segmentation methods in the literature, the proposed approach perfectly satisfies necessary conditions to implement MCMC sampling; this is very crucial to ensure that the generated samples come from the desired density.

This work has been done in collaboration with Sinan Yildirim, Tolga Tasdizen and Mujdat Cetin.

- We propose a novel shape model called Disjunctive Normal Shape Boltzmann Machine (DNSBM) to learn a binary shape distribution. The proposed approach exploits the property of the Shape Boltzmann Machine [22] for learning complex binary shape distributions and the property of Disjunctive Normal Shape Model (DNSM) [23] for representing local shape parts. DNSBM can learn shape distributions when the training set is limited and generate valid and novel samples. Although, DNSBM has not yet been applied to segmentation, the shape model has the potential to be used in a segmentation pipeline.

  This work has been done in collaboration with Fitsum Mesadi, Tolga Tasdizen and Mujdat Cetin.

1.4 Thesis organization

This thesis is organized as follows:

1.4.1 Chapter 2: Background

In this chapter, we give an overview of the concepts that are necessary for understanding the background of this thesis. These include nonparametric density estimation, Markov chain Monte Carlo methods, and level set methods.

1.4.2 Chapter 3: Nonparametric Joint Shape and Feature Priors for Image Segmentation

In this chapter, we propose a novel segmentation algorithm that exploits nonparametric joint shape and feature priors. First, we provide an overview the related work in the literature. Second, we give our motivation and contributions. Then, we introduce the proposed approach and present the experimental results. Finally, we conclude and briefly mention potential directions for future work.

1.4.3 Chapter 4: Markov Chain Monte Carlo Sampling-based Methods for Image Segmentation with Nonparametric Shape Priors

In this chapter, we propose two novel Markov chain Monte Carlo sampling-based approaches that exploits nonparametric shape priors for image segmentation. First, we describe a non-exhaustive survey of the existing MCMC sampling-based image segmentation methods. Second, we give our motivation for developing MCMC sampling-based segmentation approaches with nonparametric shape priors. In the following two sections, we present the proposed approaches together with the technical details and experimental results of each piece of work.

1.4.4 Chapter 5: Disjunctive Normal Shape Boltzmann Machine

In this chapter, we propose a novel shape model for learning binary shape distributions called Disjunctive Normal Shape Boltzmann Machine. First, we briefly introduce existing models in the literature that have potential to learn binary shape distributions. Then, we describe our motivation for developing the proposed shape model and our contributions in this work. Later, we introduce the proposed shape model. Finally, we present experimental results, conclusion, and future work.

1.4.5 Chapter 6: Conclusion

In this chapter, we conclude by revisiting the contributions of this thesis. We also indicate possible research directions for future work motivated by the open problems of relevance.

Chapter 2

Background

In this chapter, we give an overview of the concepts that are necessary for understanding the background of this thesis. In particular, this chapter covers level set methods, nonparametric density estimation, and Markov chain Monte Carlo (MCMC) methods.

2.1 Level set methods

Curve evolution approaches are generally based on minimizing an energy function, $E(c)$, of segmenting curve c. This is usually achieved by updating an initial curve c with the gradient of $E(c)$ until convergence. Shape representation is crucial when implementing curve evolution.

There are two approaches for the numerical implementation of a curve evolution: Lagrangian and Eulerian (fixed coordinate system). A Lagrangian approach first divides the boundary into discrete segments and evolves these discrete points (marker points). This is the most intuitive approach, however, it brings a number of problems. First, a Lagrangian approach requires very small time steps for stable evolution of the boundary [11]. Moreover, in the case of topological changes such as constructing a hole within a closed shape, it requires complicated procedures. On the other hand, an Eularian approach called the level set method can avoid the stability problem and can naturally handle topological changes.

Level set is a widely used shape representation to implement curve evolution based segmentation methods [11]. Level set methods are numerical techniques for tracking evolving surfaces which can handle topological changes such as holes and

shapes with multiple unconnected components. When using level sets for curve evolution in image segmentation, we expect to evolve the curve towards the region that we want to segment. This is achieved by initializing a curve somewhere in the image and evolving it with the gradient of an energy function until convergence. Level set methods use an implicit representation of the curve by operating on a function in one dimension higher.

Let us consider a closed curve $c \in \mathbb{R}^2$ that divides the image domain Ω into three parts: the region inside the curve R, the region outside the curve R^c and the boundary c. The idea of level set representation proposed by Osher and Sethian [11] is to define a smooth function $\phi(x)$ such that $\phi(x) = 0$ represents the boundary C. This function ϕ is called as a level set function and has the following property:

$$\begin{aligned} \phi(x) < 0, x \in R \\ \phi(x) > 0, x \in R^c \\ \phi(x) = 0, x \in c \end{aligned} \tag{2.1}$$

Note that there are many level set functions given a boundary c. However, given a level set function, the boundary can be uniquely determined.

In order to model curve evolution, it is a common practice make the level set a function of time as well as space. Following this practice, we can write the level set function as $\phi = \phi(x, t)$ where t indicates artificial time. Then, the curve c at a given time t becomes the isocontour of ϕ at zero level, i.e. $c(t) = \{x : \phi(x, t) = 0\}$.

We can define the level set function as $\phi : \Omega \times [0, \infty) \rightarrow \mathbb{R}$ where Ω indicates the image domain and $[0, \infty)$ indicates the time domain. The level set function is initialized at time zero and evolved in time until it stops.

As we mentioned above, there are many level set functions that indicate the same boundary c. In practice, the level set function is generally constructed using the signed distance function as

$$\phi_0 = \phi(x, t = 0) = \pm d. \tag{2.2}$$

In Equation 2.2, $\pm d$ is the signed Euclidean distance from each point $x \in \Omega$ to the closest point on the boundary c. If $x \in R$, the sign of the Euclidean distance is

negative and if $x \in R^c$ the sign of the Euclidean distance is positive. By definition, Euclidean distance is zero if $x \in c$ and the motion of the curve is described by matching the new curve to the zero isocontour of the level set function. The level set value of a point, x, on c is always zeros as the curve propagates:

$$\phi(x,t) = 0. \tag{2.3}$$

Differentiating the above equation with respect to t, we obtain

$$\phi_t(x,t) + \nabla\phi(x,t) \cdot \frac{\partial x}{\partial t} = 0 \tag{2.4}$$

where, ϕ_t is the partial derivative of ϕ with respect to t. Let us also define a function called the speed function F that drives the curve to the desired location. More specifically, F is the speed in the outward normal direction to the level set interface. Then, we can write F as

$$F = \frac{\partial x}{\partial t} \cdot N \tag{2.5}$$

where

$$N = \frac{\nabla\phi}{|\nabla\phi|}$$

is the outward unit normal to the level set function ϕ.

We can rewrite Equation 2.4 as

$$\phi_t + \frac{\nabla\phi}{|\nabla\phi|}\frac{\partial x}{\partial t}|\nabla\phi| = 0$$

$$\phi_t + (\frac{\partial x}{\partial t}N)|\nabla\phi| = 0. \tag{2.6}$$

Note that we have

$$\frac{\partial x}{\partial t} = F.N$$

Then, we can write Equation 2.6 as

$$\phi_t + F.|\nabla\phi| = 0. \tag{2.7}$$

If ϕ is a signed distance function, it satisfies the Eikonal equation [24]

$$|\nabla\phi| = 1.$$

In this case, the outward normal vector is written as

$$N = \nabla\phi. \tag{2.8}$$

Using signed distance functions have some useful features such as simplifying computations of several quantities and allowing more stable computations. When implementing a curve evolution framework with level sets, numerical errors can accumulate in each update of the level set function and signed distance properties are not retained. To avoid these problems, it is a good practice to reinitialize the level set function to a signed distance function during curve evolution [25] [26].

As we mentioned above, an initial level set ϕ_0 is updated in time using the speed function at the corresponding time point and the outward normal direction. Therefore, given ϕ_t the task is to find the update after some time increment ∇t that produces ϕ_{t+1}. This is achieved by the Euler method [26] by approximating ϕ_t at time t as

$$\phi_t = \frac{\phi_{t+1} - \phi_t}{\nabla t}. \tag{2.9}$$

Finally, by plugging the above equation into Equation 2.7, we get the following update equation for the level set function in time

$$\phi_{t+1} = \phi_t - \nabla t(F_t.|\nabla\phi_t|) \tag{2.10}$$

where F_t and $|\nabla\phi_t|$ indicate the speed function and the magnitude of the level set function at time t, respectively.

In our discussion about level set methods above, we assumed that the level set function ϕ is updated on the grid points in the image domain. Chopp [27] proposed an approach called narrowband method to implement level sets. The proposed approach updates the level set function only at the grid points that are within a certain neighborhood of the zero isocountour. Such points construct a band around the zero level set. We refer the reader to consult the following references for a more detailed information about narrowband methods [28] [10] [11].

2.2 Nonparametric density estimation

Probability density functions have been heavily involved in many statistical analysis problems. For example in Bayesian inference, the posterior density is computed

by using likelihood and prior probability densities. For a particular problem, underlying densities can be used for statistical analysis if they are already known.

The task of density estimation can be divided into two main categories: parametric and nonparametric. Parametric density estimation basically makes an assumption about the underlying density where the mathematical structure of the density is already known, e.g., Gaussian. Then, the task of parametric density estimation is to find the unknown parameters of this particular density, e.g., mean and variance of a Gaussian density. Although parametric density estimation methods are computationally efficient, the assumption about the underlying density may not hold in general in real applications. On the other contrary, nonparametric density estimation methods do not make any assumptions about the underlying probability density. Instead, they learn from a density with unknown structure. Nonparametric density estimation methods suffer from large computational costs, however, they have much more potential to model unknown densities than parametric density estimation methods.

In the following section, we introduce a nonparametric density estimation method called Parzen density estimator. In this thesis, we use Parzen density estimation to estimate shape densities for image segmentation tasks.

2.2.1 Parzen density estimator

The idea of nonparametric density estimation was originally proposed by Parzen [29], Rosenblatt [30] and Cacoullos [31]. In nonparametric density estimation, the problem is to estimate an unknown underlying density $p(x)$ from N i.i.d. samples $x_1, x_2, \ldots, x_N$ drawn from $p(x)$.

Parzen density estimation is a kernel-based density estimation approach given by

$$\widehat{p}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sigma} k(\frac{x - x_i}{\sigma}) \tag{2.11}$$

where $k(.)$ is called a kernel function which satisfies

$$\int k(x) \mathrm{d}x = 1, k(.) \geq 0.$$

The parameter σ is called kernel size, bandwidth, or smoothing parameter. It is very common practice to use a Gaussian density as the kernel in Parzen density estimation, in which case the estimator becomes:

$$\widehat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} k(x - x_i, \sigma) \tag{2.12}$$

where

$$k(x, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-x^2/(2\sigma^2)}.$$

Kernel size is a crucial parameter in Parzen density estimation. By playing with the shape and size of the kernel, different density estimates can be obtained. For example, a larger kernel size will produce a more smooth density estimate whereas a small one will make the density 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 kernel shape [32].

Asymptotically, a good kernel size is expected to decrease as the number of samples grow. In particular, Parzen [29] demonstrated that the following conditions are necessary for asymptotic consistency of the density estimator:

$$\begin{aligned} \lim_{N\to\infty} \sigma &= 0, \\ \lim_{N\to\infty} N\sigma &= \infty, \end{aligned} \tag{2.13}$$

In general, for a d-dimensional random vector, it is known that $\sigma = cN^{-1/(d+4)}$ is asymptotically optimal in density estimation for some constant c [33] [32]. However, for finite N, 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 Kullback-Leibler (KL) divergence $D(p||\widehat{p})$ [34]. Minimizing KL divergence with respect to kernel size σ is equivalent to maximizing

$$\int p(x) \log \widehat{p}(x) \mathrm{d}x.$$

Since we do not know the true density p in advance, we maximize an estimate of this quantity:

$$\begin{aligned}\int p(x)\log\widehat{p}(x)\mathrm{d}x &= E_p[\log\widehat{p}(X)] \\ &\approx \frac{1}{N}\sum_{i=1}^{N}\log\widehat{p}(x_i)\end{aligned} \tag{2.14}$$

Thus the following ML kernel with leave one out becomes a good choice for kernel size σ:

$$\begin{aligned}\sigma_{ML} &= arg\max_{\sigma}\sum_{i}\log\widehat{p}(x_i) \\ &= arg\max_{\sigma}\sum_{i}\log\frac{1}{N-1}\sum_{j\neq i}\frac{1}{\sigma}k(\frac{x-x_i}{\sigma})\end{aligned} \tag{2.15}$$

2.3 Markov chain Monte Carlo (MCMC) methods

In this section, we provide a brief introduction to Markov chain Monte Carlo methods.

2.3.1 Motivation for Monte Carlo sampling

The idea of Monte Carlo was first proposed by Metropolis and Ulam in [35]. We also refer reader to consult the references in [36] and [37].

Let us assume that we are given a set of $N \geq 1$ samples $X_1, \ldots, X_N$ where $X_i \in \mathcal{X} \subset \mathbb{R}^d$ for some $d \geq 1$. Note that the samples are independent and identically distributed (i.i.d.) from an unknown distribution P for a random variable X, i.e.,

$$X_1, \ldots, X_N \overset{i.i.d.}{\sim} P.$$

Let us further assume that we are expected to compute an estimate of the expectation (mean value) of X with respect to P using the samples $X_1, \ldots, X_N$ drawn from P. Given a probability density function, $p(x)$, of P, we can write the expected value as follows:

$$\mathbb{E}_P(X) = \int_{\mathcal{X}} xp(x)\mathrm{d}x. \tag{2.16}$$

An estimate of the expected value can be obtained using the samples as follows:

$$\mathbb{E}_P(X) \approx \frac{1}{N}\sum_{i=1}^{N} X_i. \tag{2.17}$$

Analogously, we can estimate the expectation of a certain function $\Phi : \mathcal{X} \to \mathbb{R}$ with respect to P as

$$P(\Phi) = \mathbb{E}_P(\Phi(X)) = \int_{\S} \Phi(X)p(x)\mathrm{d}x. \tag{2.18}$$

Then, the estimator of the function Φ can be written as the mean of samples evaluated at Φ,

$$\mathbb{E}_P(\Phi(X)) \approx \frac{1}{N}\sum_{i=1}^{N} \Phi(X_i). \tag{2.19}$$

Note that the problem of estimating the mean of a function with respect to P is the generalization of estimating the mean of X with respect to P, this special case is obtained when $\Phi(X) = X$.

When we do not know anything about P but have samples from it, we can estimate the quantity in Equation (2.19). In the case that we explicitly know P, we can exactly calculate the expected value using Equation (2.18).

In this thesis, we deal with problems in which we know P up to some extent but we are not given any samples from it. In this scenario, we can generate i.i.d. samples from P as many as we want. However, we cannot compute the integral in Equation (2.18) or it takes a really long time to compute that we do not want to do. In such cases, the integral is said to be intractable. Therefore, the only option is to generate i.i.d. samples from P to estimate the quantity in Equation (2.19). This simple approach constructs the basis of Monte Carlo methods. Once the samples from a distribution is generated, there is no need to deal with intractable integrals to find an estimate. This brings us to the problem of generating samples from P.

In many problems, sampling from P is not a trivial task. In the literature, there are some methods that exactly generate samples from P. These are the method of inversion [36] [38] and the rejection sampling method [36] [39].

In many real applications, it is very rare to be able to generate exact samples from the desired distribution. We generally encounter with this scenario in Bayesian

inference where the distribution that we want to sample from is the posterior distribution of some variable X given $Y = y$ which can be written as follows:

$$\begin{aligned} p_{X|Y}(x|y) &= \frac{p_X(x)p_{Y|X}(y|x)}{\int p_X(x')p_{Y|X}(y|x')\mathrm{d}x'} \\ &= \frac{p_{X,Y}(x,y)}{\int p_{X,Y}(x',y)\mathrm{d}x'} \\ &\propto p_X(x)p_{Y|X}(y|x) \end{aligned} \tag{2.20}$$

In general, $p_{X|Y}(x,y)$ is either too costly or impossible to perform one of the exact sampling algorithms. Therefore, majority of the efforts have been spent to generate approximate samples in the literature. In this thesis, we exploit from a family of such methods called Markov chain Monte Carlo (MCMC) which we briefly survey in the following section.

2.3.2 Markov chain Monte Carlo (MCMC) methods

An MCMC method is based on a discrete-time ergodic Markov chain which has its stationary distribution as π. There are two widely used MCMC sampling approaches in the literature: Metropolis-Hastings sampling [40] [41] and Gibbs sampling [42] [43].

Markov chain

A stochastic process $\{X_n\}_{n\geq 1}$ on $\mathcal{X}$ is said to be a Markov chain if its probability law defined from the initial distribution $\eta(x)$ and a sequence of Markov transition kernels (probabilities, densities) $\{M_n(x'|x)\}_{n\geq 2}$ define the finite dimensional joint distribution as

$$p(x_1,\ldots,x_n) = \eta(x_1)M_2(x_2|x_1)\ldots M_n(x_n|x_{n-1})$$

for all $n \geq 1$.

The random variable X_t is called the state of the chain at time t and $\mathcal{X}$ is the state-space of the chain.

The definition of the Markov chain leads to the characteristic property of a Markov chain which is called as the weak Markov property. The property states that

the current state of the chain at time n conditioned on its entire history depends only on the previous state at time $n-1$ which can be written as follows:

$$p(x_n|x_{1:n-1}) = p(x_n|x_{n-1}) = M_n(x_{n-1}, x_n).$$

Metropolis-Hastings sampling

The Metropolis-Hastings algorithm requires a Markov transition kernel Q on $\mathcal{X}$ to propose new values from the old ones. Let us assume that $q(.|x)$ is the density of $Q(.|x)$ for any x. A candidate value for x_n given the previous sample x_{n-1} is proposed as $x' \sim q(x_n|x_{n-1})$. The candidate sample x' is accepted with the acceptance probability $\alpha(x_{n-1}, x')$ where the function $\alpha : \mathcal{X} \times \mathcal{X} \to [0, 1]$ is defined as

$$\alpha(x, x') = min\left\{1, \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}\right\}$$

where $x, x' \in \mathcal{X}$. If the acceptance probability $\alpha(x, x')$ is above the threshold u where $u \sim \mathcal{U}(0, 1)$ the candidate x' is accepted such that $x_n = x'$. Otherwise, the candidate is rejected and x_n is assigned to the next state as $x_n = x_{n-1}$.

The Metropolis-Hastings algorithm is given in Algorithm 1.

Algorithm 1 Metropolis-Hastings sampling

Initialize $x_1 \in \mathcal{X}$.
for $n = 2, 3, \ldots$ **do**
    Sample $x' \sim q(x_n|x_{n-1})$.
    Compute $\alpha(x, x') = min\left\{1, \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}\right\}$.
    $u \sim \mathcal{U}(0, 1)$.
    if $\alpha(x, x') > u$ **then**
        $x_n = x'$. ▷ Accept the candidate
    else
        $x_n = x_{n-1}$. ▷ Reject the candidate
    end if
end for

Gibbs sampling

Gibbs sampling [42] [43] is another popular MCMC method which can be used when X has more than one dimension. Let us assume that X has $d > 1$ dimensions such that $X = \{x_1, \ldots, x_d\}$. The Gibbs sampler generates samples from each of the full conditional distributions $\pi_k(x_k|x_{1:k-1}, x_{k+1:d})$. Then, the Gibbs sampler produces a Markov chain by sampling one component, x_k, at a time using the corresponding conditional density π_k. The overall Gibbs sampling algorithm is given in Algorithm 2.

Algorithm 2 Gibbs sampling

1: Initialize $X_1 \in \mathcal{X}$.
2: **for** $n = 2, 3, \ldots$ **do**
3: **for** $k = 1, \ldots, d$ **do** $x_{n,k} \sim \pi_k(x_{n,k}|x_{n,1:k-1}, x_{n-1,k+1:d})$
4: **end for**
5: **end for**

Chapter 3

Nonparametric Joint Shape and Feature Priors for Image Segmentation

Segmentation of images that include limited and low quality data is a challenging problem and requires prior information about the shape to be segmented for an acceptable solution. For example, given a training set of car shapes, a partially occluded car object in an image can be segmented by exploiting prior shape information obtained from the training set. The problem becomes more complex when the training set of shapes involves examples from multiple classes (e.g., car, truck, plane, etc.) leading to a multimodal shape density. In this work, we focus on segmentation problems in which shape distributions are multimodal and complex, but just the shape prior information is not sufficient for effective segmentation due to, e.g., severe occlusion. The proposed approach deals with the problem by incorporating discriminative class-dependent feature priors together with shape priors into the segmentation process. We demonstrate that the proposed approach overcomes the limitations of existing segmentation methods that use only shape priors. The method introduced in this chapter has been published in [44].

3.1 Related work

One of the earliest attempts to include a prior information in image segmentation is the active contour model, also called "snakes", by Kass et al. [9]. Snakes use a general regularity term as the prior, where the roughness and length of the curve serve as a penalty, which is based on the assumption that smoother and shorter curves are more likely [1]. However, in many applications a more informative object-

type specific shape prior can be learned from training samples. In this regard, active shape models (ASM) proposed by Cootes et al. [45] are powerful techniques for segmentation using shape priors. Variants of the ASM, their applications to different image segmentation areas, and a review can be found in [46–50].

In the original ASM, a training set of shapes represented by landmarks is used to construct allowable shape variations via principal component analysis (PCA). The use of linear analysis tools such as PCA in ASMs limits the domain of applicability of these techniques to shape priors involving only unimodal densities. That is, the original ASMs assume that the training shapes are distributed according to a unimodal, Gaussian-like distribution; hence, the technique cannot model more complex (multimodal) shape distributions.

Several methods have been proposed to handle multimodal distributions of shapes by extending ASMs [12–14]. These approaches include the use of mixture of Gaussians [12], manifold learning techniques [13] and kernel PCA [14, 51]. However, these approaches use parametric probability distributions, which may not model very complex shape variations [52]. In addition, the explicit (landmark-based) shape representation used in ASMs has two major shortcomings. First, annotating landmark points with correct correspondences across all example shapes can be difficult and time consuming. Second, the extensions of the technique to handle topological changes are not straightforward. To overcome the limitations of landmark-based representation, level set based shape priors were proposed [15, 16]. Because of their implicit nature, level set methods do not need landmarks and can easily handle topological changes [53, 54]. In [15] and [16], shape variability is captured using PCA on signed distance functions of level sets. However, these techniques work well only when the shape variation is small due to their use of PCA. Therefore, they cannot handle multimodal shape densities.

In order to learn multimodal shape densities, Kim et al. [1] and Cremers et al. [17] proposed nonparametric density estimation based shape priors using level sets. These methods estimate the prior shape density by extending Parzen density estimator over the distances between the level set representations of the evolving curve and training shapes. These ideas have also been extended to the problem of segmenting multiple objects through the use of coupled shape priors [18]. An

interesting usage of nonparametric shape priors proposed by Foulonneau et al. [2] computes Legendre moments from binary images as shape descriptors and uses distances between descriptors instead of level sets for estimating the prior shape density. The approach also exploits appealing properties of Legendre moments for intrinsic alignment. The approaches of Kim et al.[1], Cremers et al. [17] and Foulonneau et al. [2] use a simple data term that assumes the foreground and the background intensities are piecewise-constant [55]. In the literature, there are also methods that combine nonparametric shape priors with learning-based data terms [3, 56, 57]. Using a more sophisticated data term significantly improves the segmentation quality when the object foreground and background have complex densities. Some other recent work that exploits nonparametric shape priors and a more detailed review of the level set based segmentation methods can be found in [19–21, 58–61].

3.2 Motivation

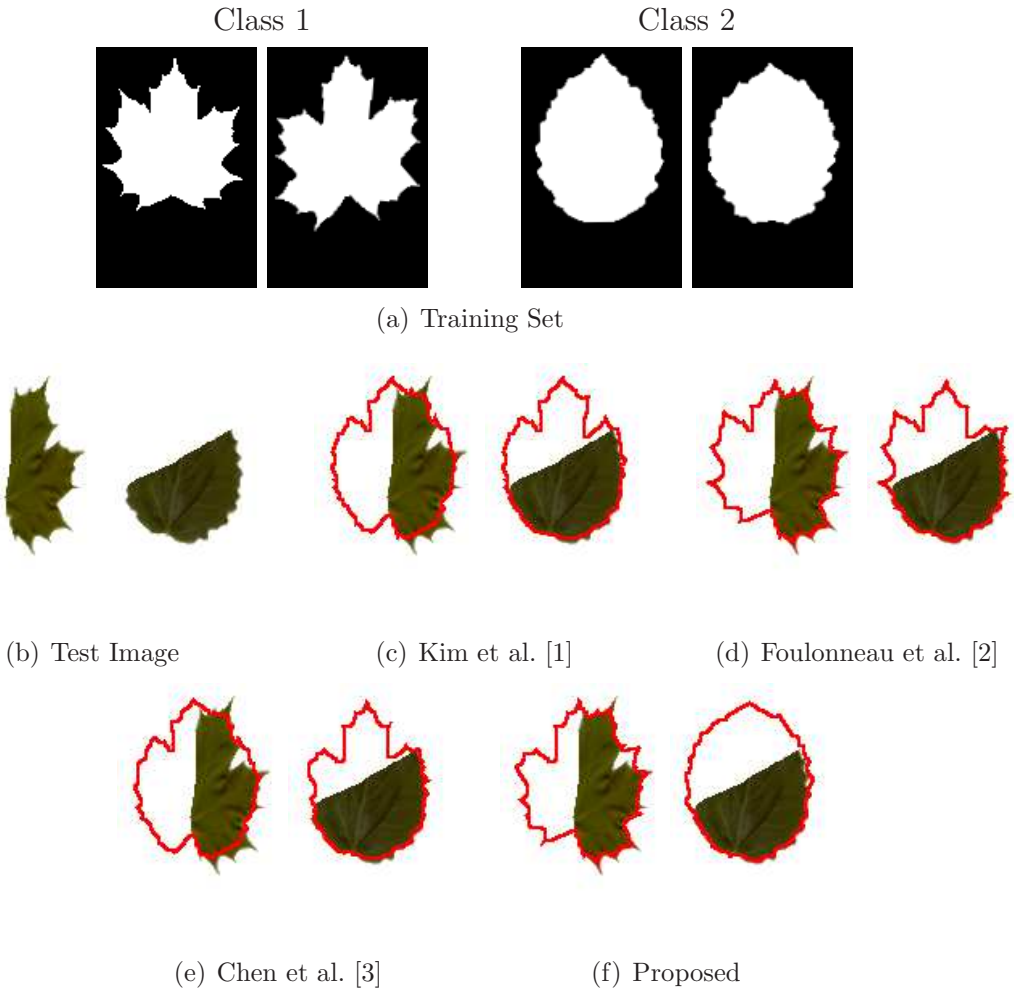
The methods [1–3, 17, 56, 57] that use nonparametric shape priors performs well in the presence of occlusion and missing data. They also capable of handling multimodal shape densities. However, the shortcomings of these methods arises when the level of occlusion and missing data increases and when the underlying shape density is multimodal. This is due to the fact that the prior density is estimated by extending Parzen density estimator over the distances between the evolving curve and training shapes. These methods use gradient descent to minimize an energy function including data and shape priors terms. During gradient descent, a curve represented by level sets is evolved by a data-driven force together with the weighted average of the training shapes where the weight of each training shape is usually inversely proportional to its distance to the evolving curve (the exact form of the weights is determined by the specific metric used to measure distances between shapes). Therefore, when the observed data are very limited, the evolving curve can be more similar to training shapes from a different class based on the distance metric. In these cases, the evolving curve is driven toward a shape from a different mode of the shape density, which yields inaccurate segmentation results.

We illustrate the aforementioned drawback of Kim et al. [1], Foulonneau et al. [2]

and Chen et al. [3] through the example shown in Figure 3.1[1]. In this example, we use a training set that contains samples from two different leaf shape classes as shown in Figure 3.1(a). Note that the boundaries of the leaf shapes are uneven in class 1 and smooth in class 2. We have 2 test images from each class as shown in Figure 3.1(b). Note that the test images are severely occluded; almost half of the leaf shapes do not appear. Since the curve found by the data term is more similar to class 2 based on the distance metric, Kim et al. [1] produce segmentation results that are more similar to the shapes in class 2 in both test images. The major difference between Chen et al. [3] and Kim et al. [1] is the design of the data term. Since the data provide very little information in the test images, the effect of the data term is very limited in the segmentations. Therefore, Chen et al. [3] produce very similar results with Kim et al. [1] as shown in Figure 3.1(e). The method of Foulonneau et al. [2] produces segmentation results that are more similar to the shapes in class 1 (see Figure 3.1(d)). This means that estimating the prior shape density based on the distances between Legendre moments does not help to have segmentation results from the correct mode of the shape density in the presence of severe occlusion.

This motivates us to deal with the shortcomings of the existing methods by incorporating discriminative class-dependent features to the kernel density estimation process. For example, circularity of the shapes in Figure 3.1 is an important feature for identifying different leaf classes. In such cases, jointly estimating feature and shape prior density can yield more accurate segmentations as shown in Figure 3.1(f).

[1]Note that these three methods are representative ones; Kim et al. [1] estimate prior density using distances between shapes, Foulonneau et al. [2] estimate prior density using distances between Legendre moments and Chen et al. [3] use intensity prior-based data term together with the shape prior term. The other nonparametric shape prior-based methods exhibit a similar behavior with one of these methods.



(a) Training Set

(b) Test Image (c) Kim et al. [1] (d) Foulonneau et al. [2]

(e) Chen et al. [3] (f) Proposed

Figure 3.1: Toy example that demonstrates motivation of the proposed method.

3.3 Contributions

Our contribution in this work is a segmentation algorithm that performs segmentation by exploiting nonparametric joint shape and feature priors. Unlike the state-of-the-art methods that perform segmentation using nonparametric shape density estimation, we exploit learned discriminative class-dependent features (geometric or appearance-based) extracted from specific parts of the scene relative to the object of interest and incorporate the joint shape and feature prior density into the segmentation process. In particular, we combine a data term and a joint shape and feature prior term within a Bayesian framework to form the energy functional for segmentation. To the best of our knowledge, nonparametric joint shape and feature priors have not been proposed for image segmentation in the literature. By estimating a

more discriminative prior density, our algorithm is able to find better segmentations based on the shape posterior density.

Our approach may seem similar to the methods proposed by Cremers et al. [62] and Chan et al. [63]. However, these approaches and the proposed approach focus on completely different problems. In [62] and [63], given a scene with multiple different types of objects, the problem is to segment a particular object that is included in the training set. In this work, we focus on the problem of segmenting an object using the correct shape priors when the training set contains shapes from different classes.

A precursor of this work were presented in [64]. The approach in [64] considers the problem of segmenting objects having multimodal shape densities as a joint classification and segmentation problem. The method gives a hard classification decision at some stage of the segmentation process by extracting some features. Once the class decision is made, the curve evolution process continues by using the training shapes in this particular class. The major drawback of the approach in [64] is that the outcome of the segmentation is highly depend on the classification decision which forces the algorithm to produce a segmentation result from the class.

Preliminary results of this work were presented in [65]. The proposed work advances its preliminary version in several major ways. In particular, (1) while [65] was focused on the specific problem of spine segmentation, in this work we significantly expand the domain of applicability of this new idea; (2) we consider and use new types of features in our framework; (3) we present the results of an expanded experimental analysis on a variety of data sets, together with quantitative comparison to the results of several state-of-the-art methods; (4) we provide a more detailed technical development and discussion of the proposed method; (5) we present an expanded coverage of related work.

3.4 The proposed method

3.4.1 The energy function

In this section, we propose an energy function that exploits nonparametric joint shape and feature priors for image segmentation. Let c be the evolving curve, f

be the feature vector and y be the intensity image. Then, the posterior probability density function of c and f can be written using Bayes' rule as follows:

$$p(c, f|y) = \frac{p(y|c, f)p(c, f)}{p(y)} \tag{3.1}$$

where,

$$p(y|c, f) = \frac{p(f|y, c)p(y|c)}{p(f|c)}. \tag{3.2}$$

Plugging in Equation (3.2) into (3.1) yields

$$p(c, f|y) \propto p(f|y, c)p(y|c)p(c) \tag{3.3}$$

and $p(c)$ can be written as

$$p(c) = \int p(c, f)\,\mathrm{d}f. \tag{3.4}$$

Then, Equation (3.3) becomes

$$p(c, f|y) \propto p(y|c)p(f|y, c) \int p(c, f)\,\mathrm{d}f. \tag{3.5}$$

Let us assume that we observe a feature vector $\hat{f}$ either from data or from boundary. Such features could involve geometric, textural, or appearance-based information about the object to be segmented. From this point on, one can proceed with various assumptions on the probability densities involved. For feature extraction, we assume that features can be extracted perfectly based on the data as well as information about the boundary when it reaches a reasonable state. This leads to the degenerate density:

$$p(f|y, c) = \delta(f - \hat{f}) \tag{3.6}$$

where, $\delta(.)$ is the Dirac delta function. Also, we learn $p(c, f)$ nonparametrically from the training data. Since $\hat{f}$ is already observed, Equation (3.5) can be written as follows:

$$p(c, \hat{f}|y) \propto p(y|c)p(c, \hat{f}). \tag{3.7}$$

Note that, $p(c, \hat{f})$ is also equivalent to the slice of $p(c, f)$ at $\hat{f}$ which is $p(c|f = \hat{f})$. Therefore, Equation (3.7) and the following equation are identical.

$$p(c, \hat{f}|y) \propto p(y|c)p(c|f = \hat{f}). \tag{3.8}$$

In this work, we use level sets to represent c. Level set representation is essentially a mapping

$$\phi : \{0, 1\}^{M \times N} \rightarrow \mathbb{R}^{MN}$$

from the binary space to the real space. In the literature, it has been found more convenient to work with level sets to represent c to handle topological changes and its effectiveness when computing gradients. In the rest of this chapter, we work with $x = \phi(c)$. Therefore, Equations (3.7) and (3.8) becomes

$$p(x, \hat{f}|y) \propto p(y|x)p(x, \hat{f}) \tag{3.9}$$

and

$$p(x, \hat{f}|y) \propto p(y|x)p(x|f = \hat{f}). \tag{3.10}$$

Hence, given the simplifying perfect feature extraction assumption in Equation (3.6), the learned joint shape and feature density is used through conditioning on the extracted feature. This conditioning guides the segmentation process, possibly towards the correct mode of the multimodal shape density. If needed, one could certainly relax this assumption in our framework, and develop an optimization algorithm for maximizing the posterior density in Equation (3.1) to infer both the feature and the shape based on the data and the learned joint prior.

The data term we use is the piecewise-constant version of the Mumford-Shah functional [55, 66]. We use this data term as a representative one, since it has been previously used in a variety of applications [17, 64]. One can consider using more sophisticated data terms such as those involving mutual information [67], J-Divergence [68], and Bhattacharya distance [69]. We discuss estimating the joint shape and feature prior density, $p(x, \hat{f})$, in the following section.

By simply taking the negative logarithm of Equation (3.7), we can define the following energy function to be minimized for segmentation.

$$\begin{aligned} E(x,\hat{f}) &= -\log p(y|x) - \log p(x,\hat{f}) \\ &= \beta\Big[\int_{c_{in}} (y(x)-m_{in})^2 \mathrm{d}x + \int_{c_{out}} (y(x)-m_{out})^2 \mathrm{d}x\Big] - \log p(x,\hat{f}) \end{aligned} \tag{3.11}$$

where $y(.)$ is the intensity image, c_{in} (c_{out}) is the region inside (outside) of the segmenting curve x, m_{in} (m_{out}) is the average intensities in these regions, and β is a constant that determines the balance between the data and the prior terms which we set $\beta = 1$.

3.4.2 Building joint shape and feature priors

Let us assume that we have n aligned training shapes $\mathbf{c} = \{c_1, c_2, \ldots, c_n\}$, their level set representations $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ and a corresponding set of feature vectors $\mathbf{f} = \{f_1, f_2, \ldots, f_n\}$ extracted from intensity images. The basic idea we use is that the segmenting curve x will be more likely if it is similar to the training shapes and $\hat{f}$ is similar to the training feature vectors. In order to measure the similarity between curves, we need to compare c with the training shapes in $\mathbf{x}$. However, when x and the training shapes in $\mathbf{x}$ are not aligned, a direct comparison of x with the shapes in $\mathbf{x}$ includes not only shape differences but also artifacts due to pose difference such as translation, rotation, and scaling. In order to remove pose artifacts, we align x with the shapes in $\mathbf{x}$ into $\tilde{x}$, where $\tilde{x}$ is the aligned version of x. Also, recall that shapes in $\mathbf{x}$ are already aligned. Similarly, in order to extract pose invariant features, all feature vectors should be extracted after alignment. Any kind of rigid alignment approach can be used to obtain an aligned training set of shapes from its unaligned version for which we use the approach proposed by Tsai et al. [15]. Then, the joint shape and feature density is estimated using Parzen density estimation as follows[2]

$$p(\tilde{x}|f=\hat{f}) \propto p(\tilde{x},\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} k(d(\tilde{x},x_i), d(\hat{f},f_i), \sigma_x, \sigma_f) \tag{3.12}$$

[2]Note that in Parzen density estimation, class labels of the shapes in the training set are not available.

where $d(\cdot,\cdot)$ is a distance metric, $k(\cdot,\cdot,\sigma_x,\sigma_f)$ is a 2D kernel with shape kernel size σ_x and with feature kernel size σ_f. For the kernel sizes σ_x and σ_f, we use an ML kernel with leave-one-out [32]. Note that, the composite of the 2D kernel and the distance metrics plays the role of an infinite dimensional kernel. A variety of distance metrics can be used in Equation (3.12) [1]. In our experiments, we use the template distance metric [1], d_T, for shape distance and the L_2 distance metric, d_{L_2}, for feature distance.

Note that, we compute the joint shape and feature prior density for the aligned curve, $\tilde{x}$, in Equation (3.12) to remove the pose artifacts as we mentioned above. We explain how to relate $p(\tilde{x},\hat{f})$ to $p(x,\hat{f})$ in our segmentation method in the following section.

3.4.3 Segmentation algorithm

The aim of the proposed segmentation approach is to minimize the energy functional in Equation (3.11) by gradient descent, and the task comes down to computing the gradient flow for the curve c. The overall gradient flow is the sum of the two terms, one based on the data term and the other based on the shape and feature prior term. The gradient flow for the data term is given by

$$\frac{-\partial \log p(y|x)}{\partial x} = \beta\Big[-(y(x)-m_{in})^2 + (y(x)-m_{out})^2\Big]\vec{N}, \tag{3.13}$$

where $\vec{N}$ is the outward curve normal [55].

However, we cannot compute $\frac{\partial \log p(x,\hat{f})}{\partial c}$ directly from the shape and feature prior term due to the need for removing pose differences mentioned in Section 3.4.2. Instead, we first compute $\frac{\partial \log p(\tilde{x},\hat{f})}{\partial \tilde{x}}$ and relate it to $\frac{\partial \log p(x,\hat{f})}{\partial x}$. The gradient flow $\frac{\partial \log p(\tilde{x},\hat{f})}{\partial \tilde{x}}$ for the joint shape and feature prior term is given by

$$\begin{aligned}\frac{\partial \log p(\tilde{x},\hat{f})}{\partial \tilde{x}} &= \frac{1}{p(\tilde{x},\hat{f})} \times \frac{1}{n} \times \frac{1}{\sigma_x \times \sigma_f} \\ &\quad \times \sum_{i=1}^{n} \Big(k(d_T(\tilde{x},x_i), d_{L_2}(\hat{f},f_i), \sigma_x, \sigma_f) \\ &\quad \times d_T(\tilde{x},x_i) \times (d_{L_2}(\hat{f},f_i))^2 \times (1-2H(x_i)) \Big).\end{aligned} \tag{3.14}$$

where $H(\cdot)$ is the Heaviside function. The derivation of the gradient flow in Equation (3.14) is a straightforward extension of the derivation in [1] and is given in Appendix 7.1.

In order to compute $\frac{\partial \log p(x,\hat{f})}{\partial x}$ from $\frac{\partial \log p(\tilde{x},\hat{f})}{\partial \tilde{x}}$, we need a pose parameter, $\mathbf{p}$, that aligns C with the shapes $\mathbf{x}$ into $\tilde{x}$ in each iteration of the gradient descent (see line 10 in Algorithm 3). After $\frac{\partial \log p(\tilde{x},\hat{f})}{\partial \tilde{x}}$ is computed (see line 12 of Algorithm 3), $\frac{\partial \log p(x,\hat{f})}{\partial x}$ is obtained by applying reverse transformation with pose parameters $\mathbf{p}$ to the force $\frac{\partial \log p(\tilde{x},\hat{f})}{\partial \tilde{x}}$ (see line 13 in Algorithm 3). In other words, gradient of the shape and feature prior is computed for $\tilde{x}$, gradient force is reverse back to its original pose and the whole gradient update is performed. Note that the alignment process can be done intrinsically during the curve evolution as in [2, 17]. We choose to perform this process explicitly as in [1].

Finally, the proposed segmentation method that exploits nonparametric joint shape and feature priors is given in Algorithm 3.

3.5 Experimental results

In this section, we present experimental results on 4 different data sets using various discriminative class-related features. In the MNIST and the aircraft data sets, features are synthetically generated. The remaining 2 data sets, the Swedish leaf and the dendritic spines, are completely real data sets.

We compare the performance of the proposed approach with three different methods: Kim et al. [1], Foulonneau et al. [2] and Chen et al. [3]. We obtain quantitative results by comparing segmentation results with ground truths using Dice scores [70] and Hausdorff distance [71]. Dice score takes values between 0 and 1 where 1 indicates a perfect match whereas low values of Hausdorff distance indicate better results.

3.5.1 MNIST handwritten digits data set

In this section, we present experimental results on 3 different settings of the MNIST handwritten digits data set. We use the shapes in the training set shown in Figure 3.2 in all experimental settings. Experimental settings differ from each

Algorithm 3 Segmentation using nonparametric joint shape and feature priors

1: Initialize x

2: **for** $t = 0 \rightarrow t_{data}$ **do** $\triangleright$ t_{data} : time when the data driven curve evolution converges

3: **if** $t = t'$ **then** $\triangleright$ t' : time when the feature is extracted

4: Align x with the shapes in $\mathbf{x}$ into $\phi_{\tilde{x}}$.

5: Extract feature vector $\hat{f}$.

6: **end if**

7: Update x with the data force given in Equation (3.13).

8: **end for**

9: **for** $t = t_{data} + 1 \rightarrow t_{converge}$ **do** $\triangleright$ $t_{converge}$: time when data + joint shape and feature priors driven curve evolution converges

10: Align x with the shapes in $\mathbf{x}$ into $\tilde{x}$.

11: Compute the data force for x using the Equation (3.13).

12: Compute the joint shape and feature force $\frac{\partial \log p(\tilde{x},\hat{f})}{\partial \tilde{x}}$ as given in Equation (3.14).

13: Reverse the force $\frac{\partial \log p(\tilde{x},\hat{f})}{\partial \tilde{x}}$ to its original pose $\frac{\partial \log p(x,\hat{f})}{\partial x}$ using the reverse pose parameters found in step 10.

14: Update x with the sum of the data force computed in step 13 and the joint shape and feature force computed in step 12.

15: **end for**

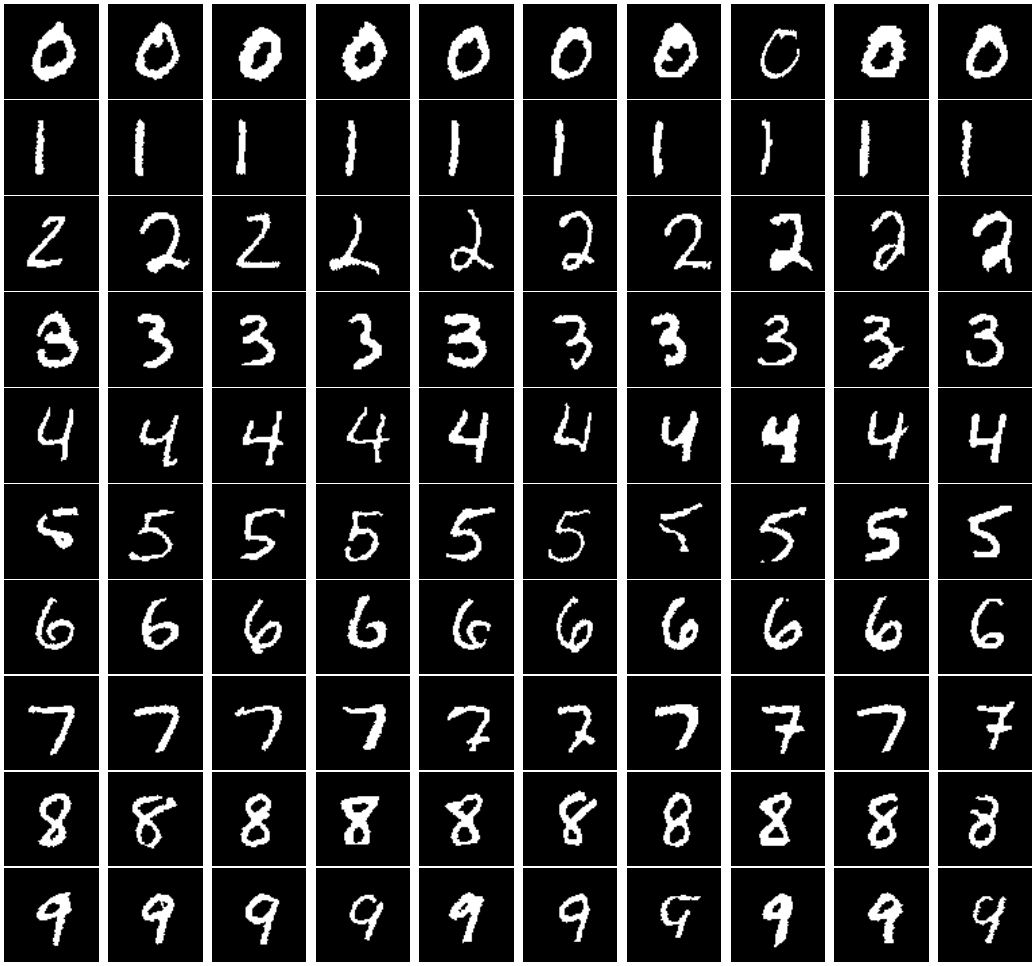


Figure 3.2: Training set of shapes for the MNIST handwritten digits data set

other in terms of the feature vectors that are exploited for segmentation. This experiment demonstrates that our approach can learn effectively from a relatively small training data set. The approach could also exploit information in larger data sets when available.

In the first experimental setting of the MNIST data set, each training shape in Figure 3.2 is obtained from an intensity image which contains gray-level intensities drawn from a Gaussian distribution with different means for different classes in foreground regions. One exemplary intensity image from each digit class is shown in Figure 3.3. We estimate the mean intensity value in the foreground region using the corresponding intensity images of each training set. We use the mean values to form the training set of feature vectors $\mathbf{f}$. We perform experiments on the test images shown in the second row of Figure 3.4. In all test images, we first segment the apparent part of the object using only the data term (lines 2 - 8 in Algorithm 3).

Figure 3.3: Training sets that are used to obtain feature vectors. First row: the first training setting in which each digit class contains gray-level intensities drawn from a Gaussian distribution with different means in foreground region, second row: the second training setting in which each digit class contains different colors in foreground region, third row: the third training setting in which each digit class contains different colors in background region. Note that our training sets to obtain feature vectors contain 10 samples for each class and we display only one sample from each class for the sake of brevity.

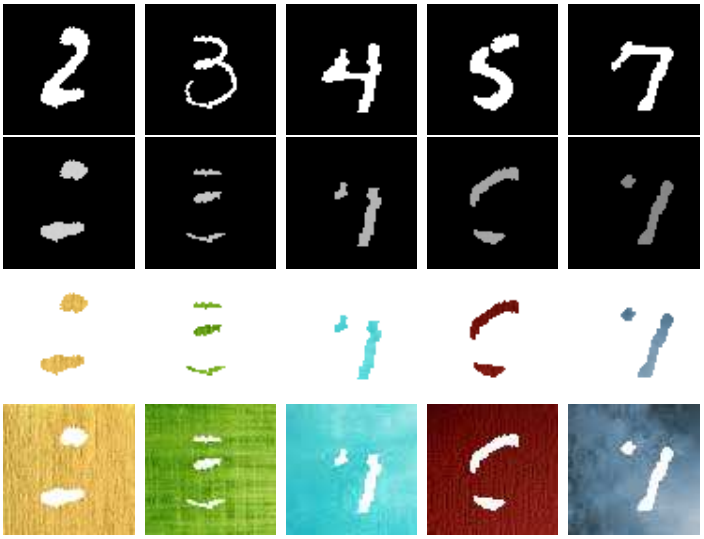


Figure 3.4: Test images for the MNIST data set. First row: ground truth, second row: the first experimental setting, third row: the second experimental setting, fourth row: the third experimental setting.

Then, the feature vector $\hat{f}$ is extracted as the mean intensity value in the foreground region of the initial segmentation. Note that, in this experimental setting the feature vector $\hat{f}$ and the feature vectors in $\mathbf{f}$ contain a scalar value. Also, note that the extracted feature value strongly depends on the data driven (initial) segmentation. Then, we keep evolving the curve using the nonparametric shape and feature priors together with the data term (lines 9 - 15 in Algorithm 3). We also perform experiments on the same test images using the approaches of Kim et al. [1], Foulonneau et al. [2] and Chen et al. [3]. Visual segmentation results of all approaches are shown in Figure 3.5. The visual results demonstrate that the proposed approach generates segmentations that are closer to the ground truths whereas the other methods converges to a wrong mode of the posterior shape density in most test images. We also provide quantitative comparisons of the segmentation results with respect to ground truth using Dice score (see Table 3.1) and Hausdorff distance (see Table 3.2). The quantitative results with both metrics demonstrate the potential of the proposed approach.

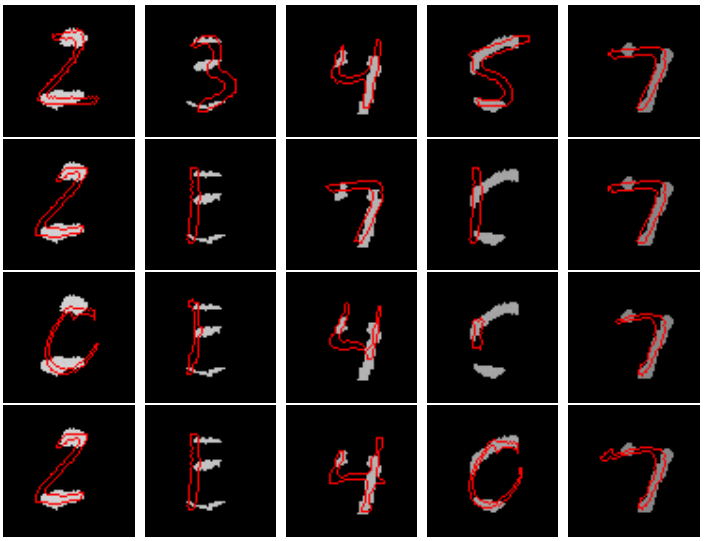


Figure 3.5: Visual results of the first experimental setting of the MNIST data set. First row: the proposed method, second row: Kim et al. [1], third row: Foulonneau et al. [2], fourth row: Chen et al. [3].

In the second experimental setting of the MNIST data set, intensity images of the training shapes in Figure 3.2 contain different colors in foreground regions for different classes as shown in the second row of Figure 3.3. In this experiment, each feature vector is obtained by concatenating RGB histograms computed from the

Table 3.1: Dice score results on the first experimental setting of the MNIST data set.

Digit	2	3	4	5	7
Proposed	**0.6217**	**0.4341**	**0.7167**	**0.7906**	0.6809
Kim et al. [1]	0.5736	0.1771	0.4738	0.2294	**0.6870**
Foulonneau et al. [2]	0.3456	0.1814	0.6040	0.2298	0.6308
Chen et al. [3]	0.5736	0.1732	0.7042	0.4822	0.5915

Table 3.2: Hausdorff distance results on the first experimental setting of the MNIST data set.

Digit	2	3	4	5	7
Proposed	8.000	**11.313**	**5.385**	**6.082**	**6.082**
Kim et al. [1]	**5.656**	20.000	13.601	20.000	7.000
Foulonneau et al. [2]	11.313	20.000	12.083	20.124	8.246
Chen et al. [3]	**5.656**	20.000	**5.385**	10.1980	7.211

foreground region of the corresponding intensity image. All training feature vectors in $\mathbf{f}$ are constructed by following the same procedure. We use 5 test images shown in the third row of Figure 3.4 in this experiment. Similar to the previous experiment, we find the apparent part of the digits using only the data term. Then, we compute the RBG histograms from the intensities that lie inside the segmenting curve and form $\hat{f}$ by concatenating the histogram of each color channel. Then, we continue the curve evolution using our shape and feature-based segmentation approach. Visual segmentation results of the proposed approach and the all competing approaches are shown in Figure 3.6. We also provide the Dice score results in Table 3.3 and Hausdorff distance results in Table 3.4. The results clearly show the superiority of our approach with respect to other approaches.

Finally, in the third experimental setting, we design an experimental setting similar to the second one. In this setting, background regions contain different colors for each digit classes as shown in the third row of Figure 3.3. Similar to the second experimental setting, we construct $\mathbf{f}$ by exploiting the RGB histograms from the intensity images that correspond to background regions. We use the test images given

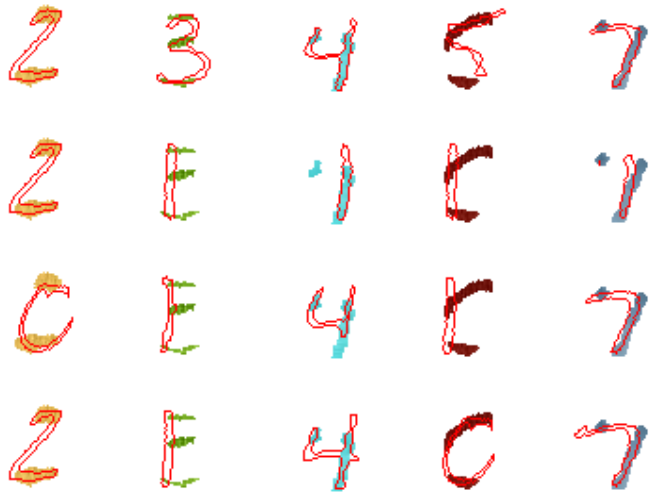


Figure 3.6: Visual results of the second experimental setting of the MNIST data set. First row: the proposed method, second row: Kim et al. [1], third row: Foulonneau et al. [2], fourth row: Chen et al. [3].

Table 3.3: Dice score results on the second experimental setting of the MNIST data set.

Digit	2	3	4	5	6
Proposed	**0.5790**	**0.5690**	**0.7458**	**0.5313**	**0.7032**
Kim et al. [1]	0.5736	0.1699	0.5492	0.2770	0.4751
Foulonneau et al. [2]	0.3446	0.1814	0.5743	0.2192	0.6570
Chen et al. [3]	0.5736	0.1732	0.7042	0.4891	0.5915

in the fourth row of Figure 3.4. In all test images, once we find the apparent boundaries using the data term, we extract $\hat{f}$ by computing the RGB histograms from the background region and concatenating them into a single feature vector. As in the above experiments, the proposed approach achieves better segmentation results than the approaches we compete both visually (see Figure 3.7) and quantitatively (see Tables 3.5 and 3.6).

3.5.2 The Swedish leaf data set

In this section, we present evaluations of the proposed approach on the Swedish leaf data set [72]. The Swedish leaf data set contains leaf images obtained from 15

Table 3.4: Hausdorff distance results on the second experimental setting of the MNIST data set.

Digit	2	3	4	5	6
Proposed	**5.656**	**6.324**	**5.385**	14.317	8.246
Kim et al. [1]	**5.656**	19.416	23.194	18.248	10.816
Foulonneau et al. [2]	11.313	20.000	12.083	20.000	7.000
Chen et al. [3]	**5.656**	20.000	**5.385**	**10.198**	**7.211**

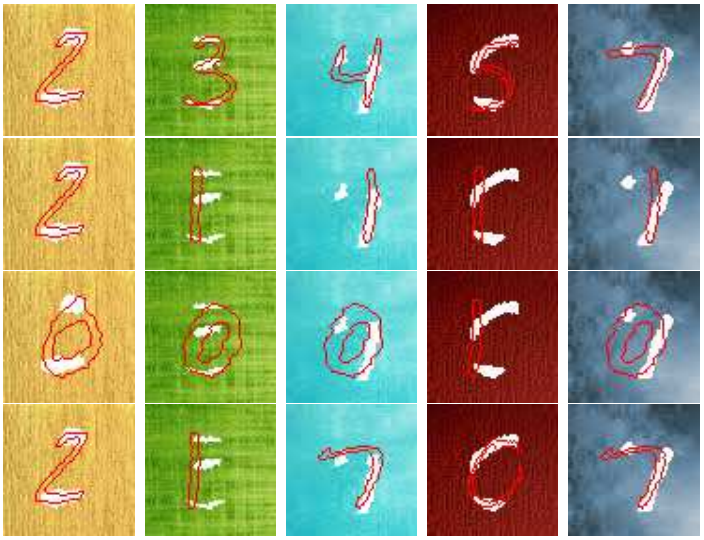


Figure 3.7: Visual results of the third experimental setting of the MNIST data set. First row: the proposed method, second row: Kim et al. [1], third row: Foulonneau et al. [2], fourth row: Chen et al. [3].

different tree classes. We choose two classes among them: Acer and Populus tremula. The data set is designed for classification purposes and it only contains RGB leaf images. We obtain binary images that are used for training by manually segmenting 10 leaf images from each class as shown in Figure 3.8. In order to construct a training set of feature vectors $\mathbf{f}$, we compute circularity of the boundaries in each binary training shape. Circularity of the boundary is a discriminative geometric feature for Acer and Populus tremula classes.

We perform experiments on 10 test leaf images (5 test images from each class and none of which is included in the training set), shown in Figure 3.9. Similar to the previous experiments, we find the apparent boundaries using only the data

Table 3.5: Dice score results on the third experimental setting of the MNIST data set.

Digit	2	3	4	5	7
Proposed	**0.5736**	**0.5809**	**0.7093**	**0.5949**	**0.6779**
Kim et al. [1]	**0.5736**	0.1695	0.5510	0.2766	0.4388
Foulonneau et al. [2]	0.5018	0.4016	0.5490	0.2192	0.4889
Chen et al. [3]	0.5736	0.1732	0.4369	0.4822	0.5915

Table 3.6: Hausdorff distance results on the third experimental setting of the MNIST data set.

Digit	2	3	4	5	6
Proposed	**5.656**	**6.403**	**5.099**	**5.000**	**7.000**
Kim et al. [1]	**5.656**	19.416	23.086	18.248	20.223
Foulonneau et al. [2]	12.806	7.071	12.165	20.000	15.132
Chen et al. [3]	**5.656**	20.000	12.649	10.198	7.211

term and set $\hat{f}$ as the circularity of the boundary. Visual segmentation results of all approaches are shown in Figure 3.10. The visual results demonstrate that the approaches of Kim et al. [1] and Chen et al. [3] tends to drive the segmenting curve toward a shape from Populus tremula class in all test images. Unlike Kim et al. [1] and Chen et al. [3], the method of Foulonneau et al. [2] converges to the mode that corresponds to Acer class in all test images. With the aid of using the discriminative feature priors along with the shape priors, the proposed approach achieves segmentations from the correct mode of the shape density. The Dice score results are 0.9409 for the proposed method, 0.9456 for the method of Kim et al. [1], 0.9030 for the method of Foulonneau et al. [2] and 0.9335 for the method of Chen et al.[3] on average of 10 test images. The average Hausdorff distance results are 10.5742 for the proposed method, 12.7036 for the method of Kim et al. [1], 17.0145 for the method of Foulonneau et al. [2] and 13.6214 for the method of Chen et al. [3]. Note that Dice score results are close to each other even the competing methods produce segmentations from a wrong mode of the shape density. Since the shapes in different classes are very similar and Dice score measures the overlap between the

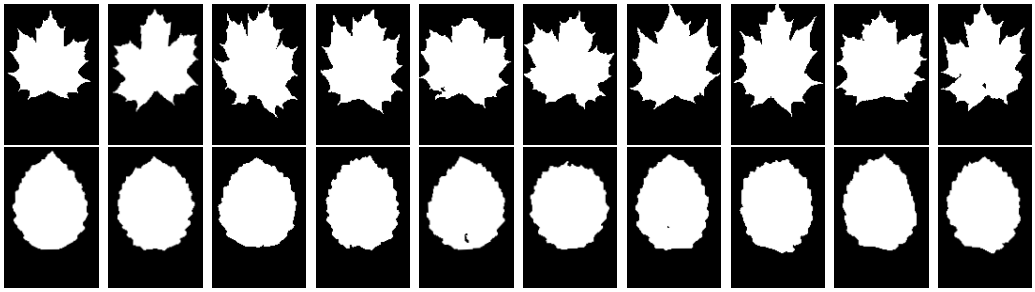


Figure 3.8: Training set of shapes for the Swedish leaf data set. First row: Acer, second row: Populus tremula.

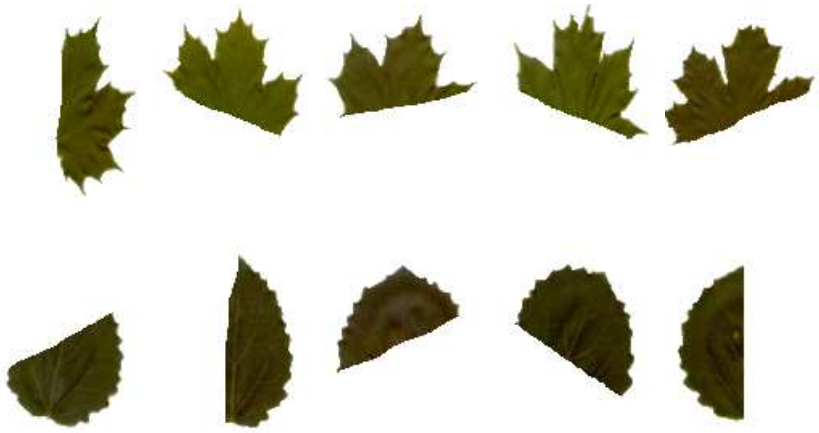


Figure 3.9: Test images for the Swedish leaf data set. First row: Acer, second row: Populus tremula.

segmentation and the ground truth, these results are expected. Hausdorff distance better quantifies the difference in the visual results in this experiment.

3.5.3 The airplane data set

In this section, we evaluate the performance of our segmentation approach on the airplane data set [73]. The airplane data set contains 7 different airplane classes. In our experiments, we take a subset of two of them: F-14 wings opened and Harrier. We use 10 airplane shapes from each class for training as shown in Figure 3.11. Each airplane training shape in Figure 3.11 is obtained from an intensity image as shown in Figure 3.12. Note that, in Figure 3.12, airplane shapes from different classes contain different textural foreground regions. This means that textural features obtained from the foreground region can be discriminative class-dependent features

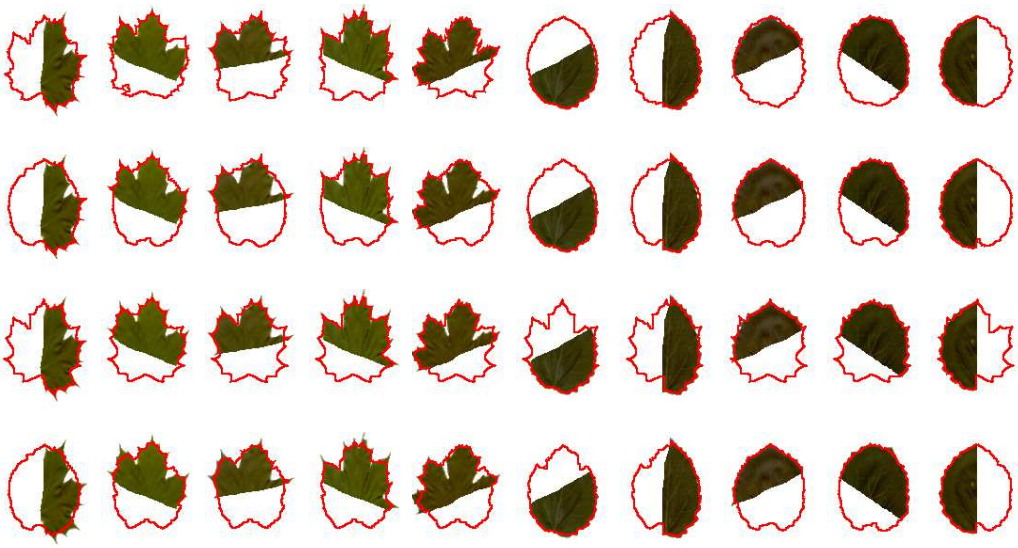


Figure 3.10: Visual segmentation results on the Swedish leaf data set. First row: proposed method, second row: Kim et al. [1], third row: Foulonneau et al. [2], fourth row: Chen et al. [3].

for this data set. For each training shape, we extract 3 different textural features from the foreground region: correlation, energy, and homogeneity. We form each feature vector f_i in $\mathbf{f}$ by concatenating these values into a single vector.

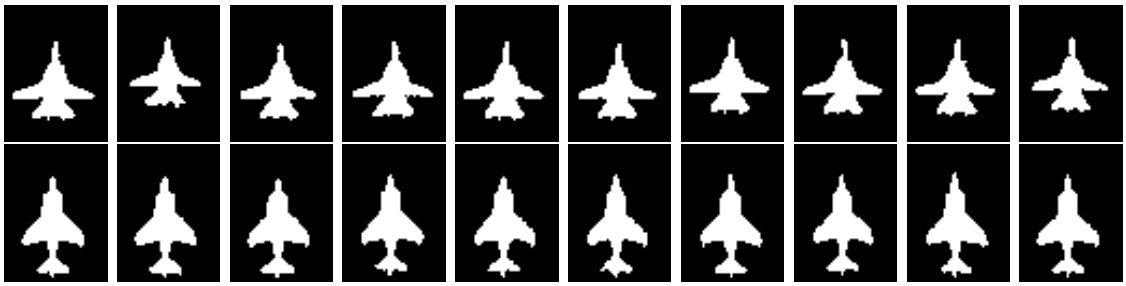


Figure 3.11: The airplane data set. First row: F-14 wings opened, second row: Harrier.

We compare the performance of the proposed approach with Kim et al. [1], Foulonneau et al. [2] and Chen et al. [3] on 10 test images shown in Figure 3.13. Note that the test images are not included in the training set. When segmenting test images, we extract three textural features (correlation, energy, and homogeneity) after the data driven segmentation and concatenate into a single vector $\hat{f}$. Visual segmentation results on the airplane data set are shown in Figure 3.14. According to the visual results, the proposed approach drives the segmenting curve toward the

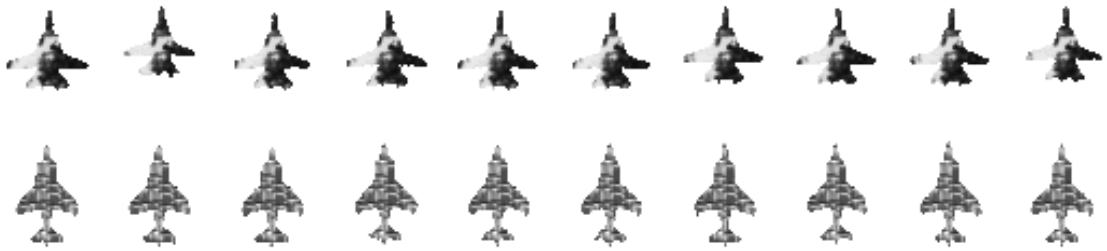


Figure 3.12: Training set that are used to obtain the feature vectors. Note that each airplane shapes from different classes contain different textures.

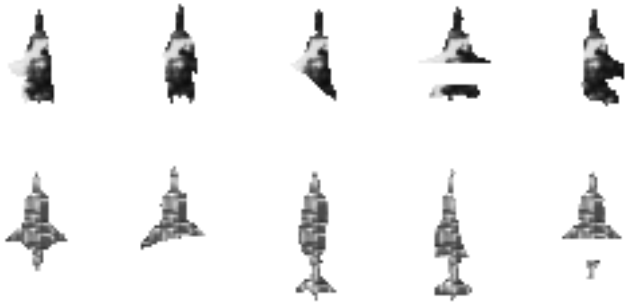


Figure 3.13: Test images for airplane data set. First row: F-14 wings opened, second row: Harrier.

correct mode of the shape density in all test images. When the tail of an Harrier type airplane is occluded, it looks more similar to the F-14 wings opened airplane type. In such cases, Kim et al. [1], Foulonneau et al. [2] and Chen et al. [3] converges to a F-14 wings opened type airplane. Such results can be observed in the first, the second and the fifth test images of the Harrier class. The average Dice score (Hausdorff distance) results on all test images with respect to ground truths are 0.9153 (1.7899) for the proposed method, 0.8746 (6.1726) for the method of Kim et al. [1], 0.8762 (5.9271) for the method of Foulonneau et al. [2] and 0.8748 (6.5479) for the method of Chen et al. [3]. The quantitative results indicate the positive effect of using additional class-dependent features along with the shape prior.

3.5.4 The dendritic spine data set

In this section, we present experimental results on a dendritic spine data set. The data set is obtained from Neuronal Structure and Function laboratory of Champalimaud Neuroscience Foundation, Lisbon.

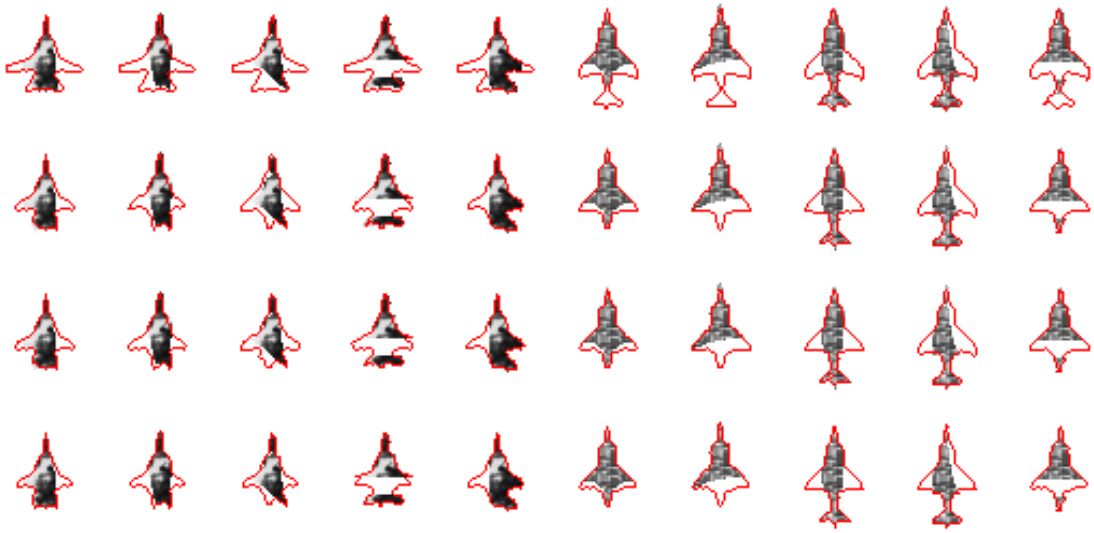


Figure 3.14: Visual segmentation results on the airplane data set. First row: proposed method, second row: Kim et al. [1], third row: Foulonneau et al. [2], fourth row: Chen et al. [3].

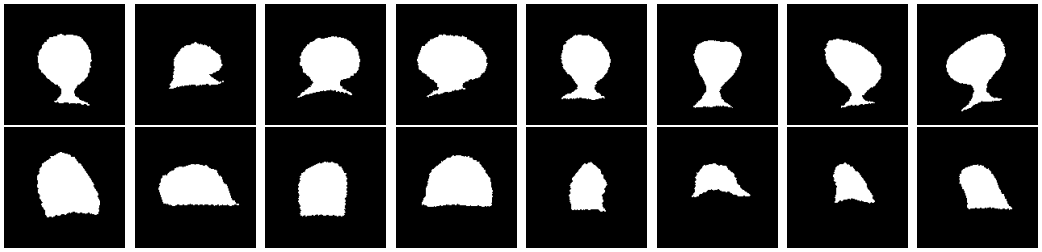
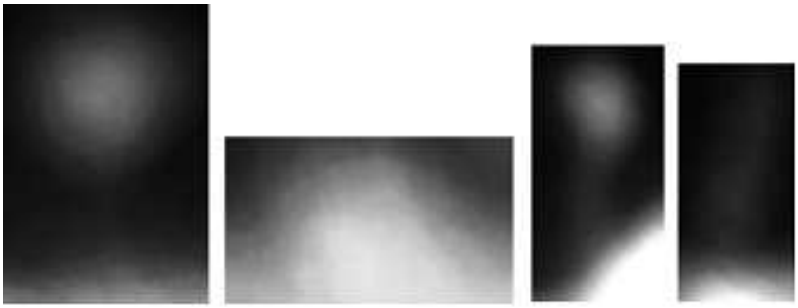


Figure 3.15: Training set for dendritic spine data set. The first 8 spines from the left are mushroom and the remainings are stubby.

In the literature, dendritic spines are generally grouped into four classes: mushroom, thin, stubby, and filopodia (see Figure 3.16). In our experiments, we use training samples from mushroom and stubby classes. The dendritic spine data set contains 88 mushroom and 27 stubby 2D spine intensity images together with the expert's manual segmentations. In our experiments, we use 8 mushroom and 8 stubby dendritic spine shapes shown in Figure 3.15 for training and the remaining 80 mushroom and 19 stubby spines for testing. We perform two different types of experiments with the dendritic spine data set; one is by using appearance-based and the other is by using geometric features. We also compare the segmentation performance of our approach with the approaches of Kim et al. [1], Foulonneau et al. [2] and Chen et al. [3].



(a) Intensity images

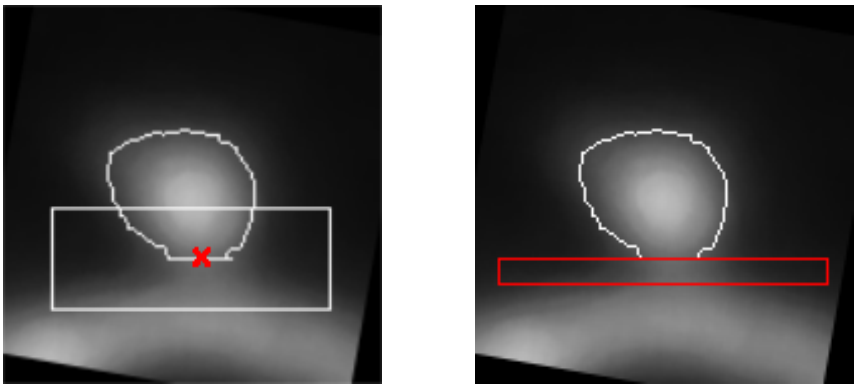
(b) Manual Segmentations

Figure 3.16: Intensity and corresponding manually annotated binary image examples from each spine class. From left to right: Mushroom, Stubby, Thin, and Filopodia.

Spine neck is an important feature that helps to distinguish mushroom and spine classes. Spine head is common for spines in both classes and can be segmented roughly only using the information obtained from the data [64]. Given that spine neck is located in the area below the spine head if it exists, we can extract both appearance and geometric features exploiting the information in this region. We explain how to extract both types of features below:

First, we describe our appearance-based features. Intensity profiles below the spine head provides distinguishable features for spines from different classes [64]. First, we grab a rectangular region such that the bottom point of the spine head (shown by a red cross in Figure 3.17(a)) lies at the center of the rectangle. The second rectangular region shown in Figure 3.17(b) is drawn such that it is located just below the spine head. We fix the size of the first and the second rectangles to 40×110 and 10×130, respectively, in a 150×150 ROI. Using these two rectangular regions, we construct three sets of feature vectors from the training set for classification.

The first set of feature vectors are obtained by summing up the intensities in the first rectangle horizontally. Similarly, the second set of feature vectors are obtained by vertical summation of the intensities in the same rectangle. We present the statistics of these two feature vectors extracted from the training set for each class in Figure 3.18(a) and 3.18(b). In these figures, error bars indicate one standard deviation around the mean. The final set of feature vectors are the histograms of intensities in the second rectangular region. We present average of these histograms for each spine class in Figure 3.18(c). Visual inspection of these feature vectors indicate that they contain discriminatory information about the spine class. Once we extract these three feature vectors from the corresponding intensity images of each training shape, a feature vector f_i is obtained by concatenating them. $\hat{f}$ is also extracted by exploiting the intensity information in the rectangular regions shown in Figure 3.17 as mentioned above. The final segmentation is obtained by evolving the segmenting curve with the data and the shape and feature priors terms.



(a) First region (b) Second region

Figure 3.17: Regions where a potential neck is likely to be located.

Next, we describe the geometric features we use in spine segmentation. Spine neck length is an important geometric feature for identifying different spine classes [74]. In order to compute spine neck length, we follow a procedure consisting of multiple steps. First, we apply Otsu thresholding [75] to get a rough segmentation of the dendritic branch part (the part where the spine is connected to) and apply a fast marching distance transform [76] on this rough segmentation to compute the medial axis of the dendrite. Dendrite segmentation is refined by applying a locally adaptive sized disk-shaped structuring element around the medial axis of the den-

Table 3.7: Average Dice score and Hausdorff distance results on 99 dendritic spines.

	Proposed method w/ apperance priors	Proposed method w/ feature priors	Kim et al. [1]	Foulonneau et al. [2]	Chen et al. [3]
Dice Score	**0.7492**	0.7474	0.6424	0.7348	0.7238
Hausdorff Distance	**19.2002**	20.7133	31.1413	26.0494	25.6581

drite to remove the spines. Once the head of the spine of interest is segmented, a fast marching algorithm [76] computes paths from the center of the spine head to a number of candidate target locations on segmented dendrite through the spine neck. This results in a neck path for each target location. Further, we apply three constraints to select the neck path from these candidate paths. These constraints are: neck path length, path complexity (L_1-norm of path derivatives), and path smoothness (L_1-norm of image intensities along the path). We select the neck path that has collectively the lowest value for these three constraints. Computed neck paths for a mushroom and a stubby spine are presented in Figure 3.19. Note that the computed neck path starts from the center of the spine head. Therefore, for correct computation of the neck length, we have to remove the path part that lies in the spine head. To achieve this, we first compute the radius of the spine head, r, by fitting a circle using the Hough Circle Transform on spine head segmentation and subtract it from the length of the computed path [74]. We compute the neck length for each training shape to form $\mathbf{f}$. When segmenting a test image, we compute the neck length into $\hat{f}$ in the same manner.

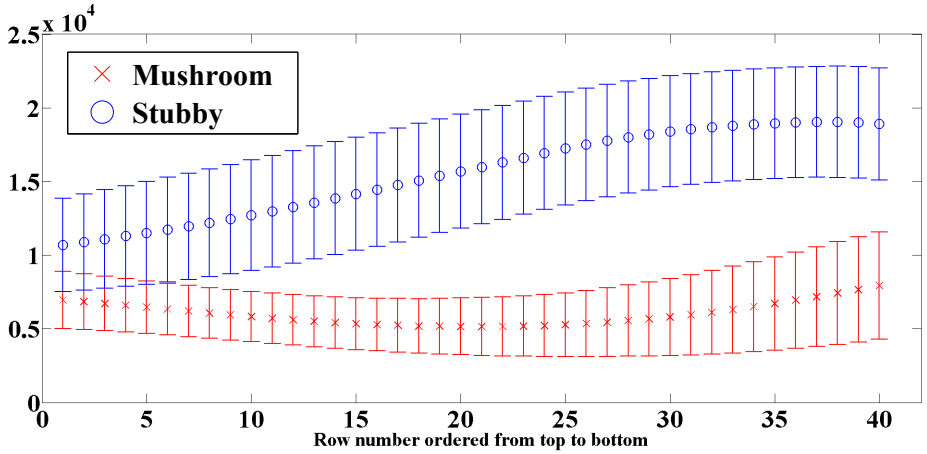
Some visual results that are obtained using the proposed approach (both for appearance-based and geometric features) and the other competing approaches are shown in Figure 3.20. We also evaluate the performance of these segmentation methods quantitatively using Dice score and Hausdorff distance. The average of both Dice score and Hausdorff distance results of all methods are shown in Table 3.7. In all experiments, the best and the second best quantitative results are obtained by the proposed approach with appearance-based feature priors and geometric feature

priors, respectively.

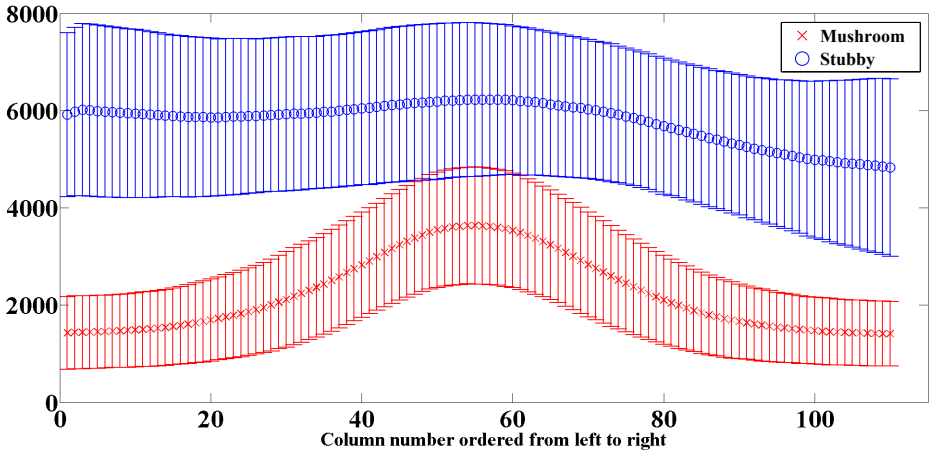
3.6 Conclusion

We have proposed a segmentation method that exploits joint nonparametric shape and feature priors. The proposed method minimizes an energy function that includes a joint nonparametric shape and feature priors term together with the data term using level sets and gradient descent. We provide experimental results on a variety of real and synthetic data sets involving multimodal and complex shape density estimation problems. Experimental results demonstrate that the proposed algorithm achieves better segmentations than the state-of-the-art approaches that use nonparametric shape priors and can be applied to different data sets from various domains.

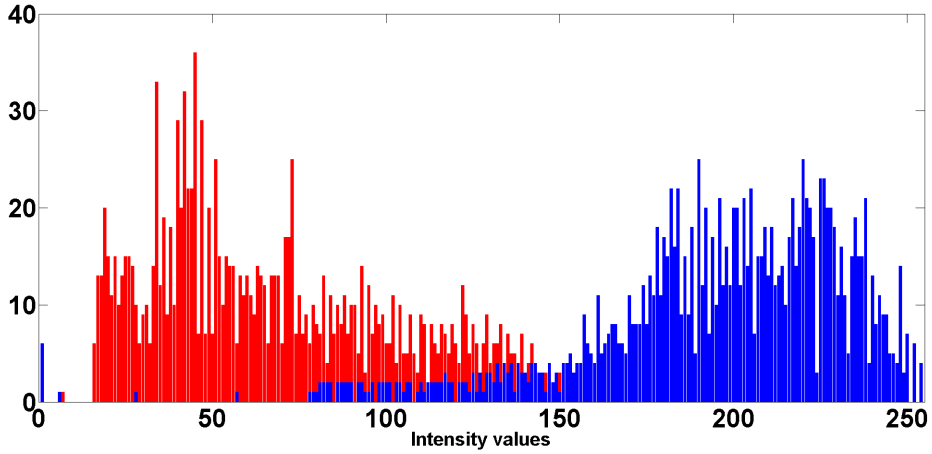
One possible future direction of the proposed method might be developing a similar approach by using a different shape representation than level sets, e.g. Disjunctive Normal Shape Models [19, 23]. Our approach can also be modified slightly and be used as a joint segmentation and classification approach. To this end, classes (perhaps corresponding to modes in the shape density) may be inferred during the segmentation phase and this probabilistic inference may then be used to update the weights of the training samples to drive the segmentation.



(a) Statistics (mean ± one standard deviation) of the first feature vector based on training data.

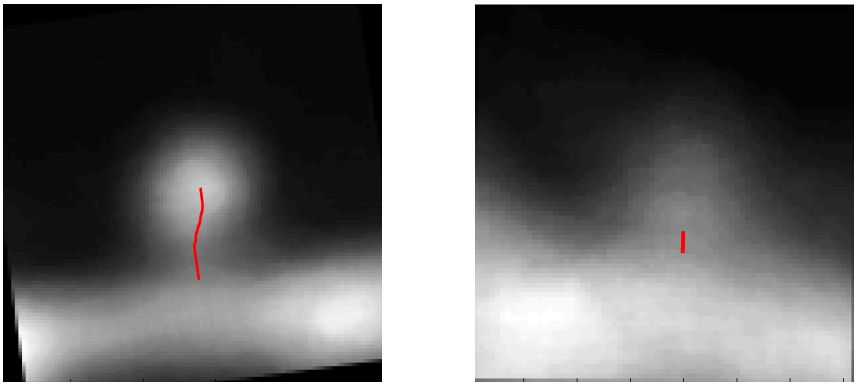


(b) Statistics (mean ± one standard deviation) of the second feature vector based on training data.



(c) Mean of the third feature vector based on training data.

Figure 3.18: Visualization of different sets of appearance-based feature vectors. Red indicates mushroom and blue indicates stubby spines.



(a) Mushroom (b) Stubby

Figure 3.19: Computed neck paths for a mushroom and a stubby spine are shown in red.

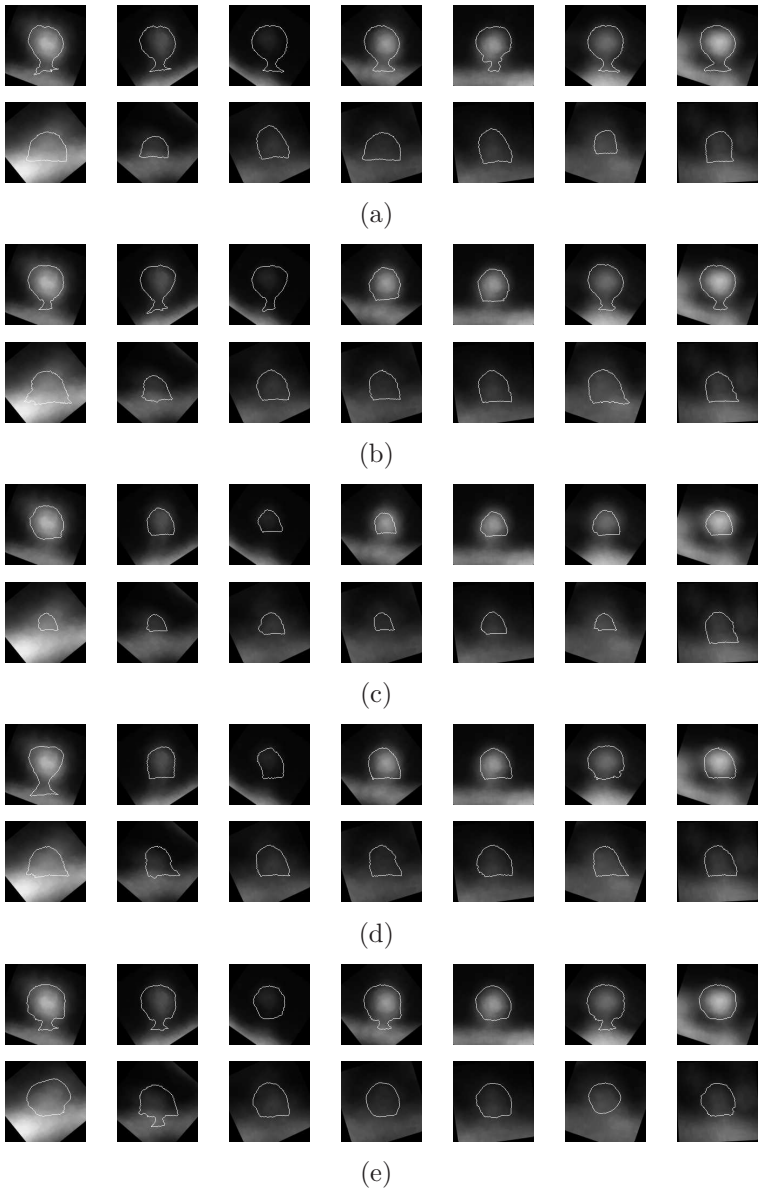


Figure 3.20: Visual segmentation results on the dendritic spine data set. (a) proposed method with appearance-based feature priors, (b) proposed method with geometric feature priors, (c) Kim et al. [1], (d) Foulonneau et al. [2], (e) Chen et al. [3]. Note that in each subfigure, the spines in the first row are mushroom, the ones in the second row are stubby spines.

Chapter 4

Markov chain Monte Carlo Sampling-based Methods for Image Segmentation with Nonparametric Shape Priors

Segmenting images of low quality or with missing data is a challenging problem. Integrating statistical prior information about the shapes to be segmented can improve the segmentation results significantly. Most shape-based segmentation algorithms optimize an energy functional and find a point estimate for the object to be segmented. This does not provide a measure of the degree of confidence in that result, neither does it provide a picture of other probable solutions based on the data and the priors. With a statistical view, addressing these issues would involve the problem of characterizing the posterior densities of the shapes of the objects to be segmented. For such characterization, we propose a Markov chain Monte Carlo (MCMC) sampling-based image segmentation algorithms that use nonparametric shape priors. In addition to better characterization of the statistical structure of the problem, such an approach would also have the potential to address issues with getting stuck at local optima, suffered by existing shape-based segmentation methods. The proposed approaches are able to characterize the posterior probability density in the space of shapes through their samples, and to return multiple solutions, potentially from different modes of a multimodal probability density, which would be encountered, e.g., in segmenting objects from multiple shape classes. We present promising results on a variety of data sets.

4.1 Related work

Incorporating prior shape density into the segmentation process has been widely studied in the literature. A non-exhaustive survey of optimization-based active contour models are given in Section 3.1. All these methods minimize an energy function containing both data fidelity and shape terms, and find a solution at a local optimum.

In order to have a more detailed information about the characteristic of the posterior density, Markov chain Monte Carlo (MCMC) based methods have been proposed. There are a limited number of MCMC-based image segmentation methods in the literature. Moreover, most of these methods generate samples from the posterior density by assuming the prior density is uniform [77], [78], [79]. In other words, they do not use any prior knowledge about shapes. Therefore, such methods are not capable of segmenting objects when the intensities provide very limited information about object boundaries (due to occlusion, noise, missing data etc.). Among these approaches, Fan et al. [77] have developed a method using explicit (marker-based) representations of shape. The proposal distribution generates a candidate sample by randomly perturbing a set of marker points selected on the closed curve. The random perturbation is obtained by generating noise from unit Gaussian distribution and smoothing the noise using a low pass filter. Due to the use of marker points in perturbation, this approach is only applicable to segmentation of simply connected shapes; it cannot represent shapes that include holes and disconnected components. Later, Chang et al. [78] have proposed an efficient MCMC sampling approach on a level set-based curve representation that can handle topological changes. Random curve perturbation is performed through an addition operator on the level set representation of the curve. Additive perturbation is generated by sampling from a Gaussian distribution. Also, some bias is introduced to the additive perturbation with the gradient of the negative logarithm of the posterior density (whose prior density is uniform) to achieve faster convergence. Both Fan et al. [77] and Chang et al. [78] do not satisfy the necessary conditions to implement MCMC since they compute the probability of generating a perturbation approximately. These methods do not explicitly define the proposal distribution; instead they define how to sample from this distribution. The ways they follow to generate a candidate sample

is mostly based on generating a noise vector smoothed with a low-pass filter. The filtered noise vector is then added to the curve to obtain the candidate. Since there are infinite noise and low pass filter combinations that result the same perturbation, the exact computation of this probability is not possible. Chang et al. [78] further extends their methods to achieve order of magnitude speed up in convergence by developing a sampler whose samples at every iteration are accepted [79]. This is achieved by designing a Gibbs-like proposal distribution.

The only sampling-based segmentation approach that uses shape prior in the literature is proposed by Chen et al. [80]. The approach uses the shape prior term suggested by Kim et al. [1] and Cremers et al. [17] to handle multimodal shape densities. Samples are generated by constructing a smooth normal perturbation at a single point on the curve which preserves the signed distance property of the level set. The method is restricted to segmentation of simply connected shapes due to its inability to handle topological changes. Therefore, the approach is not applicable to shapes with complex boundaries.

Although not directly related to the proposed approaches, De Bruijne et al. [81] use a sequential Monte Carlo approach, particle filtering, for segmentation. The method exploits both shape and local appearance priors for segmentation and use particle filtering for optimization purpose. Therefore, the method only returns a single segmentation result. The method captures the shape variation using principal component analysis (PCA) using the assumption that the underlying shape distribution is unimodal. Therefore, it cannot handle cases when the prior shape density is multimodal.

4.2 Motivation

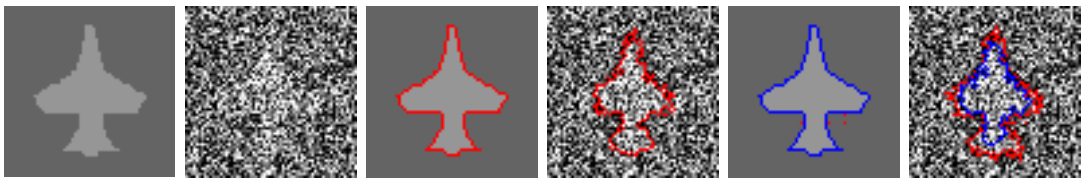
The optimization-based methods introduced in Section 3.1 minimize an energy function containing both data fidelity and shape terms, and find a solution at a local optimum. Having such a point estimate does not provide any measure of the degree of confidence/uncertainty in that result. Moreover, the point estimate does not provide information about the characteristics of the posterior density and may suffer from returning a solution at a local optimum. There might be multiple

reasonable solutions to a segmentation problem especially from different modes of the posterior density if the underlying density is multimodal.

As we mentioned in the previous section, MCMC-based segmentation methods are proposed to deal with these shortcomings of the optimization-based methods. However, most of the MCMC-based segmentation methods assume that the underlying prior shape density is unimodal, i.e., they are not capable of segmenting objects in the case of occlusion, missing data and severe noise. The only method that exploits nonparametric density estimation to learn prior shape density can only segment closed objects since it cannot handle topological changes.

These problems motivate us to develop Markov-Chain Monte Carlo (MCMC) based segmentation approaches that generates samples from posterior density by exploiting prior shape densities. Ideally, such an approach should generate samples from posterior density to 1) provide more information about the degree of confidence/uncertainty in the segmentation result, 2) overcome the issue of being stuck at local optimum, 3) present multiple meaningful segmentations; potentially from different modes of the posterior density.

Figure 4.1 and 4.2 contains illustrative examples that address some of the shortcomings of the optimization-based segmentation approaches and solutions of our MCMC sampling approaches. Let us assume that we are given a training set of binary aircraft images (a "unimodal" prior shape density). Let us also assume that we have two test images with zero and a high amount of noise as shown in Figure 4.1(a). The results of the optimization-based segmentation approach of Kim et al. [1] are shown in Figure 4.1(b). We choose the approach of Kim et al. [1] as a representative one since it is optimization-based and can handle both "unimodal" and "multimodal" shape densities using nonparametric shape priors. As shown in Figure 4.1(b), Kim et al. [1] produces quite well segmentation results on both test images. However, it does not provide any measure of the degree of confidence/uncertainty that arises due to the noise. On the other hand, MCMC sampling-based approaches return a confidence boundary obtained by multiple samples from posterior density (see Figure 4.1(c)). Note that the variations between the low and high confidence boundaries increases as the noise increase; which is expected.

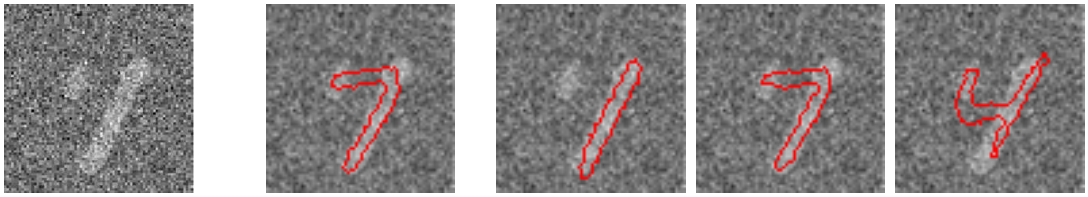


(a) Test Images.

(b) Kim et al. [1].

(c) Low (blue) and high (red) confidence boundaries produced by the MCMC-based sampling framework proposed in this chapter.

Figure 4.1: The first motivating example of using MCMC shape sampling for image segmentation



(a) Test Image.

(b) Kim et al. [1].

(c) Samples from different modes of the posterior density drawn by the proposed approach.

Figure 4.2: The second motivating example of using MCMC shape sampling for image segmentation

As the second motivating example, let us assume that we have a training set of binary handwritten digits for each digit class. Let us segment the test image in Figure 4.2(a) using the approach of Kim et al. [1]. Kim et al. [1] successfully drive an initial curve to a mode of the posterior density and produce the single segmentation shown in Figure 4.2(b). However, it does not give any idea about other probable segmentations. The MCMC sampling-based approaches return multiple meaningful solutions by drawing samples from different modes of the posterior density as show in Figure 4.2(c).

4.3 MCMC shape sampling for image segmentation with nonparametric shape priors

In this section, we introduce our first MCMC shape sampling approach for image segmentation exploiting nonparametric shape priors. The proposed approach is published in [58].

4.3.1 Contributions

Our contributions in this work are twofold. First, as the major contribution, we present a Markov chain Monte Carlo (MCMC) sampling approach that uses nonparametric shape priors for image segmentation. Our MCMC sampling approach is able to characterize the posterior shape density by returning multiple probable solutions and avoids the problem of getting stuck at a single local optimum. To the best of our knowledge, this is the first approach that performs MCMC shape sampling-based image segmentation through an energy functional that uses nonparametric shape priors and level sets. We present experimental results on several data sets containing low quality images and occluded objects involving both unimodal and multimodal shape densities. As a second contribution, we provide an extension within our MCMC framework, that involves a local shape prior approach for scenarios in which objects consist of parts that can exhibit independent shape variations. This extension allows learning shapes of object parts independently and then merging them together. This leads to more effective use of potentially limited training data. We demonstrate the effectiveness of this approach on a challenging segmentation problem as well.

4.3.2 Metropolis-Hastings sampling in the space of shapes

With a Bayesian perspective, segmentation can be viewed as the problem of estimating the boundary c based on image data, y:

$$p(c|y) \propto \exp(-E(c)) \tag{4.1}$$

where,

$$E(c) = E_y(c) + E_{shape}(c) = -\log p(y|c) - \log p_c(c) \tag{4.2}$$

In this work, we present an algorithm to draw samples from $p(c|y)$ which is, in general, a complex distribution and is not possible to sample from directly.

In this work, we use level sets to represent c. Level set representation is essentially a mapping

$$\phi : \{0,1\}^{M\times N} \rightarrow \mathbb{R}^{MN}$$

from the binary space to the real space. In the literature, it has been found more convenient to work with level sets to represent c to handle topological changes and its effectiveness when computing gradients. In the rest of this chapter, we work with $x = \phi(c)$. Therefore, the problem turns into generating samples from $p(x|y)$.

MCMC methods were developed to draw samples from a probability distribution when direct sampling is non-trivial. We use Metropolis-Hastings sampling [40] which has been previously used for image segmentation [77,78,80]. In Metropolis-Hastings sampling, instead of directly sampling from p, a proposal distribution q is defined and samples from q are accepted in such a way that samples from p are generated asymptotically. The Metropolis-Hastings acceptance probability is defined as

$$Pr\Big[x^{(t+1)} = x'|x^{(t)}\Big] = \min\Bigg[\underbrace{\frac{\pi(x')}{\pi(x^{(t)})}.\frac{q(x^{(t)}|x')}{q(x'|x^{(t)})}}_{\text{Metropolis-Hastings ratio}}, 1\Bigg]. \tag{4.3}$$

The Metropolis-Hastings threshold, η, is randomly generated from the uniform distribution in $[0,1]$. The candidate (proposed) sample $\S^{(t+1)}$ is accepted if $Pr\Big[x^{(t+1)} = x'|x^{(t)}\Big]$ is greater than η. Otherwise, $x^{(t+1)} = x^{(t)}$. In Equation (4.3), $x^{(t)}$ and x' represent the current sample and proposed sample, respectively. The superscripts (t) and $(t+1)$ denote the sampling iteration count, and $\pi(x) \propto \exp(-E(x))$. After a sufficient number of iterations (i.e., the mixing time) a single sample from the posterior is produced by converging to the stationary distribution. Evaluating the acceptance probability is a key point in MCMC methods. Correct evaluation of the acceptance probability satisfies the sufficient conditions for convergence to the desired posterior distribution: detailed balance and ergodicity. Therefore, the problem turns into the correct computation of forward $q(x^{(t+1)}|x')$ and reverse $q(x'|x^{(t+1)})$ transition probabilities of the proposal distribution.

4.3.3 The proposed method

We assume that the curve at the 0^{th} sampling iteration, $x^{(0)}$, is the curve that is found by minimizing only the data fidelity term, $E_y(x)$. We use piecewise-constant version of the Mumford-Shah functional [55, 66] for data driven segmentation. One can consider optimizing more sophisticated energy functions such as mutual information [67], J-Divergence [68], and Bhattacharya Distance [69] to obtain $x^{(0)}$. Also, using an MCMC sampling based approach for data driven segmentation can enrich the sampling space since it would allow subsequent MCMC shape sampling to use several initial curves to start from. After the curve finds all the portions of the object boundary identifiable based on the image data only (e.g., for a high SNR image with an occluded object, one would expect this stage to capture the non-occluded portions of the object reasonably well), we activate the process of generating samples from the underlying space of shapes using nonparametric shape priors.

The overall proposed MCMC shape sampling algorithm is given in Algorithm 4. The steps of the algorithm are explained in the following three subsections.

Algorithm 4 MCMC Shape Sampling

1: **for** $i = 1 \rightarrow M$ **do** ▷ M : # of samples to be generated
2: Randomly select class of $x^{(0)}$ as introduced in Section 4.3.3.
3: **for** $t = 0 \rightarrow (N-1)$ **do** ▷ N : # of sampling iterations
4: Generate candidate sample $\tilde{x'}$ from curve $\tilde{x}^{(t)}$ as introduced in Section 4.3.3.
▷ The steps between 5 - 10 are introduced in Section 4.3.3
5: Calculate Metropolis-Hastings ratio, Pr
6: $\eta = \mathcal{U}_{[0,1]}$
7: **if** $(t+1) = 1$ OR $\eta < Pr$ **then**
8: $\tilde{x}^{(t+1)} = \tilde{x'}$ ▷ Accept the candidate
9: **else**
10: $\tilde{x}^{(t+1)} = \tilde{x}^{(t)}$ ▷ Reject the candidate
11: **end if**
12: **end for**
13: **end for**

Random class decision

Suppose that we have a training set $\mathbf{x} = \{x_1, \ldots, x_n\}$ consisting of shapes from n different classes where each class $x_i = \{x_{ij} | j \in [1, m_i] \in \mathbb{Z}\}$ contains m_i different example shapes. We align training shapes x_{ij} into $\tilde{x}_{ij}$ using the alignment approach presented in Tsai et al. [15] in order to remove the artifacts due to pose differences such as translation, rotation, and scaling.

We exploit the shape prior term $p_x(x)$ proposed by Kim et al. [1] to select the class of the curve $\tilde{x}^{(0)}$. The prior probability density function of the curve evaluated at sampling iteration zero is

$$p_x(\tilde{x}^{(0)}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \sum_{j=1}^{m_i} k(d_{L_2}(\tilde{x}^{(0)}, \tilde{x}_{ij}), \sigma) \tag{4.4}$$

where $k(., \sigma)$ is a $1D$ Gaussian kernel with kernel size σ, $d_{L_2}(.,.)$ is the L_2 distance metric and ϕ denotes the level set representation of a curve. Also, note that $\tilde{x}^{(0)}$ is the aligned version of $x^{(0)}$ with the training set. By exploiting Equation (4.4), we can compute the prior probability density of the shapes in x_i evaluated at $\tilde{x}^{(0)}$, $p'_{x_i}(\tilde{x}^{(0)})$, as follows

$$p'_{x_i}(\tilde{x}^{(0)}) \propto \frac{1}{m_i} \sum_{j=1}^{m_i} k(d_{L_2}(\tilde{x}^{(0)}, \tilde{x}_{ij}), \sigma). \tag{4.5}$$

We randomly select a class for shape $\tilde{x}^{(0)}$ where the probability of selecting a class is proportional to the value of $p'_{x_i}(\tilde{x}^{(0)})$ computed in Equation (4.5). When we generate multiple samples, the random class selection step helps us obtain more samples from the classes having higher probabilities.

Generating a candidate sample

In this section, we explain how to generate a candidate sample from the proposal distribution q. Once the class of $\tilde{x}^{(0)}$ is randomly selected, we perform curve perturbation exploiting the training samples in this class. Let $\tilde{x}_r$ be the set that contains the training shapes from the selected class r. We randomly choose γ training shapes from $\tilde{x}_r$ where the probability of selecting each shape is proportional to its similarity with $\tilde{x}^{(t)}$. We compute the similarity between a training shape $\tilde{x}_{rj}$ and $\tilde{x}^{(t)}$ as the value of the probability density function, s, at $\tilde{x}_{rj}$ where,

$$s_{\tilde{x}^{(t)}}(\tilde{x}_{rj}) \propto k(d_{L_2}(\tilde{x}^{(t)}, \tilde{x}_{rj}), \sigma). \tag{4.6}$$

Note that a training shape can be selected multiple times and random training shape selection is repeated in each sampling iteration. We represent the set composed of randomly selected γ training shapes at sampling iteration t by $\tilde{\mathbf{x}}_{\mathbf{R}}^{(t)}$.

Finally, we define the forward perturbation for the curve $\tilde{x}^{(t)}$ with level sets as follows:

$$\tilde{x'} = \tilde{x}^{(t)} + \alpha \mathbf{f^{(t)}} \tag{4.7}$$

We choose $\mathbf{f^{(t)}}$ as the negative gradient of the energy function given in Equation (4.2) in order to move towards a more probable configuration in each perturbation. Here, α indicates the step size for gradient descent. Note that we use randomly selected training samples, $\tilde{x}_{Rj} \in \tilde{\mathbf{x}}_{\mathbf{R}}^{(t)}$, for curve perturbation. Mathematically this is expressed as

$$\begin{aligned} \mathbf{f^{(t)}} &= -\frac{\partial E(\tilde{x}^{(t)})}{\tilde{x}^{(t)}} = \frac{\partial \log p(y|\tilde{x}^{(t)})}{\partial t} \\ &+ \frac{1}{p_{\tilde{x}^{(t)}}(\tilde{x}^{(t)})} \frac{1}{\gamma} \frac{1}{\sigma} \sum_{j=1}^{\gamma} k(d_{L_2}(\tilde{x}^{(t)}, \tilde{x}_{Rj}), \sigma)(\tilde{x}_{Rj} - \tilde{x}^{(t)}) \end{aligned} \tag{4.8}$$

In other words, updating the curve $\tilde{x}^{(t)}$ toward the negative gradient direction of the energy functional produces the candidate curve $\tilde{x'}$.

Evaluating the Metropolis-Hastings ratio

Computation of the first fraction in the Metropolis-Hastings ratio in Equation (4.3) is straightforward since $\pi(x) \propto \exp(-E(x))$. Recall that the candidate curve $\tilde{x'}$ is dependent on the forward perturbation $\mathbf{f^{(t)}}$. Therefore, we compute the forward perturbation probability by considering the value of the probability density function, s, for each randomly selected training shape $\tilde{x}_{Rj} \in \tilde{\mathbf{x}}_{\mathbf{R}}^{(t)}$ as follows:

$$q(\tilde{x'}|x^{(t)}) = \prod_{\tilde{x}_{Rj} \in \tilde{\mathbf{x}}_{\mathbf{Rj}}^{(t)}} s(\tilde{x}_{Rj}) \tag{4.9}$$

Similarly, the reverse perturbation probability in sampling iteration $(t+1)$ is computed as the probability of selecting random shapes in $\tilde{\mathbf{x}}_{\mathbf{R}}^{(t-1)}$ which have been used to produce the curve $\tilde{x}^{(t)}$:

$$q(\tilde{x}^{(t)}|\tilde{x'}) = \prod_{\tilde{x}_{Rj} \in \tilde{\mathbf{x}}_{\mathbf{Rj}}^{(t-1)}} s(\tilde{x}_{Rj}) \tag{4.10}$$

Note that, given the above formulations, computation of the reverse perturbation probability is not possible for candidate curve $\tilde{x'}^{(1)}$, the curve at sampling iteration 1, since we have to use information from sampling iteration -1 for evaluation of Equation (4.10), which is not available. Therefore, we accept the candidate sample $\tilde{x'}^{(1)}$ without evaluating the Metropolis-Hastings ratio and consider the above-mentioned steps for generating samples after sampling iteration 1.

4.3.4 Discussion on sufficient conditions for MCMC sampling

Convergence to the correct stationary distribution is crucial in MCMC methods. Convergence is guaranteed with two sufficient conditions: (1) that the chain is ergodic, and (2) that detailed balance is satisfied in each sampling iteration. Ergodicity is satisfied when the Markov chain is aperiodic and irreducible. Aperiodicity of a complicated Markov chain is a property that is hard to prove as attested in the literature [82].

Detailed balance is satisfied as long as the Metropolis-Hastings ratio in Equation (4.3) is calculated correctly. We have already described how we compute the Metropolis-Hastings ratio in the previous section. Empirical results show that a stationary distribution is most likely reached since our samples converge. Related pieces of work in [77], [78], and [80] argue that the Markov chain is unlikely to be periodic because the space of segmentations is so large. Similarly, we can also assert that our Markov chain is unlikely to be periodic. Even if the chain is periodic in exceptional cases, the average sample path converges to the stationary distribution as long as the chain is irreducible. Irreducibility of a Markov chain requires showing that transitioning from any state to any other state has finite probability. Chen et al. [80] and Chang et al. [78] provide valid arguments that the Markov chain is irre-

ducible whereas Fan et al. [77] does not discuss this property. As explained in the previous section, curve perturbation in our framework is performed with randomly selected training samples $\tilde{\mathbf{x}}_{\mathbf{R}}^{(t)}$ and each shape has finite probability to be selected at any sampling iteration. With this perspective, we can also argue that each move between shapes has finite probability in our approach.

4.3.5 Extension to MCMC sampling using local shape priors

In this section, we consider the problem of segmenting objects with parts that can go through independent shape variations. We propose to use local shape priors on object parts to provide robustness to limitations in shape training size [19, 83]. Let us consider the motivating example shown in Figure 4.3. In this example, there are three images of walking silhouettes: two for training and one for testing. Note that the left leg together with the right arm of the test silhouette involves missing regions. When segmenting the test image using nonparametric shape priors [1] based on global training shapes[1], the result may not be satisfactory (see the rightmost image in the first row of Figure 4.3), because the shapes in the training set do not closely resemble the test image. This motivates us to represent shapes with local priors such that resulting segmentation will mix and match information from independent object parts (e.g., by taking information about the the right arm from the first training shape and about the left leg from the second training shape).

Our idea of constructing local shape priors is straightforward. Once the training shapes are aligned, we divide the shapes into patches, such that each patch contains a different local shape region. Each patch is indicated by a different color in the second row of Figure 4.3. Note that the patches representing the same local shape have identical size. For MCMC shape sampling using local shape priors, it is straightforward to adapt the formulation in the previous sections to consider local priors. In particular, instead of choosing random global shapes using the values computed by Equation (4.6), we compute these values for each patch (local shape) and select random patches among all training images. Note that evaluation of forward and

[1]Unless otherwise stated, the shape priors we use are global. We explicitly refer to global shape priors when we need to distinguish them from local shape priors.

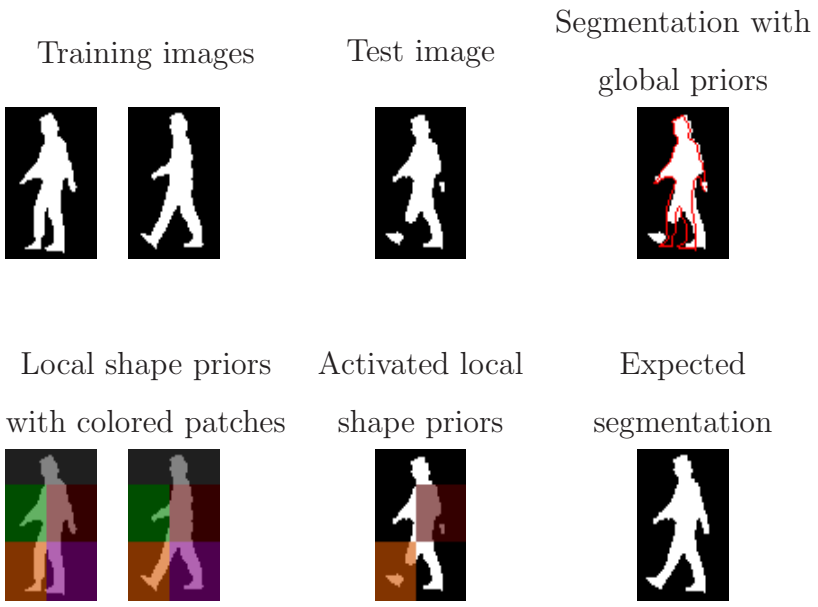


Figure 4.3: Motivating example for using local shape priors in walking silhouettes data set.

reverse perturbation probabilities should also be modified accordingly.

4.3.6 Experimental results

In this section, we present empirical results of our MCMC shape sampling algorithm on segmentation of potentially occluded objects in low-quality images. Note that, when dealing with segmentation of objects with unknown occlusions, $E_y(x)$ increases when the shape term delineates the boundaries in the occluded region. This can lead to overall increasing effect on $E(x')$ for a candidate curve and to the rejection of the candidate sample. In order to increase the acceptance rate of our approach, we use $\pi(x) \propto \exp(-E_{shape}(x))$ instead of $\pi(x) \propto \exp(-E(x))$ in our experiments involving occluded objects (see supplementary material for experiments involving missing data in which we use $\pi(x) \propto \exp(-E(x))$). This does not cause any problem in practice since the data fidelity term (together with the shape prior term) is involved in the curve perturbation step, enforcing consistency with the data.

We perform experiments on several data sets: aircraft [1], MNIST handwritten digits [84], and walking silhouettes [17]. In the following subsections, we present quantitative and visual results together with discussions of the experiments for each

data set.

Experiments on the aircraft data set

The aircraft data set [1] contains 11 synthetically generated binary aircraft images as shown in the top row of Figure 4.4. We construct the test images shown in the middle and the bottom rows of the same figure by cropping the left wings from the binary images to simulate occlusion and by adding different amounts of noise. Note that the test images shown in the middle row of Figure 4.4 (test image set - 1) have higher SNR than the ones shown in the bottom row (test image set - 2). In our experiments, we use this data set in leave-one-out fashion, i.e., we use one image as test and the remaining 10 binary images for training.

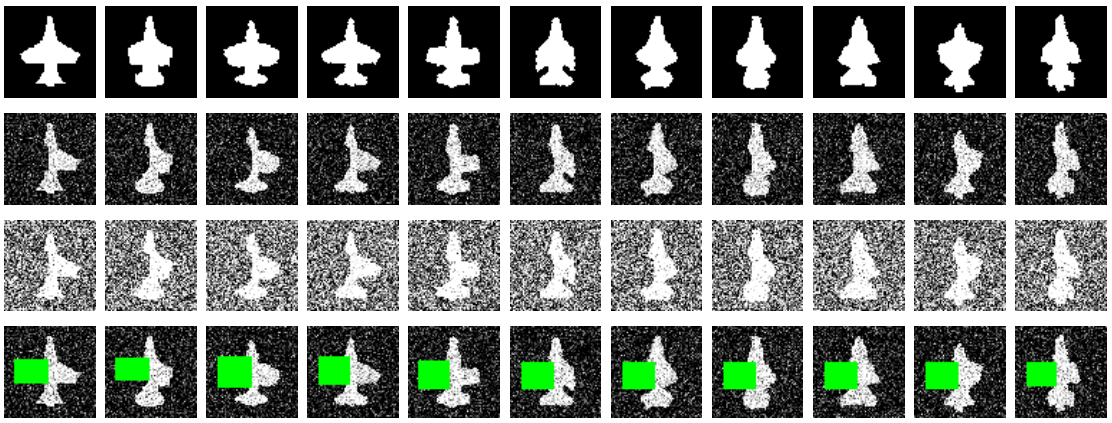


Figure 4.4: The aircraft data set. First row: Training set, second row: test image set - 1 and third row: test image set - 2, fourth row: test image set - 3. Note that green indicates missing pixels in test image set - 3.

In Figure 4.5, we present some visual and quantitative results on the first three images from the test image set - 1 shown in Figure 4.4. In this experiment, we generate 500 samples using our shape sampling approach for each test image. We also obtain segmentations using the optimization-based segmentation approach of Kim et al. [1] (see the second column of Figure 4.5). We compare each sample and the result of Kim et al. [1] with the corresponding ground truth image using precision - recall values and the F-measure. The samples with the best F-measure value are shown in the third column of Figure 4.5. Finally, we plot the precision - recall values (PR plots) for each sample and for the result of Kim et al. [1] in the fourth column

of Figure 4.5. Here, the data fidelity term keeps the curve at the object boundaries and shape prior term helps to complete the shape in the occluded part. In our approach, since we select the most probable subset of training images and evolve the curve with the weighted average of these images, the results of our approach are more likely to produce better fits for the occluded part. In the experiments shown in Figure 4.5, our approach can generate better samples than the result of Kim et al. [1] in all test images. Moreover, our algorithm is able to generate many different samples in the solution space. By looking at these samples, one can also have more information about the confidence in a particular solution.

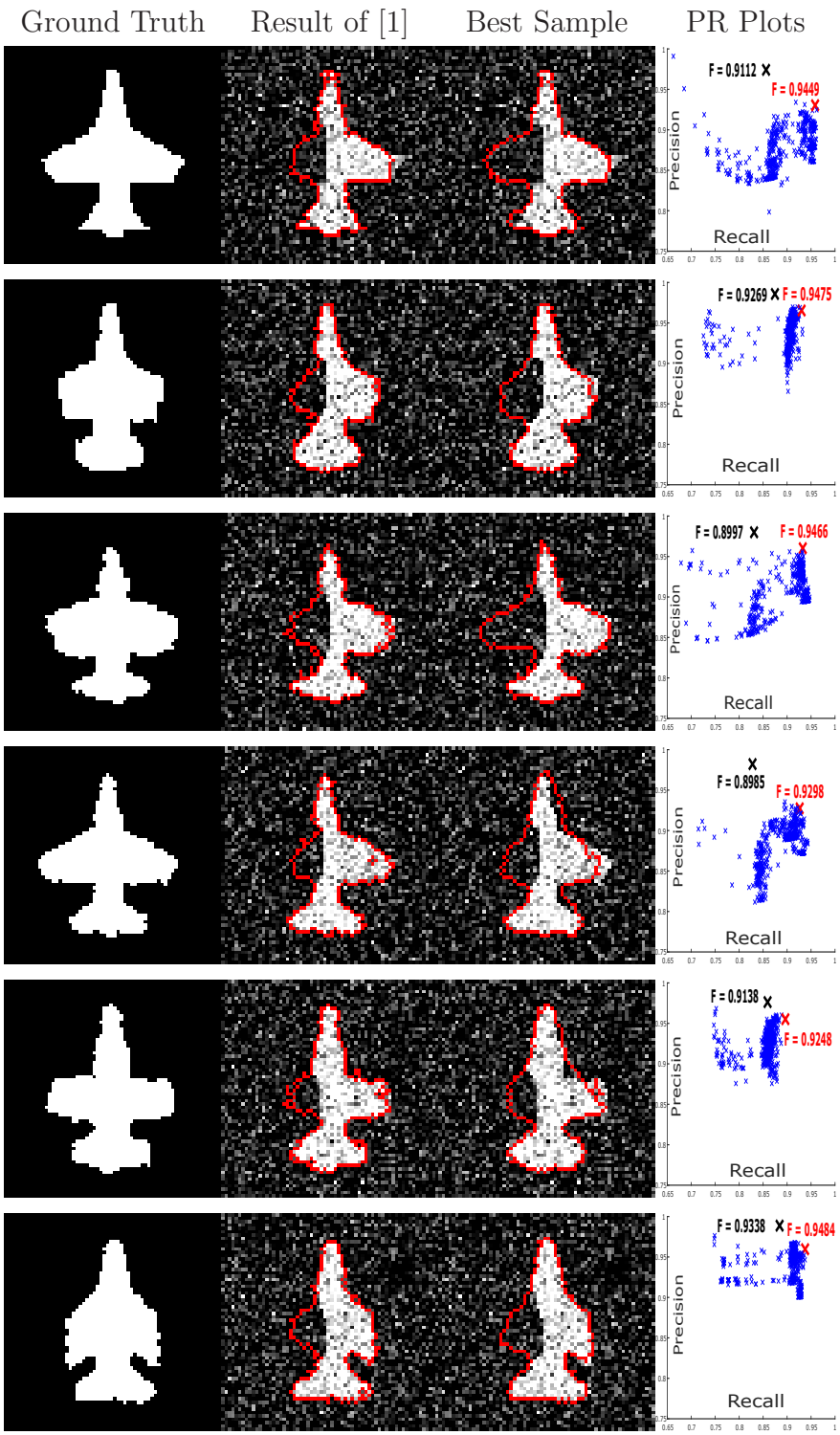


Figure 4.5: Experiments on test image set - 1 of the aircraft data set. Note that each row contains the results for a different test image. In the PR plots, '×'and '×'mark the samples produced by our approach where '×'indicates the sample with the best F-measure value, and '×'marks that of segmentation of Kim et al. [1].

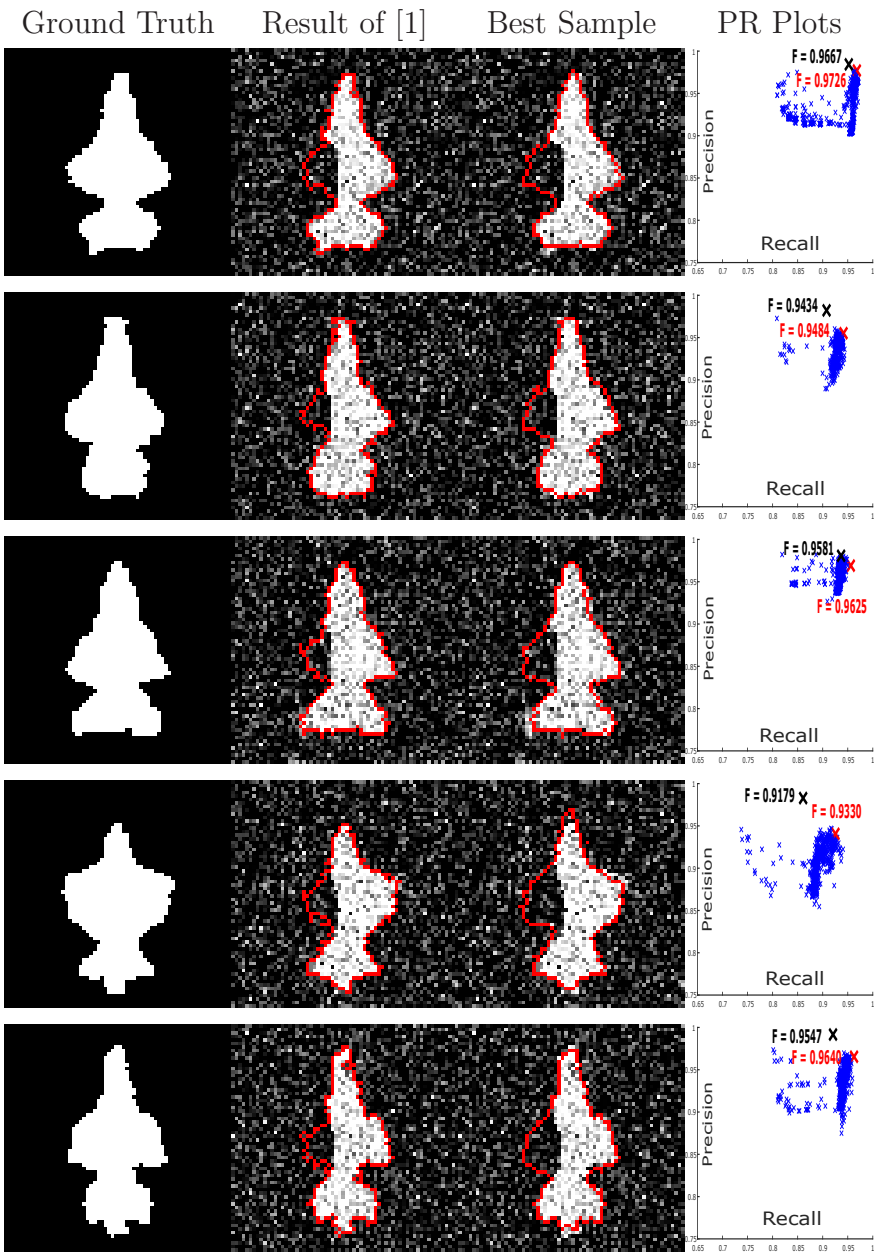


Figure 4.5 (cont.): Experiments on test image set - 1 of the aircraft data set. Note that each row contains the results for a different test image. In the PR plots, '×'and '×'mark the samples produced by our approach where '×'indicates the sample with the best F-measure value, and '×'marks that of segmentation of Kim et al. [1].

We also perform experiments on the aircraft test image set - 2 shown in Figure 4.4 and present results on the first three images in Figure 4.6. The segmentation problem in this image set is more challenging than the previous case because of lower SNR.

We perform experiments with the same settings as in test image set - 1 and present the results in the same way in Figure 4.6. In this case, we have to give more weight to the shape prior term during evolution to complete the occluded part because of the high amount of noise. Because of the limited role of the data fidelity term, the curve starts losing some part of the boundary after the shape term is turned on since the role of the data term is limited. Therefore, in this case, not only the occluded part but also the other parts of the aircraft shape approach a weighted average of the objects in the training set during curve evolution. Note from Figure 4.6 that the results of Kim et al. [1] on different test images are very similar to one another. However, our sampling approach produces more diverse samples including better ones than the result of Kim et al. [1] in terms of F-measure in most cases.

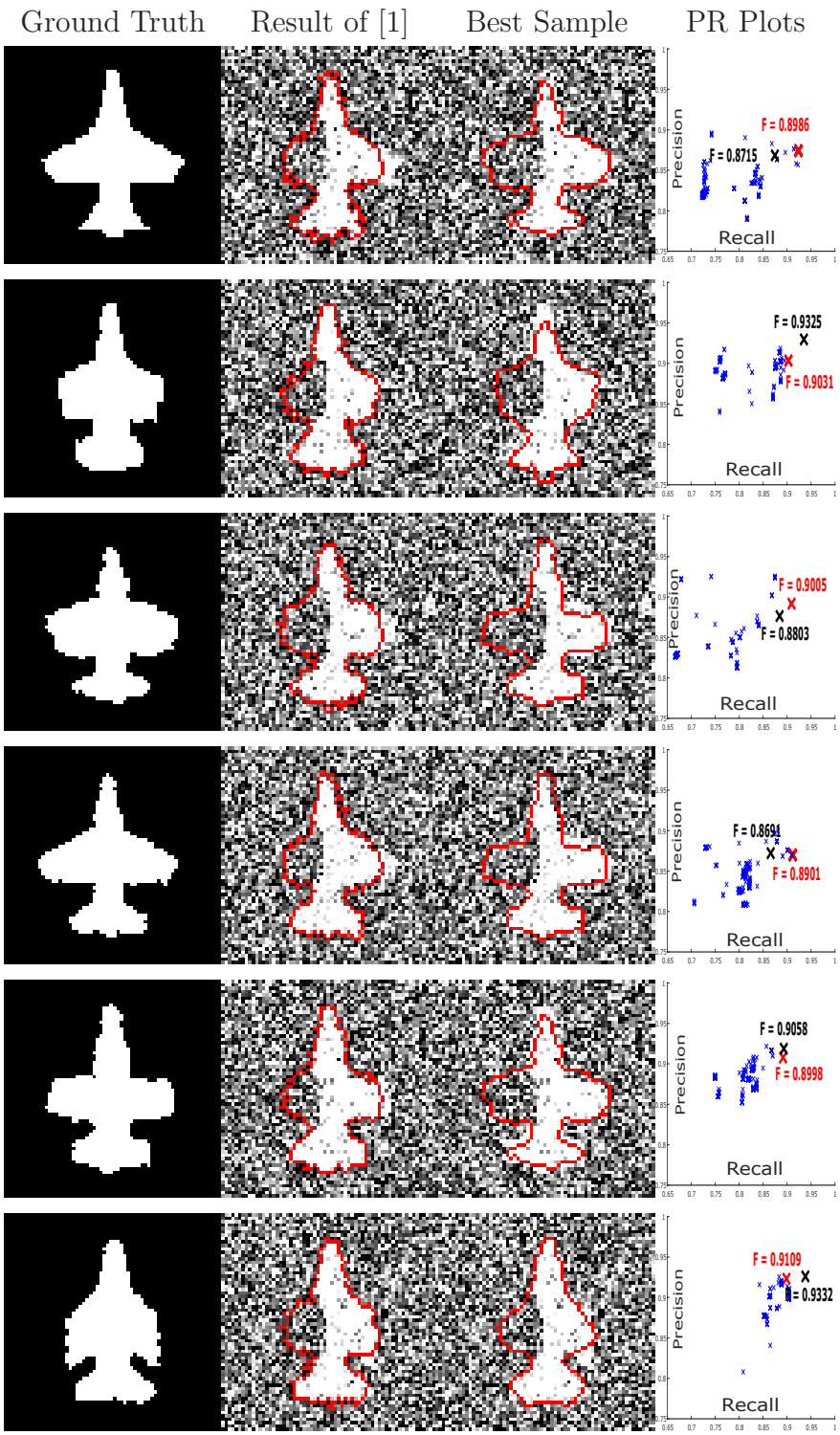


Figure 4.6: Experiments on test image set - 2 of the aircraft data set. Note that each row contains the results for a different test image. In the PR plots, '×'and '×'mark the samples produced by our approach where '×'indicates the sample with the best F-measure value, and '×'marks that of segmentation of Kim et al. [1].

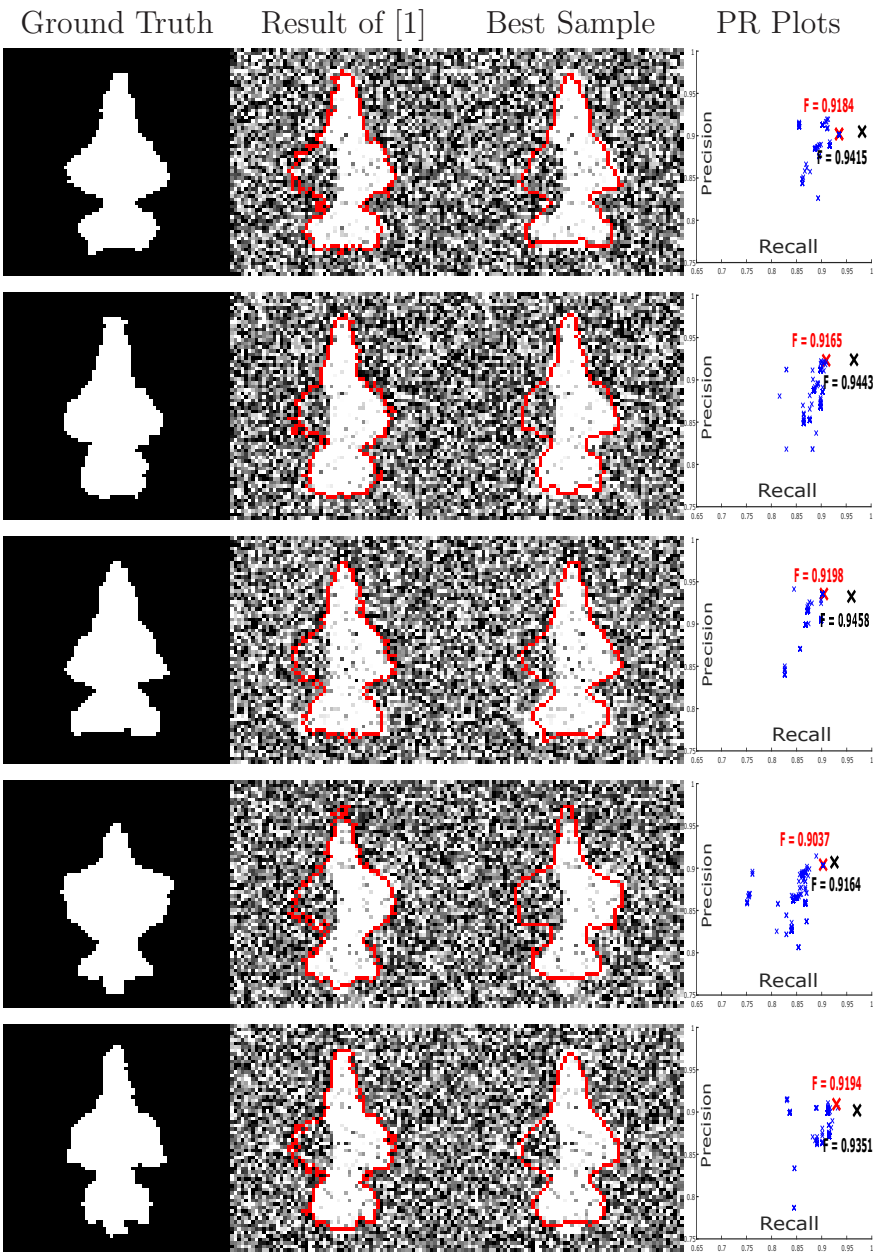


Figure 4.6 (cont.): Experiments on test image set - 2 of the aircraft data set. Note that each row contains the results for a different test image. In the PR plots, '×'and '×'mark the samples produced by our approach where '×'indicates the sample with the best F-measure value, and '×'marks that of segmentation of Kim et al. [1].

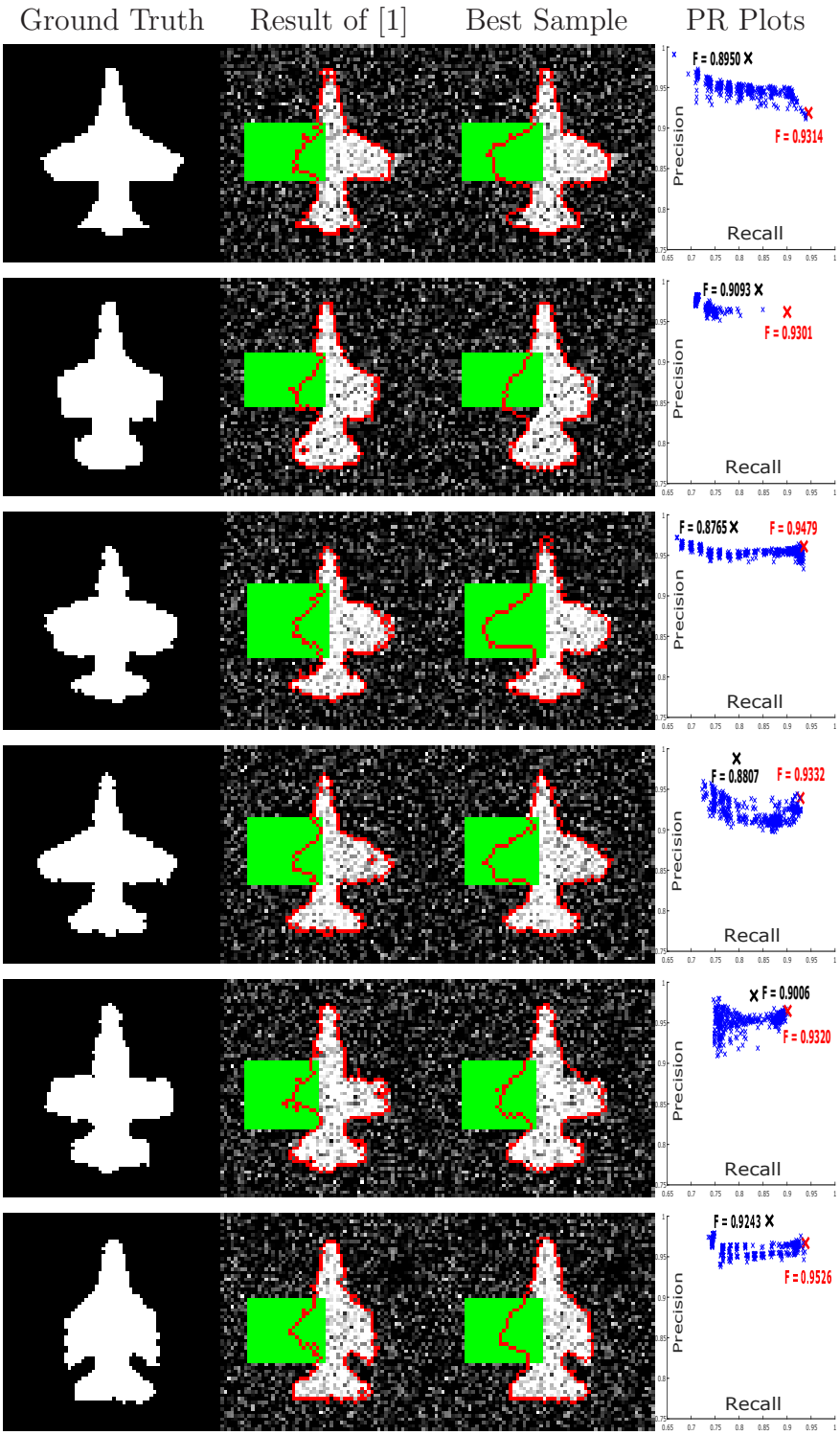


Figure 4.7: Experiments on test image set - 3 of the aircraft data set. Note that each row contains the results for a different test image. In the PR plots, '×'and '×'mark the samples produced by our approach where '×'indicates the sample with the best F-measure value, and '×'marks that of segmentation of Kim et al. [1].

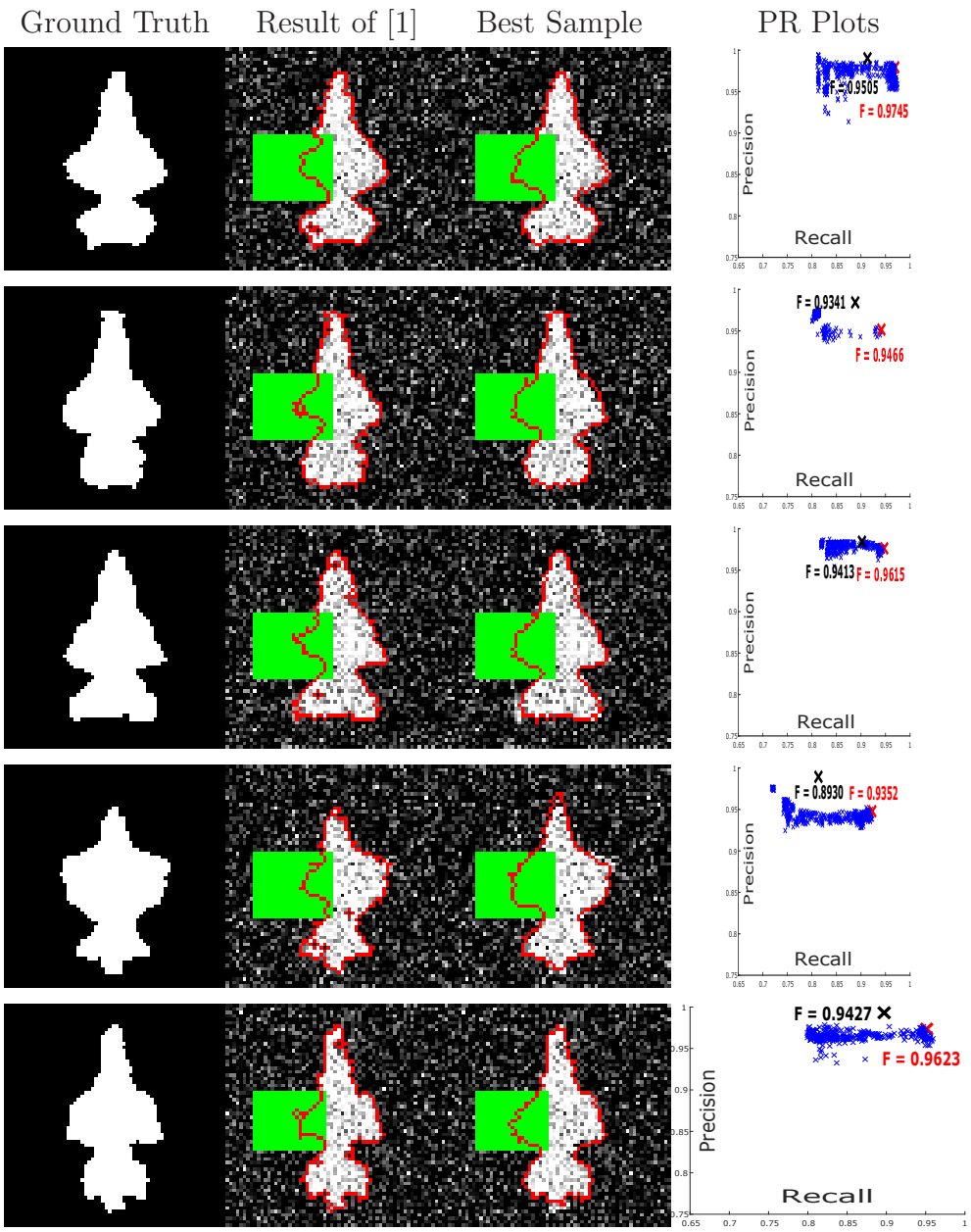


Figure 4.7 (cont.): Experiments on test image set - 3 of the aircraft data set. Note that each row contains the results for a different test image. In the PR plots, '×'and '×'mark the samples produced by our approach where '×'indicates the sample with the best F-measure value, and '×'marks that of segmentation of Kim et al. [1].

Experiments on the MNIST data set

In this section, we present empirical results on the MNIST handwritten digits [84] data set which includes a multimodal shape density (i.e, training set contains shapes

from multiple classes corresponding to different modes of the shape density). The MNIST handwritten digits data set contains 60,000 training examples and 10,000 test images from 10 different digit classes. In our experiments, we take a subset of 100 images for training such that each class contains 10 training examples. Test images, none of which are contained in the training set, are obtained by cropping some parts of the digits and adding noise. The test images that we use in our experiments are shown in Figure 4.8.

Figure 4.8: Test images from the MNIST data set. From left to right: MNIST - 1, MNIST - 2, and MNIST - 3.

In our experiments on the MNIST data set, we generate 1000 samples using our shape sampling approach. In order to interpret our results, we use three methodologies: (1) Compute the average energy for each class by considering the samples generated in that class. Choose the best three classes with respect to average energy values. Display the best three samples from each class in terms of energy. These samples are most likely good representatives of the modes of the target distribution, (2) Compute the histogram images $H(x)$ which indicate in what percentage of the samples a particular pixel is inside the boundary. This can be simply computed by summing up all the binary samples and dividing by the number of samples [77]. $H(x)$ can be computed for each class for problems involving multimodal shape densities. We draw the marginal confidence bounds, the bounds where $H(x) = 0.1$ and $H(x) = 0.9$, over the test image for each class, (3) Count the number of samples obtained from each class. This can allow a probabilistic interpretation of the results.

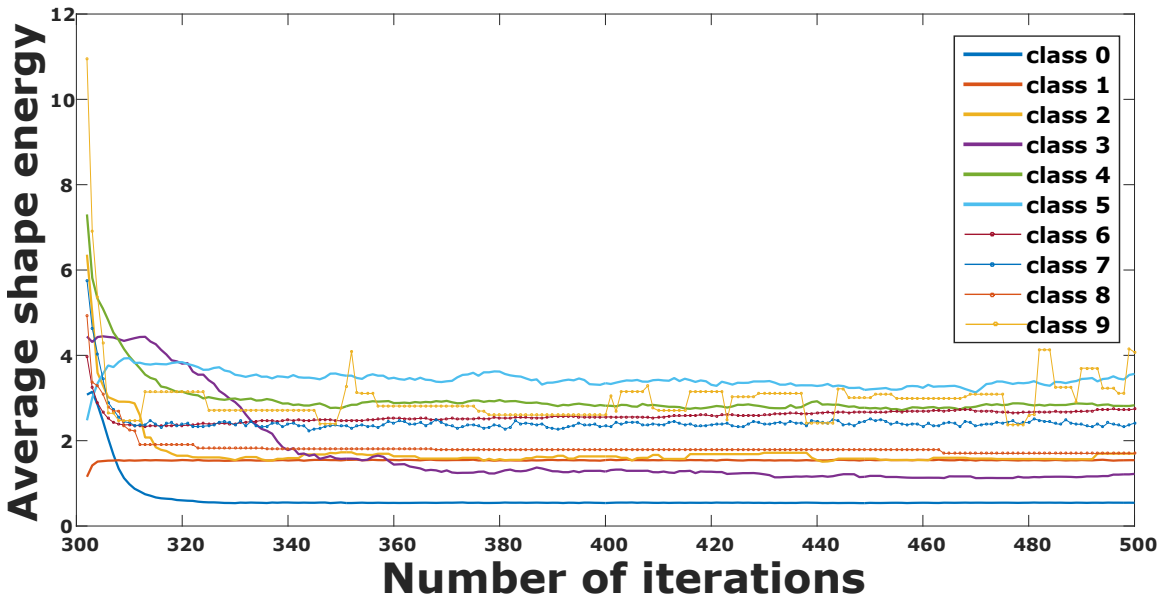


Figure 4.9: Average shape energy ($E_{shape}(x)$) across all sampling iterations for all digit classes for test image MNIST - 1. Note that the number of iterations start from 300 in x-axis because the previous iterations involve segmentation with the data term only.

Figure 4.9 demonstrates the average shape energy for each class, $E_{shape}(x)$, as a function of sampling iterations for test image MNIST - 1. We note that while the average energy appears to be smoothly converging, the energy for each sample path can sharply increase and decrease. The plot of class 9 in Figure 4.9 exhibits such a such pattern because there is only one sample generated from this class. As the number of samples generated in each class increases, the average sample path converges to a stationary distribution.

Number of samples generated from each digit class for all the three test images is shown in Table 4.1. This allows us to make a probabilistic interpretation of the segmentation results. One can evaluate the confidence level of the results by analyzing the number of samples generated from a class over all samples.

In different segmentation applications, one can investigate solutions obtained from different parts of the posterior probability density. Especially, in the case of multimodal shape densities, segmentation results obtained from multiple modes might be interesting and might offer reasonable solutions. Figure 4.10 shows some visual results obtained from the experiments on the MNIST data set. For each test

Test Image	Digit Class									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
MNIST - 1	336	433	6	18	29	38	115	16	8	1
MNIST - 2	4	691	8	3	96	9	0	120	3	66
MNIST - 3	119	661	8	1	2	11	154	14	28	2

Table 4.1: Number of samples generated for each digit class in test images from the MNIST data set.

image, we display the results from the best three digit classes where, the quality of each class is computed as the average energy, $E(x)$, of the samples in that class. Also, for each class, we show three samples having the best energy values. These results show that our algorithm is able to find reasonable solutions from different modes of the posterior density. In Figure 4.10, we also present marginal confidence bounds (MCB images) obtained from the samples in each class. The figure demonstrates the marginal confidence bounds at different levels of the histogram image, $H(x)$, for the best classes in all test images. $H(x) = 0.1$ and $H(x) = 0.9$ indicate the low probability and the high probability regions, respectively.

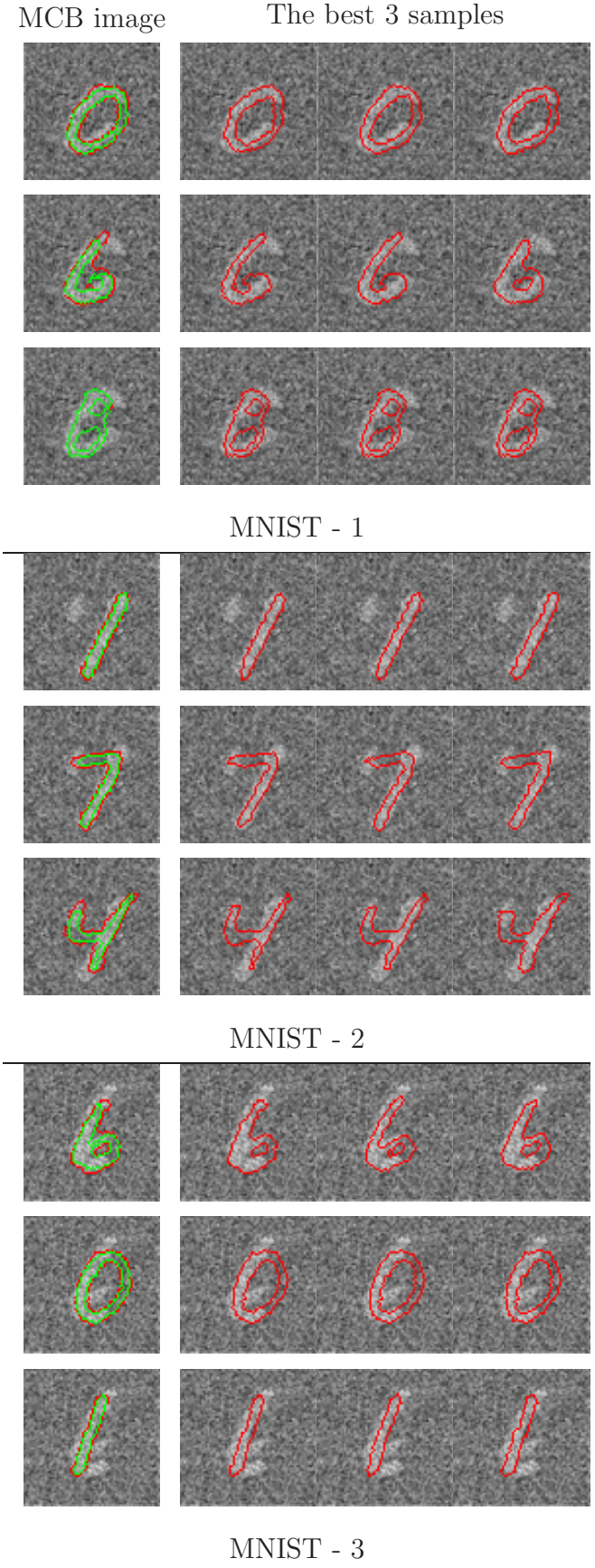


Figure 4.10: Experiments on the MNIST data set. Note that in MCB images, red and green contours are the marginal confidence bounds at $H(x) = 0.1$ and $H(x) = 0.9$, respectively.

Experiments on the walking silhouettes data set

In this experiment, we test the performance of local shape priors extension of our MCMC shape sampling approach and compare it with the one that uses global shape priors, as well as with the method of Kim et al. [1]. We choose a subset of 30 binary images of a walking person from the walking silhouettes data set [17]. A subset of 16 images shown in Figure 4.11 among these 30 binary images are used for training. The remaining 14 binary images are used to construct test images by adding a high amount of noise.

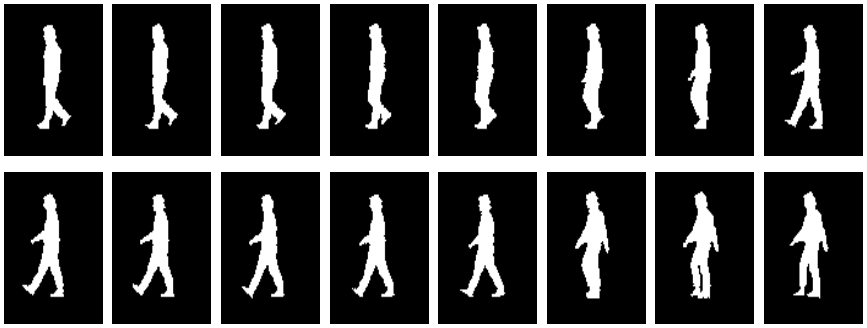


Figure 4.11: The training set for the walking silhouettes data set.

We present results on all test images in Figure 4.12. Similar to the evaluations performed for the aircraft data set, we plot the PR values for each sample obtained by our approaches (with global and local priors) and by the approach of Kim et al. [1]. According to the results, our proposed approach with global shape priors produces samples that have F-measure values better than or equal to the result of Kim et al. [1] in all test images. By using local shape priors, we can generate even better samples than both Kim et al. [1] and the approach with global shape priors. Moreover, it seems that our approach based on local shape priors is able to sample the space more effectively than the approach with global shape priors.

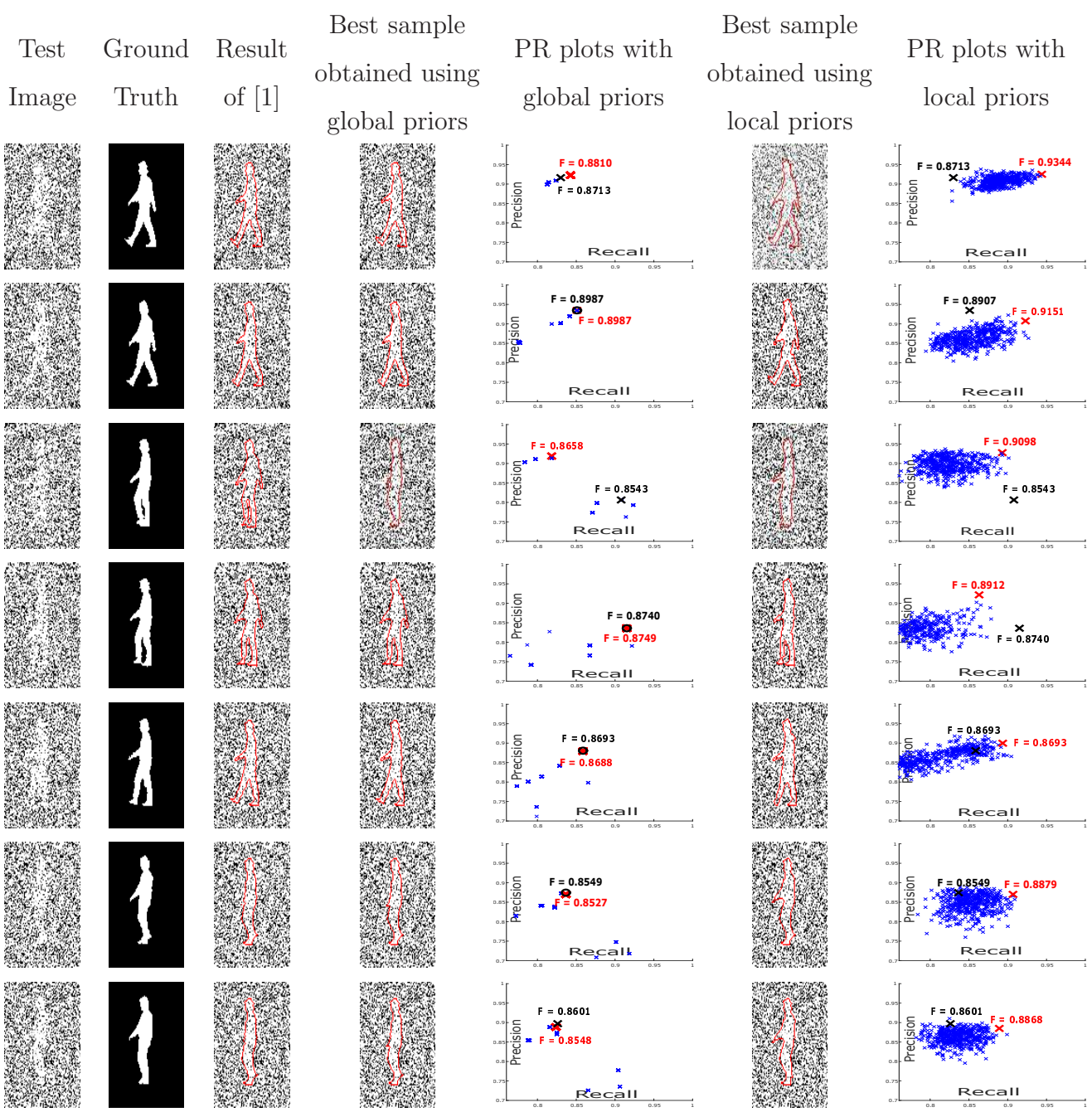


Figure 4.12: Experiments on walking silhouettes data set. In the PR curves, the '×'marks the sample having the best F-measure value obtained using the proposed approach (with either global or local shape priors), and the '×'marks that of segmentation of Kim et al. [1].

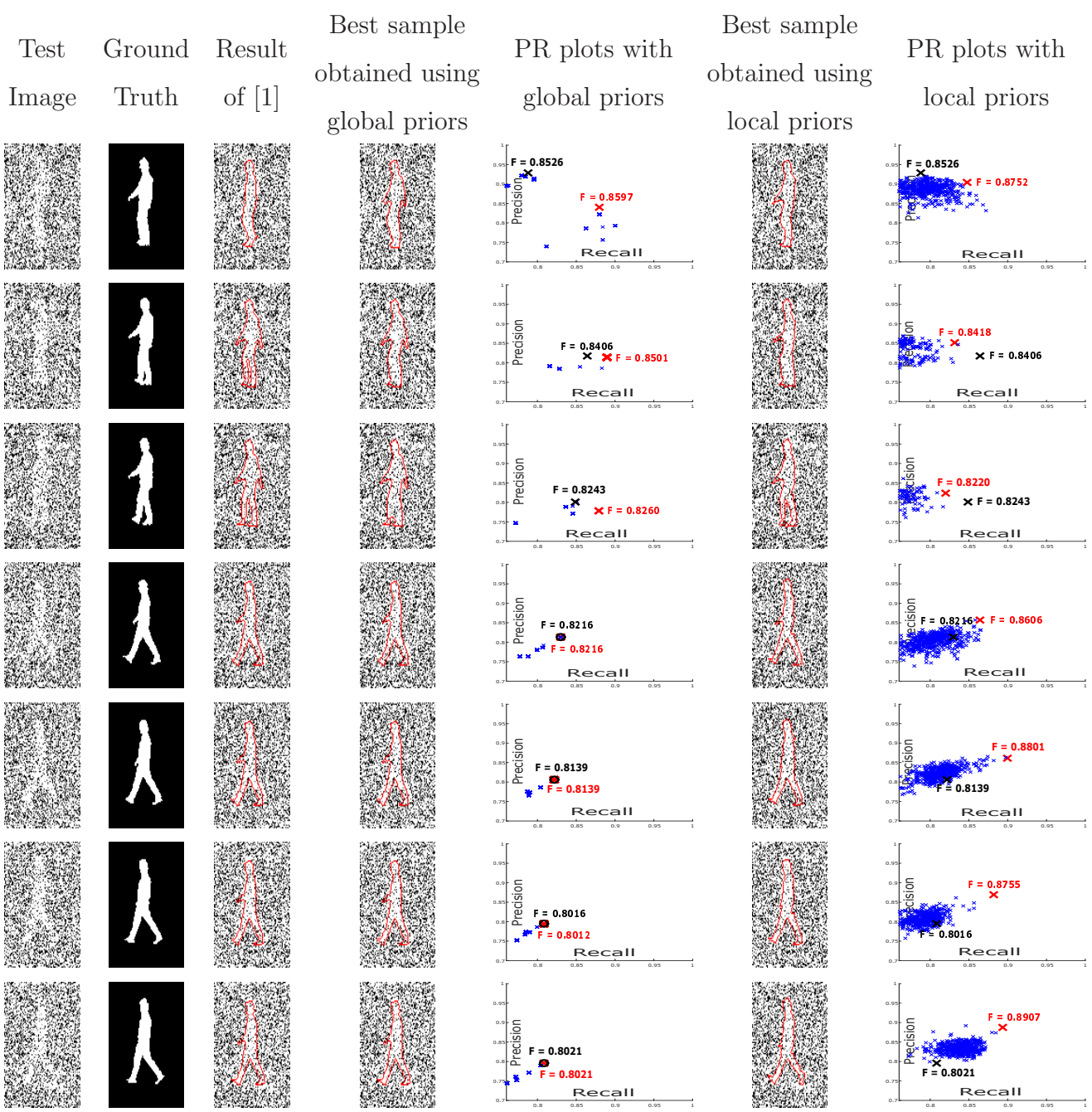


Figure 4.12 (cont.): Experiments on walking silhouettes data set. In the PR curves, the '×'marks the sample having the best F-measure value obtained using the proposed approach (with either global or local shape priors), and the '×'marks that of segmentation of Kim et al. [1].

4.3.7 Conclusion

We have presented a MCMC shape sampling approach for image segmentation that exploits prior information about the shape to be segmented. Unlike exist-

ing MCMC sampling methods for image segmentation, our approach can segment objects with occlusion and suffering from severe noise, using nonparametric shape priors. We also provide an extension of our method for segmenting shapes of objects with parts that can go through independent shape variations by using local shape priors on object parts. Empirical results on various data sets demonstrate the potential of our approach in MCMC shape sampling. The implementation of the proposed method is available at `spis.sabanciuniv.edu/data_code`.

4.4 Pseudo-marginal MCMC sampling for image segmentation using nonparametric shape priors

In this section, we introduce our pseudo-marginal MCMC shape sampling approach for image segmentation exploiting nonparametric shape priors.

4.4.1 Contribution

Our contributions in this work is a pseudo-marginal Markov chain Monte Carlo (MCMC) sampling-based image segmentation approach that exploits nonparametric shape priors. We incorporate the nonparametric shape priors with the observed data in Bayesian framework and generate samples from the resulting posterior distribution. The proposed approach is able to segment objects that suffer from severe occlusion, noise and missing data. Our pseudo-marginal MCMC sampling approach is able to characterize the multimodal posterior shape densities through its samples and avoids the problem of getting stuck at a single local optimum.

MCMC sampling approaches generally become inefficient when the size of the data set increases. In our approach, we deal with this problem by using a subsampling procedure called pseudo-marginal sampling which still guarantees having samples from the posterior density without using all examples in a given training set. To the best of our knowledge, pseudo-marginal sampling have not been used in the literature for a sampling-based image segmentation method before.

Satisfying necessary conditions to implement MCMC sampling is a crucial step

for developing an MCMC sampler. In order to satisfy these conditions, a proposal distribution should be defined in a proper way so that the probability of generating a new sample given the previous one and the probability of generating the previous sample given the new one are computed correctly. Defining such a proposal distribution is not trivial. Therefore, to the best of our knowledge, all MCMC-based segmentation approaches in the literature approximate to these probabilities including the one introduced in Section 4.3. In our approach, we define a proposal distribution for exact computation of these probabilities. This guarantees obtaining samples from the desired distribution; the posterior density in our case.

This work advances the work presented in Section 4.3 in several major ways. In particular, (1) while the method in Section 4.3 approximately satisfy the necessary conditions of MCMC sampling, in this work we perfectly satisfy these conditions; (2) we use pseudo-marginal sampling to be able to learn very large data sets; the one in Section 4.3 becomes inefficient when the size of the data set increases.

4.4.2 Model and problem definition

Probabilistic model for image segmentation

The image segmentation problem consists of estimating an unknown segmenting curve for an object that belongs an unknown class given an observed image $y \in \mathcal{Y}^{M\times N}$ of size $M \times N$ where $\mathcal{Y}$ is the set of the values that the pixels of y can take, e.g. the integers from 0 to 255. We denote the class of the object by $s \in \{1, \ldots, n\}$ where $n \geq 1$ is the total number of classes and it is known. For simplicity we assume that s has a uniform distribution over $\{1, \ldots, n\}$ so that

$$p(s) = 1/n, \quad s = 1, \ldots, n. \tag{4.11}$$

The segmenting curve we ultimately want to estimate is essentially a binary image $c \in \{0, 1\}^{M\times N}$ having the same size with y, where 0's indicate background and 1's indicate the object.

The conditional density of y given c, or the likelihood, is independent from s and denoted as $L(y|c)$. We use piecewise-constant version of the Mumford-Shah

functional [66], [55]:

$$L(y|c) = \exp\left\{ -\sum_{(i,j)\in c_{\text{in}}} (y(i,j) - \mu_{\text{in}})^2 - \sum_{(i,j)\in c_{\text{out}}} (y(i,j) - \mu_{\text{out}})^2 \right\} \tag{4.12}$$

where c_{in} (c_{out}) is the region inside (outside) of the curve c and μ_{in} and μ_{out} are the average intensities in c_{in} and c_{out}

$$\mu_{\text{in}} = \frac{1}{|c_{\text{in}}|} \sum_{(i,j)\in c_{\text{in}}} y(i,j), \quad \mu_{\text{out}} = \frac{1}{|c_{\text{out}}|} \sum_{(i,j)\in c_{\text{out}}} y(i,j)$$

One can consider using more sophisticated likelihood terms such as mutual information [67], J-Divergence [68], Bhattacharya Distance [69] and learning-based [57], [3].

We also have a training set of binary curves that are grouped into classes, that is we have

$$\mathbf{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_n\},$$

where each $\mathcal{C}_i = \{c_{i,1}, \ldots, c_{i,m_i}\}$ is the collection of $m_i \geq 1$ segmented curves for class i.

Level set representation

In this work, we construct the a non-parametric prior distribution on the unknown c given its class s using the training set $\mathbf{C}$. However, we choose to define this prior by using the level set representation. Level set representation is essentially a mapping

$$\phi : \{0,1\}^{M\times N} \to \mathbb{R}^{MN}$$

from the binary space to the real space. In the literature, it has been found more convenient to work with level sets to represent c to handle topological shape changes. As we will show below, another motivation for us to work with x instead of c is efficient use of gradients with respect to x in our methodology. Therefore we define the level set variable $x = \phi(c)$ and work with x during the rest of this section. Let us also define the level set representation of the training set as well:

$$\mathbf{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\},$$

where each $\mathcal{X}_i = \{x_{i,1}, \ldots, x_{i,m_i}\}$ with $x_{i,j} = \phi(c_{i,j})$, the level set representation of $c_{i,j}$. Now we can define the prior distribution for x given the class s. We use the Parzen density estimator for each class, leading to

$$p(x|s) = \frac{1}{m_s} \sum_{i=1}^{m_s} \mathcal{N}(x; x_{s,i}, \sigma^2 I) \tag{4.13}$$

where $\mathcal{N}(x; \mu, \Sigma)$ is the (possibly multivariate) gaussian density with mean μ and covariance Σ. This prior corresponds to a mixture of kernels (Parzen densities) with centres $x_{s,1}, \ldots, x_{s,m_s}$ with kernel size σ. Kernel density estimation is widely used in related context, see [1], [17] for examples. To determine the kernel size σ, we use an ML kernel with leave-one-out [32].

Note that ϕ is not invertible; however with an abuse of notation, we define a pseudo-inverse

$$\bar{\phi} : \mathbb{R}^{MN} \to \{0, 1\}^{M \times N}$$

that satisfies $\bar{\phi}(\phi(c)) = c$. The function $\bar{\phi}$ maps the level set representation back to binary segmenting curve such that if $x = \phi(c)$ we have $\bar{\phi}(x) = c$. We will need $\bar{\phi}$ to rewrite the likelihood in terms of y:

$$p(y|x) = L(y|\bar{\phi}(x)). \tag{4.14}$$

Bayesian formulation

Combining (4.11), (4.13), and (4.14) in terms of x, we can now write the joint density of s, x, and y

$$p(s, x, y) = p(s)p(x|s)p(y|x). \tag{4.15}$$

Bayesian image segmentation problem can be formulated as finding the posterior distribution of x and y

$$p(x|y) \propto p(y|x)p(x) = p(y|x) \sum_{s=1}^{n} p(s)p(x|s).$$

However, estimating $p(x|y)$ can be difficult since the summation over classes makes the distribution hard to infer, e.g. using Monte Carlo sampling methods. Alternatively, we aim for the joint posterior distribution of s and x given y

$$p(s, x|y) \propto p(s, x, y) \tag{4.16}$$

whose marginal is still the desired posterior $p(x|y)$. In the following section, we will describe an MCMC method to efficiently sample from $p(s, x|y)$.

4.4.3 Methodology

Metropolis-Hastings within Gibbs

An MCMC algorithm is based on updating the samples for the variable of the posterior distribution. One of the most popular MCMC algorithm is Metropolis-Hastings [40]. We could apply MH for $p(s, x|y)$ by proposing a candidate sample (s', x') and then accept or reject it with an acceptance probability. However, attempting to change and s and x at the same time may not be efficient because of the lack of intelligent proposal mechanisms which would lead to few instances of acceptance and hence a sticky Markov chain.

On the other hand, a sensible candidate for the new sample of x can be generated when s is kept the same by using the gradient information of the conditional distribution given s. This is possible by Gibbs-like moves of some random variables by conditioning the posterior density on the remainings . That is why we adopt a version of Gibbs sampling where one has updates for s and x in an alternating fashion [42], [43]. However, since the full conditional $p(x|s, y)$ is hard to sample from, we update x by using an MH move, which leads to the well known Metropolis-Hastings within Gibbs (MHwG) algorithm. MHwG for image segmentation is presented in Algorithm 5.

Algorithm 5 MHwG for $p(s, x|y)$

1: Initialize $x^{(0)}$, $s^{(0)}$.
2: **for** $t = 1 \rightarrow N$ **do**
3: Sample $s^{(t)} \sim p(s|y, x^{(t-1)}) \propto p(s)p(x^{(t-1)}|s)$
4: Use an MH move for $p(x|y, s^{(t)})$ to update $x^{(t-1)}$ to $x^{(t)}$
5: **end for**

Computational complexity of MHwG and subsampling

Observe that both conditional densities in Algorithm 5 involve $p(x|s)$ which needs to be evaluated during MH updates. This requires evaluation of m_s multivariate densities of dimension MN. This can be too costly when m_s is large, which occurs when we have a big the training set.

Towards a more computationally efficient MCMC algorithm that scales with the training data size, we consider the following unbiased estimator of $p(x|s = i)$ via subsampling:

$$\widehat{p}(x|s) = \frac{1}{\hat{m}_s} \sum_{j=1}^{\hat{m}_s} \mathcal{N}(x; x_{s,u_j}, \sigma^2 I) \tag{4.17}$$

where

$$\{u_1, u_2, \ldots, u_{\hat{m}_s}\} \subset \{1, 2, \ldots, m_s\} \tag{4.18}$$

is a subsample generated via sampling without replacement and $\hat{m}_s \ll m_s$. This approximation of the prior leads to the approximation of the conditional posterior densities:

$$\widehat{p}(s|x, y) \propto p(s)\widehat{p}(x|s) \tag{4.19}$$

$$\widehat{p}(x|s, y) \propto \widehat{p}(x|s)p(y|x) \tag{4.20}$$

However, using this approximation does not generally guarantee that the Markov Chain have an equilibrium distribution that is exactly $p(x, s|y)$. In order to deal this issue, we adopt the pseudo-marginal MH algorithm of [85] in the next section.

4.4.4 The proposed method

Assume that we have a non-negative random variable z such that given x and s, its conditional density $g_{s,x}(z)$ satisfies

$$\int_0^\infty g_{s,x}(z) z dz = p(x|s).$$

In our setting z is our approximation to $p(x|s)$, i.e. $z = \widehat{p}(x|s = i)$, and its probability density $g_{s,x}(z)$ corresponds to the generation process of this approximation.

(It will become clear that in fact we do not have to calculate $g_{s,x}(z)$ at all but we should be able to sample from it.) Define the extended posterior density with the new variable z added

$$p(x, s, z|y) \propto p(s) z g_{s,x}(z) p(y|x) \tag{4.21}$$

When we integrate z out, we see that samples for s and x from (4.21) will admit the desired posterior in (4.16):

$$p(s)p(y|x) \int z g_{s,x}(z) dz = p(s)p(x|s)p(y|x)$$

Now, the problem of generating samples from the posterior density in (4.16) can be replaced with the problem of generating samples from the posterior density in (4.21).

We propose a pseudo-marginal MHwG sampling to generate samples from $p(x, s, z|y)$. Note the important remark that this algorithm also targets $p(x, s|y)$, hence $p(x|y)$ exactly. General steps of our sampling algorithm is given in Algorithm 6. In step 3 we condition the posterior on x and update s and z. Analogously, in step 4, we update x and z by conditioning the posterior on s. In the following, we explain in detail how to perform those steps. We also give an elaborate description of how we propose the candidate values for the level set, x, which constitutes an important part of our novel algorithm from a practical point of view.

Algorithm 6 Pseudo-marginal MHwG - generic

1: Initialize $x^{(0)}$, $s^{(0)}$, and $z^{(0)}$.
2: **for** $t = 1 \to N$ **do**
3: Use an MH move for $p(s, z|y, x^{(t-1)})$ that updates $(s^{(t-1)}, z^{(t-1)})$ to $(s^{(t)}, z^{(t)})$.
4: Use an MH move for $p(x, z|y, s^{(t)})$ that updates $(x^{(t-1)}, z^{(t-1)})$ to $(x^{(t)}, z^{(t)})$.
5: **end for**
6: Outputs samples $(s^{(1)}, x^{(1)}), \ldots, (s^{(N)}, x^{(N)})$.

Update step for the class s and z

The distribution that we sample s and z in Metropolis-Hastings is $p(s, z|y, x)$. We can write this distribution as

$$p(s, z|y, x) = p(s, z|x) \propto p(s) z g_{s,x}(z). \tag{4.22}$$

Since we regard the proposal mechanism as a joint update of s and z, it is clear that the proposal generates (s', z') from the density $q(s'|s)g_{s',x}(z)$. Note that (s', z') denote the candidate samples generated from the proposal distribution. We assume that $q(s'|s)$ is a uniform distribution $\mathcal{U}\{1, n\}$ and $z' = \widehat{p}(x|s')$. Once s' and z' are sampled, they are either accepted with probability

$$\begin{aligned} &\min\left\{1, \frac{p(s')z' g_{s',x}(z') q(s|s') g_{s,x}(z)}{p(s) z g_{s,x}(z) q(s'|s) g_{s',x}(z')}\right\} \\ =&\min\left\{1, \frac{p(s')z' q(s|s')}{p(s) z q(s'|s)}\right\}, \end{aligned} \tag{4.23}$$

or keep the current values of s and z. Steps of sampling s and z from $p(s, z|y, x)$ are shown between steps 4-12 of Algorithm 7. Note that these steps correspond to step 3 of Algorithm 6. Also note that we can exactly compute the probabilities in the MH ratio in (4.23) which guarantees satisfying necessary conditions to implement MCMC sampling.

Update step for the level set x and z

The next step is to sample x and z from the conditional density $p(x, z|y, s)$ as shown in step 4 of Algorithm 6. To achieve this, we perform a Metropolis-Hastings sampling as we use for sampling $s^{(t)}$ and $z^{(t)}$. The conditional density $p(x, z|y, s)$ can be written as

$$p(x, z|s, y) \propto z g_{s,x}(z) p(y|x). \tag{4.24}$$

Also, for joint sampling of candidates (x', z') we can write the proposal density as

$$q_{s,j}(x', z'|y) = q_{s,j}(x'|x, y) g_{s,x'}(z'). \tag{4.25}$$

where j is sampled uniformly from $\{1, m_s\}$, and $z' = \widehat{p}(x'|s)$ is generated using subsampling. (The details of $q_{s,j}(x'|x, y)$ will be explained in Section 4.4.4.) Then,

the Metropolis-Hastings ratio can be computed as follows:

$$\begin{aligned} &\min\left\{1, \frac{z' g_{s',x}(z') p(y|x') q_{s,j}(x|x', y) g_{s,x}(z)}{z g_{s,x}(z) p(y|x) q_{s,j}(x'|x, y) g_{s',x}(z')}\right\} \\ =&\min\left\{1, \frac{z' p(y|x') q_{s,j}(x|x', y)}{z p(y|x) q_{s,j}(x'|x, y)}\right\} \end{aligned} \tag{4.26}$$

Proposal mechanism for the level set

The crucial part in (4.26) is designing the proposal distribution to generate a candidate curve x' from x. Let us define the energy function corresponding to the training images in class s:

$$E_s(x) := \log p(x|s) + \log p(y|x, s)$$

When the training data set is too large, calculating $\log p(x|s)$ from (4.13) may be too expensive as discussed earlier. An unbiased estimator of $E_s(x)$ would be obtained as

$$E_{s,j}(x) := \log \mathcal{N}(x; x_{s,j}, \sigma^2 I) + \log p(y|x, s)$$

where $j \sim \mathcal{U}(1, \ldots, m_s)$. The proposal distribution after sampling the j^{th} training image in class s is then constructed as follows:

$$q_{s,j}(x'|x, y) = \mathcal{N}\left(x'; x - \widehat{\nabla E}_{s,j}(x), \Sigma\right). \tag{4.27}$$

Here, the shift term $\widehat{\nabla E}_{s,j}(x)$ is an approximation to the gradient $\nabla E_{s,j}(x)$ w.r.t. x and it is given by

$$\widehat{\nabla E}_{s,j}(x) = \frac{1}{\sigma^2}(x - x_{s,j}) + \left[(y - \mu_{\text{in}})^{\odot 2} - (y - \mu_{\text{out}})^{\odot 2}\right] \tag{4.28}$$

where $(\cdot)^{\odot k}$ is the element-wise power operation for x.

In (4.28), the first term $(x - x_{s,j})/\sigma^2$ is the gradient of $\log \mathcal{N}(x, x_{s,j}, \sigma^2 I)$ and the term $(y - \mu_{\text{in}})^{\odot 2} - (y - \mu_{\text{out}})^{\odot 2}$ is a discrete approximation w.r.t. the level set representation x [55], so that the expression in (4.28) altogether is an approximation to $\nabla E_{s,j}(x)$.

Design of the proposal covariance

In the design of the proposal distribution in (4.27), the most important part is finding a co-variance matrix, Σ, that generates smooth perturbations. Generating

Algorithm 7 Pseudo-marginal MHwG - detailed

Initialize $s^{(0)}, x^{(0)}$ and generate an approximation of the prior $z^{(0)} = \widehat{p}(x^{(0)}|s^{(0)})$ using subsampling as in (4.17).

for $t = 1 \to N$ **do**

  Set $s = s^{(t-1)}$, $x = x^{(t-1)}$, and $z = z^{(t-1)}$.

  Sample $s' \sim q(s'|s)$.

  Generate an approximation $z' = \widehat{p}(x|s')$ of $p(x|s')$ using a subsample of size $\hat{m}_{s'}$ where $\hat{m}_{s'} \ll m_{s'}$.

  Compute Metropolis-Hastings acceptance ratio:

$$\alpha = \min\left\{1, \frac{p(s')z'q(s|s')}{p(s)zq(s'|s)}\right\},$$

  Sample $\eta \sim \mathcal{U}(0,1)$.

  if $\alpha > \eta$ **then**

    $s^{(t)} = s'$; $z^{(t)} = z'$; ▷ Accept the candidate

  else

    $s^{(t)} = s$; $z^{(t)} = z$; ▷ Reject the candidate

  end if

  Set $s = s^{(t)}$, and $z = z^{(t)}$.

  Uniformly draw a random number j in $\{1, \ldots, m_s\}$.

  Sample x' from $q_{s,j}(x'|x,y) = \mathcal{N}\left(x'; x - \widehat{\nabla E}_{s,j}(x), \Sigma\right)$.

  Generate an approximation $z' = \widehat{p}(x'|s)$ of $p(x'|s)$ using a subsample of size $\hat{m}_s$.

  Compute Metropolis-Hastings acceptance ratio:

$$\alpha = \min\left\{1, \frac{z'p(y|x')q_{s,j}(x|x',y)}{zp(y|x)q_{s,j}(x'|x,y)}\right\},$$

  Sample $\eta \sim \mathcal{U}(0,1)$.

  if $\alpha > \eta$ **then**

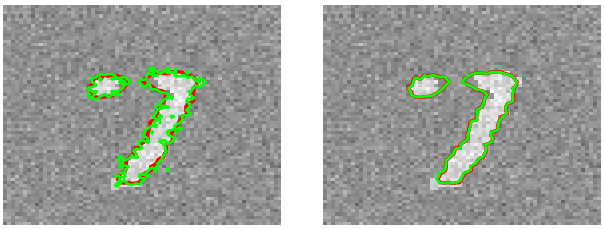
    $x^{(t)} = x'$; $z^{(t)} = z'$; ▷ Accept the candidate

  else

    $x^{(t)} = x$; $z^{(t)} = z$; ▷ Reject the candidate

  end if

end for



(a) (b)

Figure 4.13: Perturbation of a curve (red) with (a) unfiltered noise and (b) smoothed noise. Note that green indicates curves obtained after perturbing the curve shown by red.

candidate curve by smoothly and randomly perturbing the current curve is required for an MCMC shape sampling approach since smoother curves are high likely [1], [77]. This can also be observed from the visual example shown in Figure 4.13. In our proposal $\mathcal{N}(x'; x - \widehat{\nabla E}_{s,j}(x), \Sigma)$, this can be achieved by finding a co-variance matrix that yields a smooth perturbation. Therefore, Σ should be obtained such that the neighboring pixels on the curve have higher correlation than the ones that are not close. Obtaining such Σ is not trivial since it should also be decomposed such that $\Sigma = AA^T$. For this reason, most of the methods in the literature generate a Gaussian noise with identity co-variance matrix, smooth this noise with a low-pass filter and perturb the curve with the smoothed Gaussian noise. Although, such an approach practically works, the Metropolis-Hastings acceptance ratio cannot be computed properly which results the failure of the necessary MCMC conditions.

In the proposed approach, we compute a positive semi-definite co-variance matrix Σ such that it generates smooth random perturbations. The proposed algorithm to produce Σ is given in Algorithm 8. Σ has to be positive semi-definite to decompose it into $\Sigma = AA^T$.

Given an $M \times N$ image, we first generate an $MN \times MN$ matrix Z from unit Gaussian distribution. Then, we construct another $MN \times MN$ matrix F where each column of F is a smoothed version of each row in Z. By assuming F is constructed by multiplying Z by a matrix $\widehat{A}$, we can find matrix $\widehat{A}$ as follows

$$\widehat{A} = Z^{-1}F. \tag{4.29}$$

As Z is generally reversible, $\widehat{A}$ can be computed using (4.29). Given $\widehat{A}$, a co-variance

matrix $\widehat{\Sigma}$ can be computed as

$$\widehat{\Sigma} = \widehat{A}\widehat{A}^T. \tag{4.30}$$

However, generally $\widehat{\Sigma}$ is not positive semi-definite since $\widehat{A}$ is not generally a full rank matrix. Therefore, we find the closest positive semi-definite matrix to $\widehat{\Sigma}$ using the approach [86]. The closest positive semi-definite matrix is the co-variance matrix Σ we use in our shape proposal distribution. Since Σ is computed that results expected perturbation, the probabilities in the Metropolis-Hastings ratio can be exactly computed. This helps us to satisfy necessary MCMC conditions which guarantees having an equilibrium distribution that is exactly the posterior density.

Algorithm 8 Design of the proposal covariance

1: Initialize an $MN \times MN$ matrix Z whose elements are randomly generated from unit Gaussian distribution.
2: Initialize an empty $MN \times MN$ matrix F.
3: **for** $i = 1 \rightarrow MN$ **do**
4: Assign i^{th} row of Z into z.
5: Construct $\widehat{z}$ by smoothing z using a low-pass filter.
6: Assign $\widehat{z}^T$ into the i^{th} column of F.
7: **end for**
8: $\widehat{A} = Z^{-1}F$
9: $\widehat{\Sigma} = \widehat{A}\widehat{A}^T$
10: Compute Σ, the closest positive semi-definite matrix to $\widehat{\Sigma}$ using the approach in [86].

4.4.5 Experimental results

In this section, we present experimental results of our pseudo-marginal MCMC sampling approach on object segmentation. We perform experiments on various data sets each describe different advantages of using a sampling-based approach over optimization-based ones.

Degree of confidence in segmentation results

Sampling-based segmentation approaches can provide a measure of the degree of confidence in segmentation results unlike the optimization-based approaches. In this experiment, we demonstrate this advantage of the proposed approach on the aircraft data set [1]. The aircraft data set [1] contains 11 synthetically generated binary aircraft images as shown in the first row of Figure 4.14.

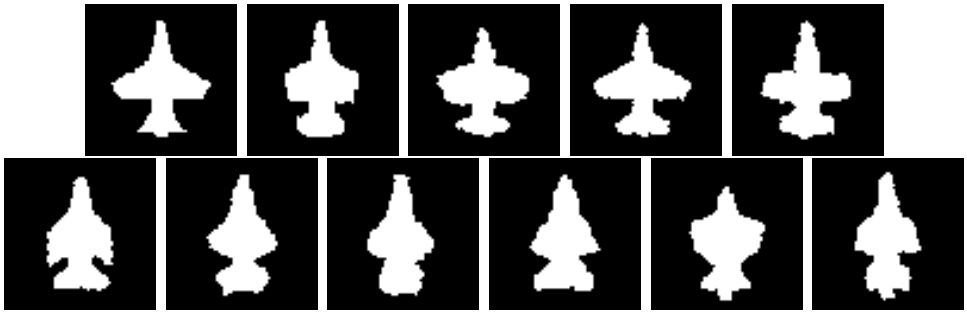


Figure 4.14: The aircraft data set.

It is expected that degree of confidence on segmentation results decreases as the noise level in the image increases. However, optimization-based segmentation approaches do not give any idea about the degree of confidence on their segmentations since they generate a solution at a local optimum. In order to demonstrate this advantage of our approach, we generate test images with different noise levels as shown in Figure 4.15. Note that each test image is constructed by adding different amount of noise to the images in Figure 4.14. When segmenting a particular test image, we exclude the corresponding binary image from the images in Figure 4.14 and use the rest for training. We generate 1000 samples using the proposed approach on each test image. We take the average of all binary samples for each test image and threshold the average image at 0 and 0.9 to find the least and most confidence marginal boundaries, respectively. The results highlighting these boundaries are shown in Figure 4.15. Note that the distance between the least and most confidence boundaries increases as the noise level increases. This also states that the uncertainty in the segmentation results increases as the noise level get higher, which is expected. Our sampling-based segmentation approach can give idea about the degree of confidence on the segmentation results by generating samples from the posterior density.

Table 4.2: Standard deviation of Dice scores between each sample and ground truth for each test image.

	Noise level			
Test Image	**0**	**10**	**50**	**100**
1	0.0002	0.0019	0.0051	0.0079
2	0.0004	0.0016	0.0042	0.0078
3	0.0004	0.0015	0.0048	0.0119
4	0.0003	0.0046	0.007	0.0128
5	0.0003	0.004	0.0067	0.0134
6	0.0005	0.0037	0.0079	0.0127
7	0.0002	0.001	0.0138	0.0101
8	0.0005	0.0013	0.0034	0.0193
9	0.0003	0.0019	0.0067	0.0119
10	0.0003	0.0011	0.0045	0.0077
11	0.0004	0.0013	0.0079	0.009
Average	0.000345	0.002173	0.006545	0.011318

We also obtain quantitative results that supports the visual results in Figure 4.16. We compare each samples with the ground truth using Dice score [70] and compute standard deviation for each test image as shown in Table 4.2. Quantitative results also demonstrate that the standard deviation of Dice score results between each sample and ground truth increases together with the noise level in the test image.

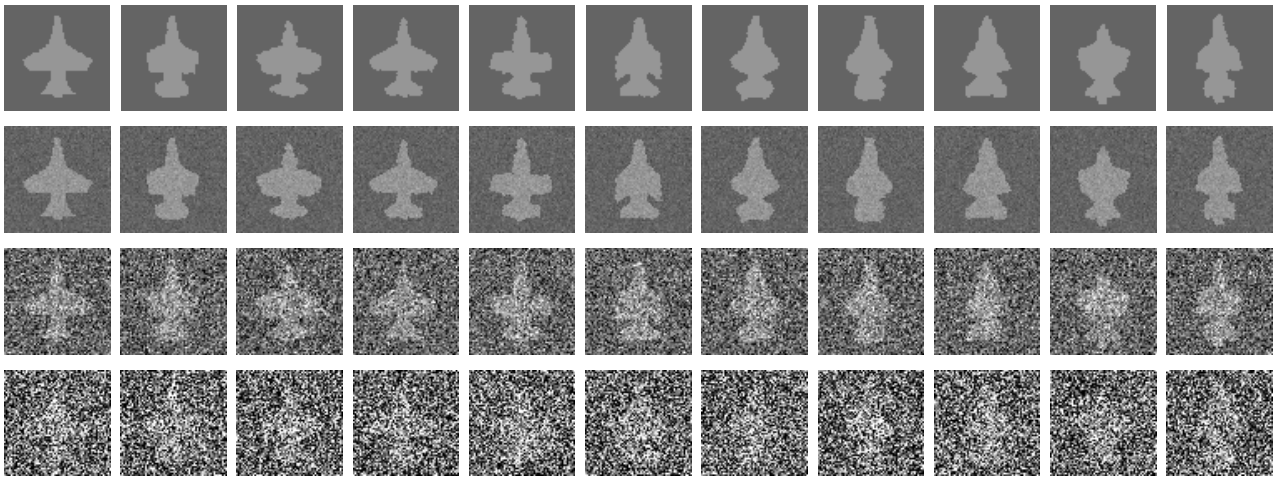


Figure 4.15: Test images used in the experiments with the aircraft data set. Note that the noise level in the test images increases from top to bottom.

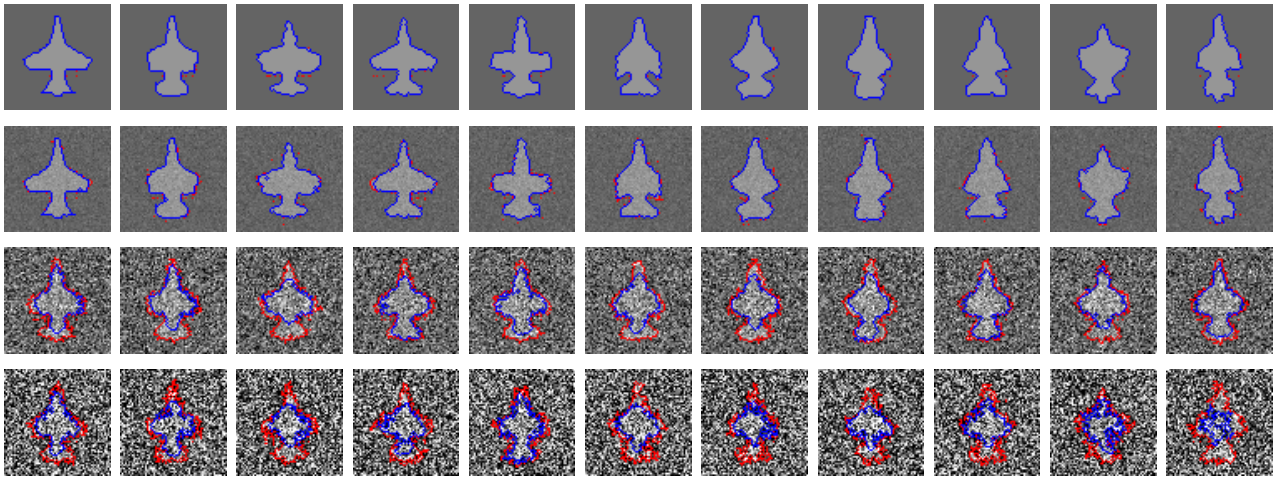


Figure 4.16: Marginal confidence bounds obtained from samples for each test image in the experiments with the aircraft data set. Note that red indicates the least confidence boundary whereas blue indicate the most confidence boundary.

Time vs. accuracy

MCMC methods are known to be inefficient when the training set size become larger. The proposed pseudo-marginal MCMC sampling approach deals with this problem by computing an unbiased estimator of the proposal density in each sampling iteration. In this section we perform experiments on the MNIST data set [84]. The MNIST data set consists of $60,000$ training and $10,000$ test examples of binary handwritten digits. A subset of examples from the training set are shown in Fig-

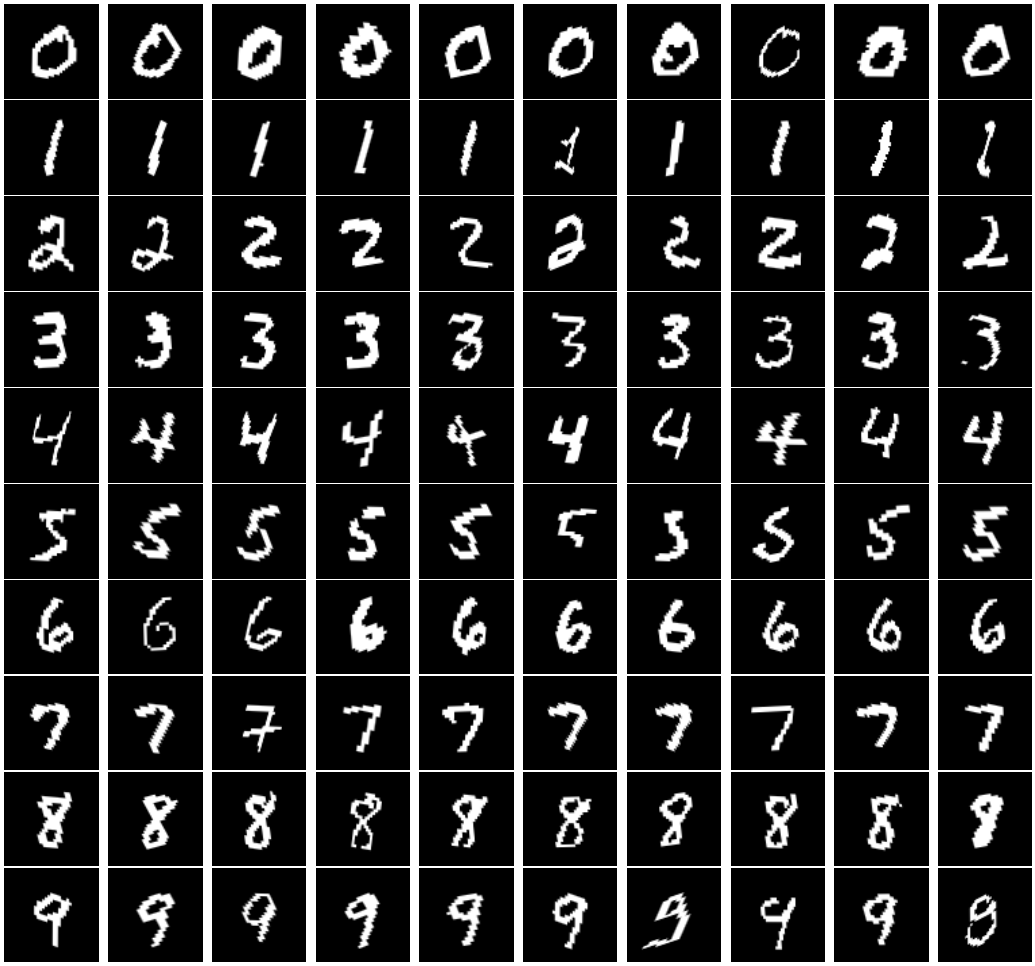
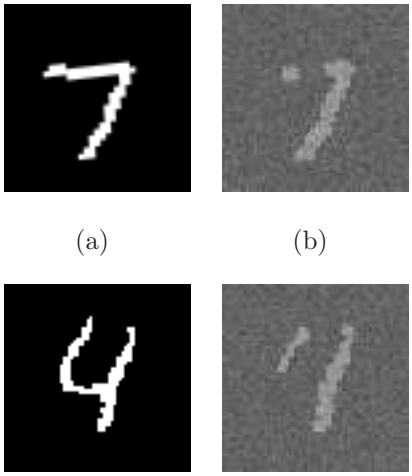


Figure 4.17: A subset of training examples from the MNIST data set.

ure 4.17. The data set is generally used for classification task. In this experiment, we take one of the images from the test set (see Figures 4.18(a) and 4.18(c)), artificially occlude some part of the digit and introduce some noise. We use the resulting images shown in Figure 4.18(b) and 4.18(d) to test the proposed approach.

We apply our pseudo-marginal shape sampling approach for segmentation of the image shown in Figure 4.18(b). In this experiment, we construct training sets by taking subsets of MNIST training set with different sizes. We segment the test image in Figure 4.18(b) for each of these training sets using our pseudo-marginal shape sampling approach. We also perform the same experiments without using pseudo-marginal sampling (conventional MCMC sampling) by computing the posterior probability using each training example in each sampling iteration. In each experiments we generate 1000 samples and record the average time for generating a single sample. The plot in Figure 4.19 shows average running time as a function of



(a) (b)

(c) (d)

Figure 4.18: Using MNIST test images for segmentation: (a)-(c) images from the MNIST test set, (b)-(d) occluded and noisy version of the images in (a)-(c) for segmentation task.

training set size for both pseudo-marginal sampling and conventional MCMC sampling. The average single sample generation time of the proposed approach does not change as training set size increases since we choose $\hat{m}_i = 10$ regardless of the training set size in all experiments.

We also measure the effect of using pseudo-marginal sampling on the accuracy of the generated samples. We compare each sample generated by both pseudo-marginal sampling and conventional MCMC sampling with the ground truth using Dice score [70][2]. The average Dice score results are given in Table 4.3. Note that the average Dice score results are very close to each other for pseudo-marginal and conventional MCMC shape sampling approaches. The very slight decrease in Dice score results of pseudo-marginal sampling can be acceptable in many applications when the huge gain in computation time is considered.

[2]Note that our algorithm can produce samples from different digit classes. In order to have a fair Dice score comparisons, we run our sampling algorithm several times until all samples are from the same class with the ground truth.

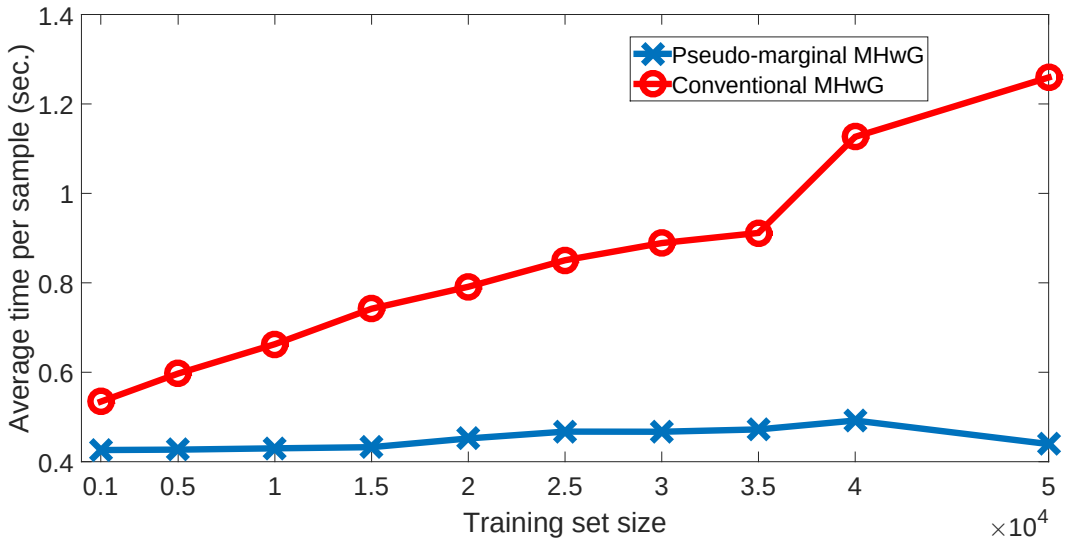


Figure 4.19: Average running time for producing a single sample as a function of training set size for both pseudo-marginal MHwG shape sampling and conventional MHwG shape sampling.

Table 4.3: Average Dice Score results of all samples for each experiment with different training set sizes.

Training Set Size	**Pseudo-marginal Shape Sampling**	**Conventional MCMC Sampling**
1000	0.7736	0.7758
5000	0.7735	0.7818
10000	0.7744	0.7765
15000	0.7756	0.7745
20000	0.7706	0.7791
25000	0.7689	0.7776
30000	0.7703	0.7767
35000	0.7700	0.7769
40000	0.7691	0.7801
50000	0.7719	0.7776
Average	0.7718	0.7777

A common way of comparing two MCMC sampling approach is to plot the

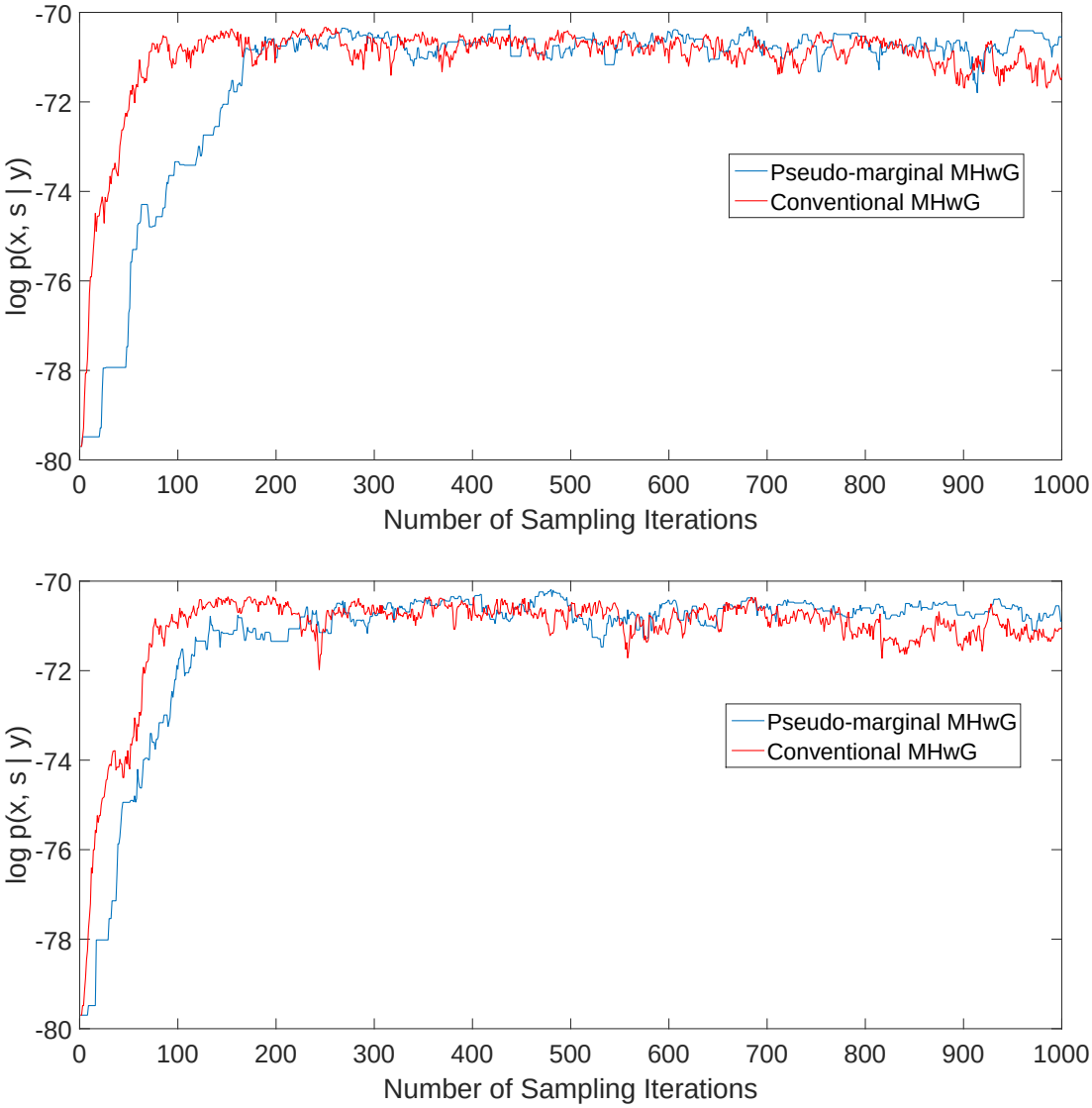


Figure 4.20: Log posterior probabilities for (left) 5000, (right) 50000 training samples

posterior probability (or logarithm of posterior probability) of each sample. We show these plots for both pseudo-marginal sampling and conventional MCMC sampling in Figure 4.20. We compute logarithm of posterior (log posterior) probabilities in the plots for training sets that contain 5000 and 50000 examples. Note that the log posterior probabilities of both both approaches are very close to each other. This means that using an unbiased estimator of the original posterior density in pseudo-marginal sampling results similar samples to the ones drawn by conventional MCMC sampling.

The advantage of conventional MCMC sampling over pseudo-marginal sampling is the acceptance rate. The conventional MCMC sampling has higher probability to

accept a candidate sample compared to the pseudo-marginal sampling. For example, in the case of 50000 training samples are available, the conventional approach accepts 926 of 1000 candidates whereas the pseudo-marginal sampling accepts 438 of 1000 candidates. The high acceptance rate of conventional MCMC sampling can be inferred from Figure 4.20 as well. In these plots, curves that correspond to the pseudo-marginal sampling have more flat regions compared to the conventional approach. The flat regions are mostly obtained when candidate samples are rejected.

Sampling from multimodal posterior densities

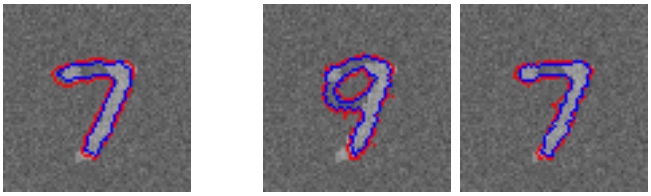
In this section, we present results of our pseudo-marginal sampling approach when sampling from multimodal posterior densities. We present results on the MNIST data set where we use the subset of the MNIST data set shown in Figure 4.17 for training. Having a multimodal prior density leads to obtain a multimodal posterior density when combined with a likelihood term in Bayesian framework. In this experiments, we draw samples from the resulting multimodal posterior density.

We apply the proposed pseudo-marginal sampling approach on the test image shown in Figure 4.18(a). We run the proposed approach three times on the test image. In each run, we constructed a chain with 1000 sampling iterations.

In the first run of the algorithm, the chain has converged at the mode of the posterior density that corresponds to digit class 7 after 140 sampling iterations. In this run, the algorithm has got stuck at this mode. Therefore, it only generated samples from digit class 7. The least and most marginal confidence boundaries obtained using all samples are shown in Figure 4.21(a).

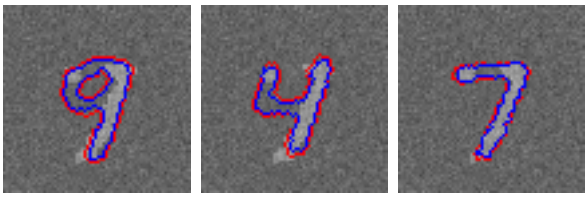
In the second run of the algorithm, the chain has converged at the mode of the posterior density that corresponds to digit class 9 after 140 sampling iterations. Until the sampling iteration 582, the algorithm draw samples from this mode. In the following iterations, the samples started to converge to the mode of digit class 7 where the convergence is achieved at iteration 634. After this point, all samples are drawn from digit class 7 until the end of the chain (sampling iteration 1000). The least and most confidence marginal boundaries obtained using samples from different modes are shown in Figure 4.21(b).

In the last run, we slightly modified our algorithm to give more freedom to



(a) The first run

(b) The second run



(c) The third run

Figure 4.21: Marginal confidence bounds obtained by samples in three different runs of the proposed approach. Note that red indicates the least confidence boundary whereas blue indicate the most confidence boundary.

discover different modes of the posterior density. To achieve this, we run the second part of our algorithm, line 4 of Algorithm 6 where we sample level set, multiple times (10 times in this experiment). This slight change does not corrupt MCMC properties and gives the algorithm the ability of moving multiple steps toward a particular mode in a single sampling iteration. In this experiment, the chain has converges to digit class 9 after 12 sampling iterations and generates samples from this class until the iteration 27. Then, the chain starts converging to mode that corresponds to digit class 4, converges at iteration 30 and generates samples of 4s until iteration 316. At iteration 317, the chain converges back to digit class 9 where if converges at iteration 320 and generates samples of 9s until the iteration 717. Lastly, the chain converges to digit class 7 at iteration 720 and generates samples from this mode until the end of the chain (the sampling iteration 1000). The least and most marginal confidence boundaries obtained using from different modes are shown in Figure 4.21(c). Note that running the level set sampling part of the algorithm for multiple times significantly increases the number of discovered modes in the posterior density. One can consider running the part where we sample class (line 3 of Algorithm 6) for multiple times for a particular application; which also does not corrupt any of the MCMC properties.

We plot the logarithm of the posterior probability of samples at each sampling iteration for three different runs (see Figure 4.22). As we mentioned above, in the first run the algorithm converges to mode of digit class 7 and generates all samples from this mode. Therefore, posterior probabilities indicated by blue curve in Figure 4.22 do not change much after convergence. The red curve shows the posterior probability of each samples obtained in the second run. Note that there are two convergence regions in the red curve. The first convergence region between iterations 140 and 582 correspond to samples from digit class 9. Note the decrease in the posterior probability after iteration 200 in this convergence region. At the iterations that correspond to decrease, the algorithm tries to converge to another mode. However, it comes back to the digit class 9. Later, the algorithm leaves digit class 9 and converges to the mode of digit class 7. The convergence can be observed in the ramp after iteration 600 in the red curve. The posterior probabilities of samples at the third run of the algorithm is shown by green curve in Figure 4.22. Note that the oscillation in the green plot is much higher than the other plots. This is because we run the level set sampling part of the algorithm multiple times in each sampling iteration. One last note from Figure 4.22 is that the mode that corresponds to digit class 7 has the higher probability than the other classes. However, even though the samples from other modes have lower probabilities they are visually reasonable segmentations given the observed data. This shows the ability of the proposed algorithm dealing with getting stuck at local optima as well as advantage of a more detailed exploration of the posterior density.

In another experiment with the MNIST data set, we use the test image shown in Figure 4.18(d). We constructed a chain with 1000 sampling iterations. We plot the logarithm of the posterior probability of samples at each sampling iteration in Figure 4.23. In this experiment, the pseudo-marginal sampling approach produces samples from digit classes 7, 4, 9, and 1. Note that samples from different classes are shown by different colors in Figure 4.23. It can be observed from the plot that the logarithm posterior probabilities of samples from digit classes 4 and 9 are very close to each other. Since samples from these classes have similar probabilities, it converging from one of these modes to another is more likely. Therefore, we see many transitions among these modes between sampling iterations 200 and 550.

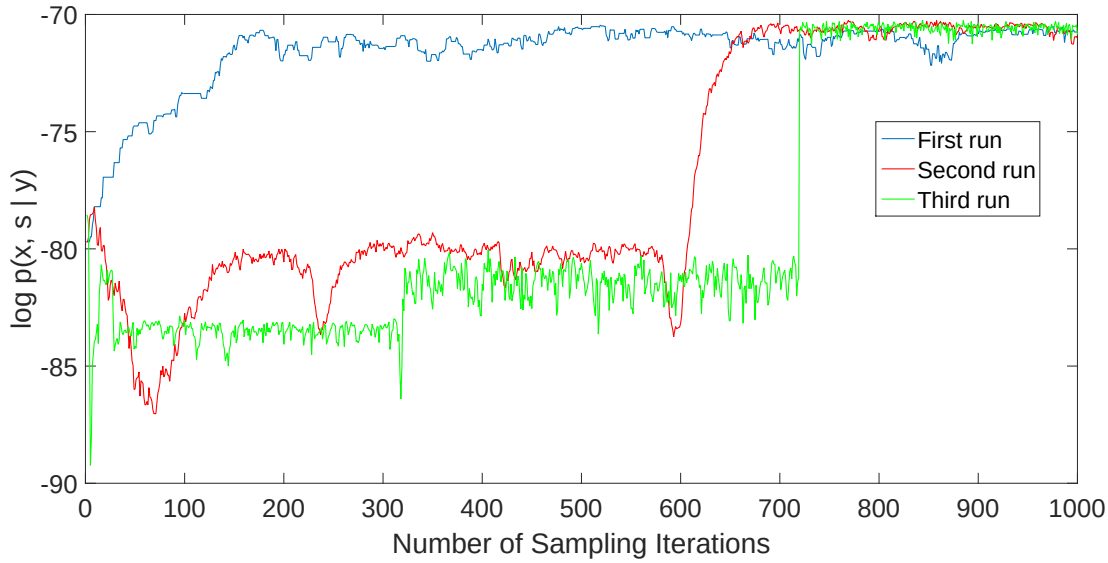


Figure 4.22: Log posterior probabilities of the samples obtained during three different runs of the algorithm on the test image in Figure 4.18(b).

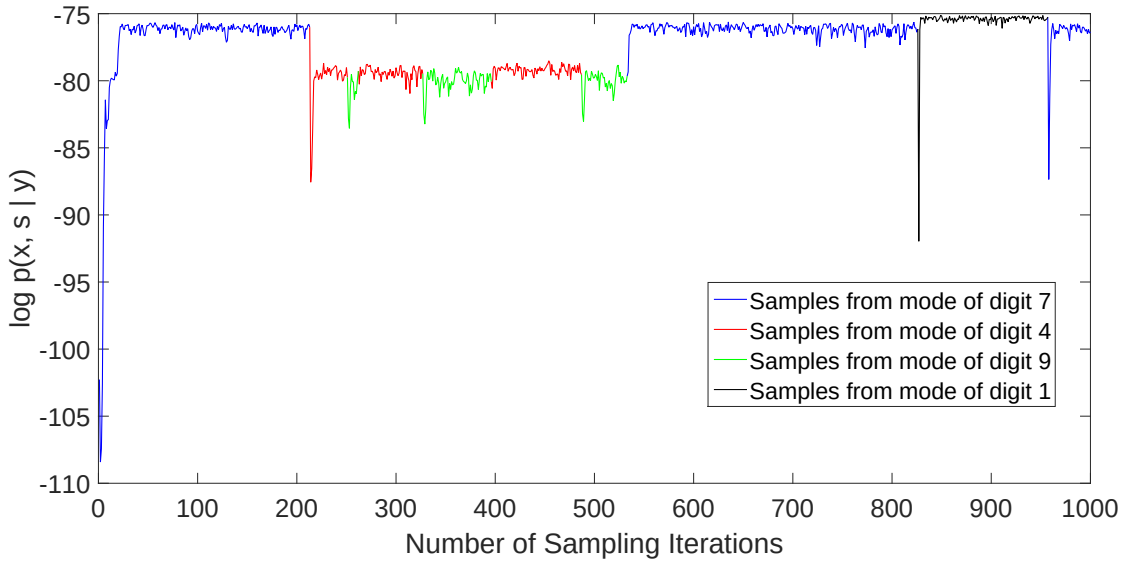


Figure 4.23: Log posterior probabilities of the samples obtained during the run of the algorithm on the test image in Figure 4.18(d).

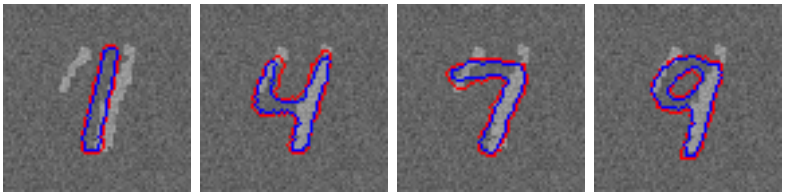


Figure 4.24: Marginal confidence bounds obtained by samples on test image shown in Figure 4.18(d). Note that red indicates the least confidence boundary whereas blue indicate the most confidence boundary.

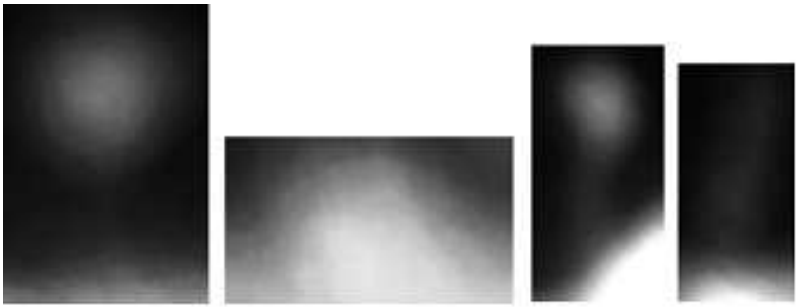
Last, we show the marginal confidence bounds of samples obtained from each digit class in Figure 4.24. It is important to note from Figure 4.23 that the samples generated from digit class 1 has the highest posterior probabilities. This is interesting because the observed data in Figure 4.18(d) do not look like a digit 1. This shows that the solutions with higher probabilities may not be the visually reasonable solutions when the data provide very limited information.

Experiments on dendritic spine data set

In this section, we present experimental results on dendritic spine data set. The data set is obtained from Neuronal Structure and Function laboratory of Champalimaud Neuroscience Foundation, Lisbon.

In the literature, dendritic spines are generally grouped into four classes: mushroom, stubby, thin, and filopodia (see Figure 4.25). The dendritic spine data set contains 88 mushroom and 27 stubby spine intensity images together with their manual segmentations. The data set does not include any thin and filopodia examples since they are barely seen spine types.

Researchers that focus on dendritic spine analysis generally identifies mushroom spines as a spine having wide head and a neck whereas stubby is defined as neckless spine [87]. However, it is very common to see spines that are in between of both definitions (see Figure 4.26). For such spines, manual labeling decisions generally change for different experts. A recent work on spine shape analysis from unsupervised learning perspective revealed clusters that include intermediate spines [88]. The uncertainty in assigning intermediate shape spines to a particular class (either mushroom or stubby) makes spine segmentation problem an interesting application



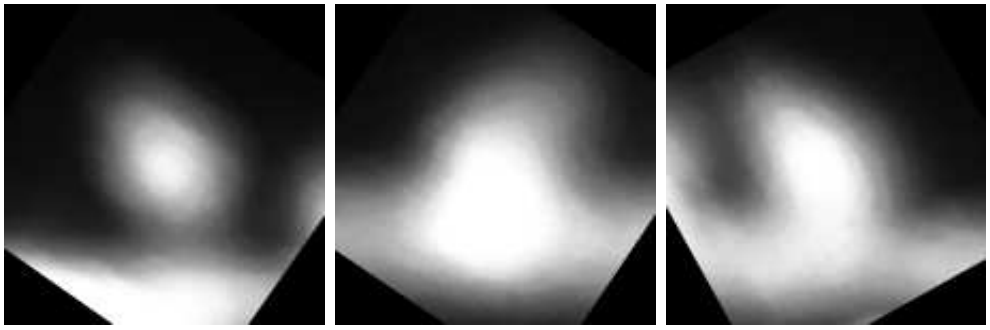
(a) Intensity images

(b) Manual Segmentations

Figure 4.25: Intensity and corresponding manually annotated binary image examples from each spine class. From left to right: Mushroom, Stubby, Thin, and Filopodia.

for a sampling-based segmentation approach.

In this experiment, we generate samples from the posterior density for segmentation of images shown in Figure 4.26. The training set we use is shown in Figure 4.27. We plot the logarithm of the posterior probability of all samples obtained for each test image (see Figure 4.28). The plot demonstrate that the samples obtained for the intermediate spine (test image shown in Figure 4.26(c)) have lower probabilities than the ones obtained for mushroom (Figure 4.26(a)) and stubby (Figure 4.26(b)) test images. This is because the shape class, s, oscillates between mushroom and stubby classes in each sampling iteration as shown in Figure 4.29. Therefore, the generated samples stuck at a low probability region between modes of mushroom and stubby. The oscillation between classes during shape class sampling reflects the uncertainty in the problem. This gives some intuition about the confidence of the results which is not possible to obtain using an optimization-based approach.



(a) (b) (c)

Figure 4.26: Visual examples of (a) mushroom, (b) stubby, and (c) intermediate spines.

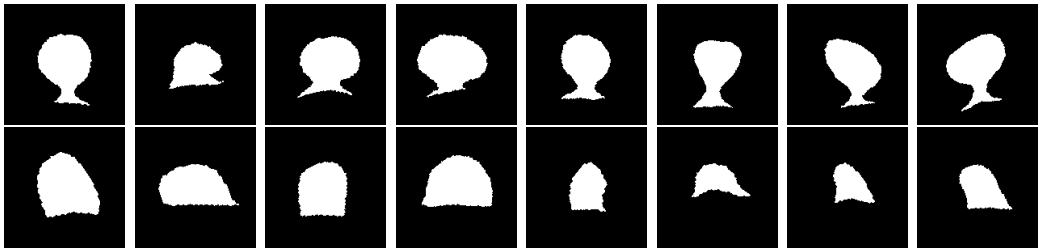


Figure 4.27: Training set for dendritic spine data set. The first row: mushroom spines, the secong row: stubby spines.

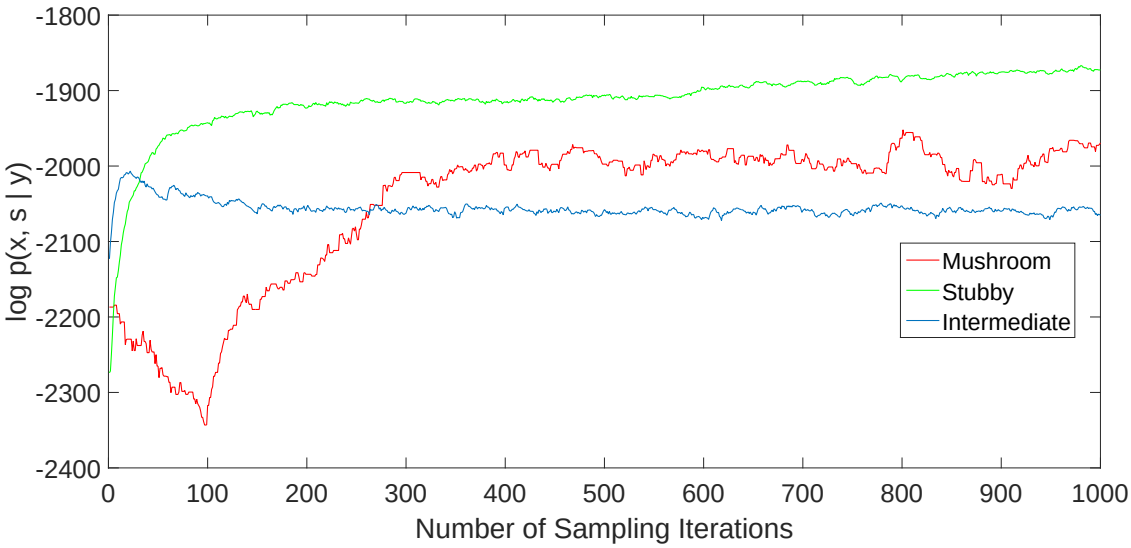


Figure 4.28: Log posterior probabilities of the samples obtained by running the algorithm on the test images in Figure 4.26.

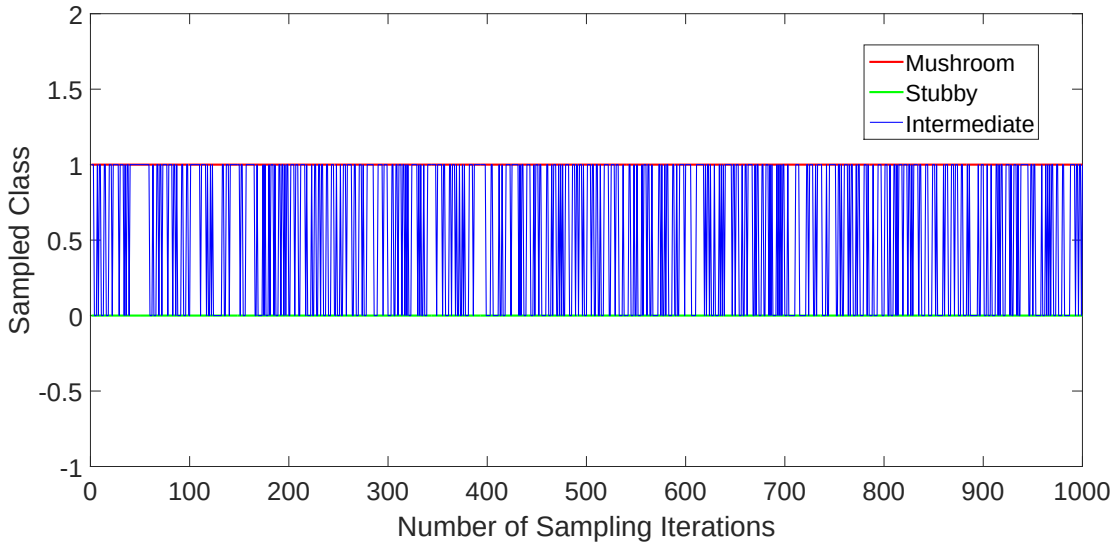


Figure 4.29: Class samples obtained by running the algorithm on the test images in Figure 4.26. Note that 0 indicates stubby and 1 indicate mushroom classes.

4.5 Conclusion

We propose a pseudo-marginal Markov chain Monte Carlo (MCMC) sampling-based image segmentation approach that exploits nonparametric shape priors. The segmentation problem is often formulated as the posterior probability of the segmenting curve given the observed data where the prior shape density is estimated using nonparametric shape priors.

The methods in the literature find the desired segmentation by performing MAP estimation on the resulting energy function. Instead, in the proposed approach, we generate samples from the resulting posterior distribution to avoid shortcomings of the optimization-based approaches. Such shortcomings include getting stuck at local optima and being unable to characterize the posterior density. The proposed MCMC sampling approach deals with all these problem while being computationally efficient unlike the conventional MCMC approaches. By using the pseudo-marginal sampling principles, the proposed approach become applicable to very large data sets; the computation time of the proposed approach does not depend on the size of the data set. Moreover, our pseudo-marginal shape sampler perfectly satisfy the necessary conditions to implement MCMC sampling which is very crucial ensuring the generated samples come from the desired distribution. Existing methods in the

literature only approximately satisfies these conditions.

We perform experimental results on various data sets to show the advantages of the proposed pseudo-marginal shape sampling approach. We believe performance of many segmentation tasks can be improved by exploring the posterior density in more detail, especially on the applications where uncertainty on the results matter.

Chapter 5

Disjunctive Normal Shape Boltzmann Machine

Shape Boltzmann machine (a type of Deep Boltzmann machine) is a powerful tool for shape modelling; however, has some drawbacks in representation of local shape parts. Disjunctive Normal Shape Model (DNSM) is a strong shape model that can effectively represent local parts of objects. In this work, we propose a new shape model based on Shape Boltzmann Machine and Disjunctive Normal Shape Model which we call Disjunctive Normal Shape Boltzmann Machine (DNSBM). DNSBM learns binary distributions of shapes by taking both local and global shape constraints into account using a type of Deep Boltzmann Machine. The samples generated using DNSBM look realistic. Moreover, DNSBM is capable of generating novel samples that differ from training examples by exploiting the local shape representation capability of DNSM. We demonstrate the performance of DNSBM for shape completion on two different data sets in which exploitation of local shape parts is important for capturing the statistical variability of the underlying shape distributions. Experimental results show that DNSBM is a strong model for representing shapes that are composed of local parts.

5.1 Related work

A strong shape model should contain two important properties: realism and generalization [22]. The first property states that the model should capture the correct shape distributions, i.e., samples that are drawn from the distribution should be valid shapes. The second constraint ensures that the samples generated from the learned distribution should also cover unseen but valid shapes.

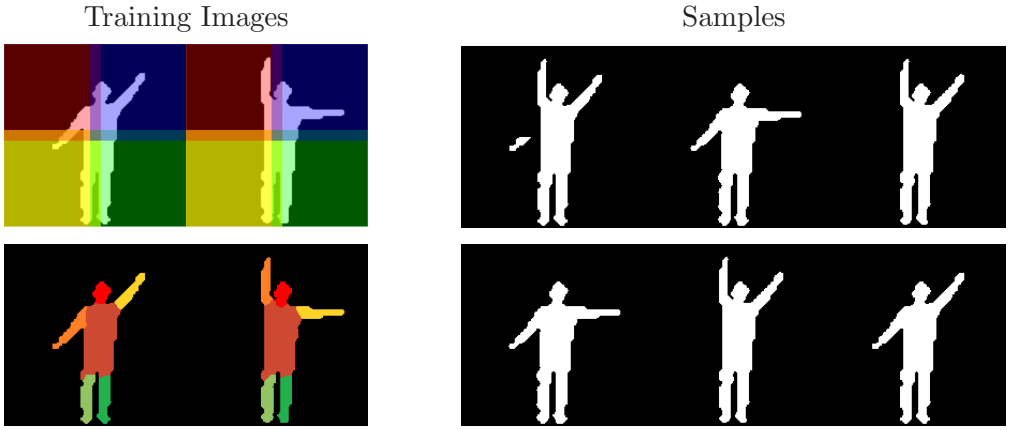


Figure 5.1: Local shape representation and shape sampling using SBM (first row) and the proposed DNSBM (second row).

There exist a variety of approaches for $2D$ shape modelling in the literature [89] [90] [91] [45] [23].

Restricted Boltzmann Machine (RBM) [92] is a model that includes a number of hidden variables $\mathbf{h}$ each connected to all image pixels (units in the visible layer $\mathbf{v}$) as shown in Figure 5.3(a). Note that there are no direct connections between the units of a layer, which makes this a bipartite graph. Hence, the energy of a configuration can be written as follows:

$$E(\mathbf{v}, \mathbf{h}) = \sum_i b_i v_i + \sum_{i,j} w_{ij} v_i h_j + \sum_j c_j h_j \tag{5.1}$$

where, i and j range over pixels and hidden variables, respectively. Then, the model can learn constraints and dependencies between pixels by learning the parameters w_{ij}, b_i, and c_j. The distribution over $\mathbf{v}$ is given by marginalizing over the hidden variables: $p(\mathbf{v}) = \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}))/Z(\theta)$, where θ represents the model parameters and $Z(\theta)$ is the partition function. This marginalization allows the model to capture dependencies between the image pixels. RBM has edges between hidden and visible variables. Therefore, all hidden units are conditionally independent given the visible units. Similarly, all visible units are conditionally independent given the hidden units. This property is useful for exact and efficient inference. Then, the conditional probabilities can be written as $p(v_i = 1|\mathbf{h}) = \sigma(\sum_j w_{ij} h_j + b_i)$ and $p(h_j = 1|\mathbf{v}) = \sigma(\sum_i w_{ij} v_i + c_j)$ where, $\sigma(\circ) = 1/(1 + \exp(-\circ))$ is the sigmoid

function. Using this property, $\mathbf{v}$ and $\mathbf{h}$ can be sampled consecutively, which can be exploited during learning the model parameters [93].

RBMs can approximate any binary distribution if an exponential number of hidden units and a large amount of training data are available [92]. The DBM is capable of learning more complex structures in the data using additional hidden units as shown in Fig. 5.3(b). The energy of a DBM with two hidden layers can be written as follows:

$$E(\mathbf{v}, \mathbf{h^1}, \mathbf{h^2}) = \sum_i b_i v_i + \sum_{i,j} w^1_{ij} v_i h^1_j + \sum_j c^1_j h^1_j + \sum_{j,k} w^2_{jk} h^1_j h^2_k + \sum_k c^2_k h^2_k \quad (5.2)$$

where, i, j, and k range over pixels, the first, and the second hidden variables, respectively. Exact inference is no longer possible in this model, however, the conditional distributions $p(\mathbf{v}|\mathbf{h^1})$, $p(\mathbf{h^1}|\mathbf{v}, \mathbf{h^2})$ and $p(\mathbf{h^2}|\mathbf{h^1})$ can be computed as in RBMs [94]. Then, computationally efficient approximate inference can be established by block-Gibbs sampling from the posterior $p(\mathbf{h^1}, \mathbf{h^2}|\mathbf{v})$ [22].

RBM and DBM are powerful models, however, they require a large number of binary images to learn the shape distributions like the other recent and powerful generative models: Generative Adversarial Network (GAN) [95] and Variational Autoencoders (VAE) [96]. In most applications, sizes of the available data sets are small since obtaining segmented binary images is an expensive process. SBM [22] is a shape model based on RBM and DBM that accurately captures the properties of binary shapes. Unlike RBM and DBM, SBM is capable of learning shape distributions even when the size of the training set is limited, by exploiting information from local shape representations. The visible units $\mathbf{v}$ of the SBM are the pixels of an $X \times Y$ binary image. SBM divides images into four equal-sized slightly overlapping patches to represent local shape parts as shown in Fig. 5.1. The first hidden layer $\mathbf{h^1}$ consists of four blocks and each block is fully connected to a particular patch. Finally, all units in $\mathbf{h^1}$ are fully connected to the units in the second hidden layer $\mathbf{h^2}$. The structure of SBM for $1D$ images is shown in Fig. 5.3(c). The structure can easily be generalized to $2D$. SBM follows the procedure in [94] to learn the model parameters and generates a new sample using block-Gibbs sampling as depicted in Figure 5.2.

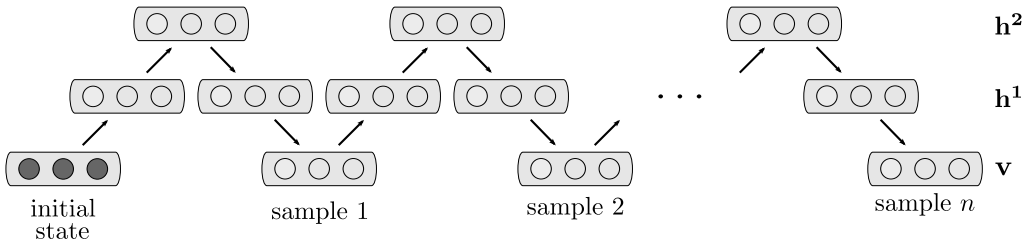


Figure 5.2: Block-Gibbs Sampling.

Recently, Erdil et al. [58] proposed a Markov chain Monte Carlo method for generating samples from shape posterior densities. Since the method represents local shape parts with patches as in SBM, it suffers from similar issues when generating a new sample.

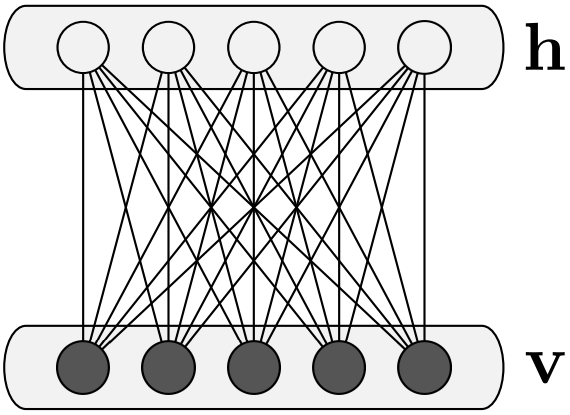
5.2 Motivation

In SBM, the patches that represent local shape parts do not correspond to a geometrically meaningful local shape parts. Here, a geometrically meaningful local shape part stands for a single physical component of the shape, for example, a particular limb (e.g., head, arm, etc.) of the standing person shown in Figure 5.1. In patch-based local shape representation, a geometrically meaningful local shape part can appear in multiple patches. For example, the left arm of the standing person shown in the first row of Figure 5.1 is contained partially in both red and yellow local regions in the first training image. Therefore, samples generated by SBM may contain unrealistic samples. For example, the sample in the third column of the first row in Figure 5.1 contains two left arms; one is raised up and the other partially appears just to the left of the body. This motivates us to develop a new shape model that exploits a better local shape part representation to learn the underlying shape distribution.

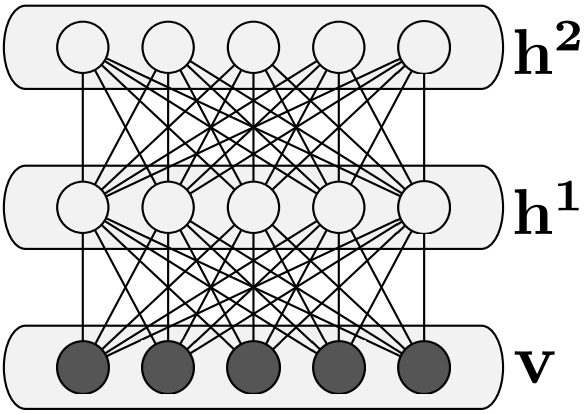
5.3 Contributions

Our contribution in this work is a new shape model called Disjunctive Normal Shape Boltzmann Machine (DNSBM) which exploits the property of SBM for learning complex binary distributions and the property of DNSM [19] for representing

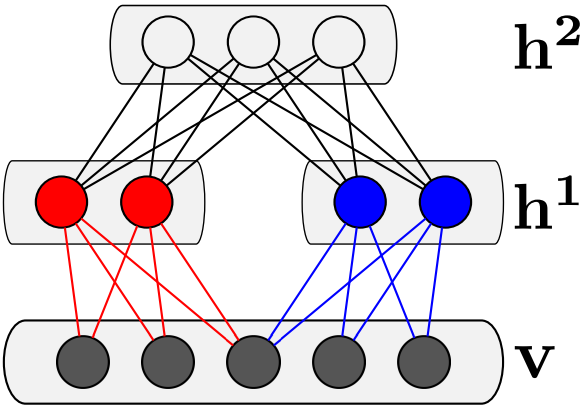
local parts of shapes. DNSM is an implicit and parametric model that represents a shape by a union of convex polytopes. In DNSM, each polytope or union of a subset of the polytopes can represent a physical local part of an object as shown in the second row of Fig. 5.1. This property of DNSM makes it a very powerful model for representing local shape parts. As we exploit that property, samples generated by our proposed DNSBM are realistic. Also, DNSBM is able to generate novel samples which are not contained in the training set by exploiting local shape parts in block-Gibbs sampling and by using the learned distribution. We train DNSBM on two different data sets in which local shape parts are important for capturing the statistical variability of the whole shape distribution and show its performance by generating samples from the distribution for shape completion. Experimental results show the effectiveness of DNSBM. Some exemplary results of DNSBM using two training examples are shown in the second row of Fig. 5.1. Here, our approach is able to generate realistic and novel samples that are not contained in the training set.



(a) RBM

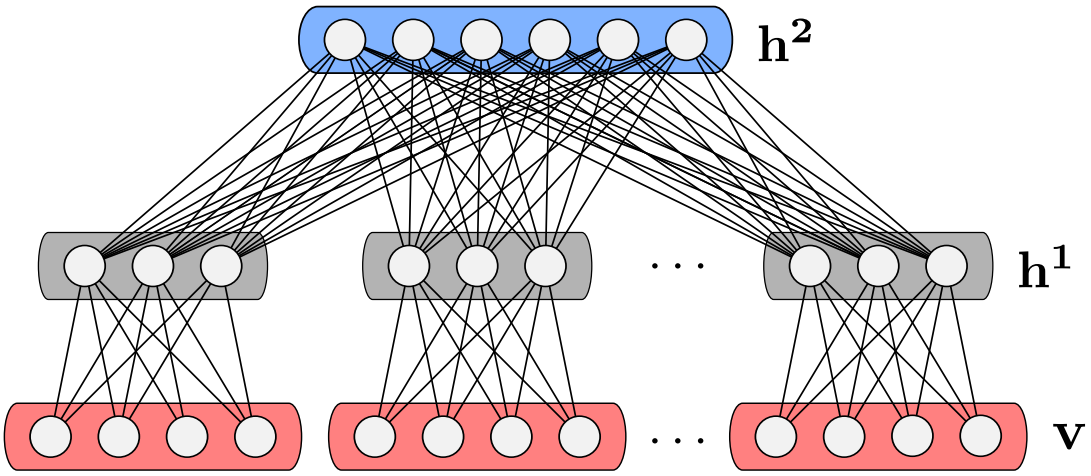


(b) DBM



(c) SBM

Figure 5.3: Undirected models for modelling binary shapes.



(a) DNSBM

Figure 5.3 (cont.): Undirected models for modelling binary shapes.

5.4 The proposed method

In this section, we provide the formulation of DNSM [19] to represent binary shapes in terms of physically meaningful local shape parts. Then, we introduce the proposed shape model: Disjunctive Normal Shape Boltzmann Machine (DNSBM).

5.4.1 Binary shape representation using DNSM

DNSM represents a shape by a union of convex polytopes. A polytope can be represented by intersection of half-spaces as shown in Fig. 5.4(b). Smooth convex polyopes can be obtained by increasing number of half-spaces (see Fig. 5.4(c)).

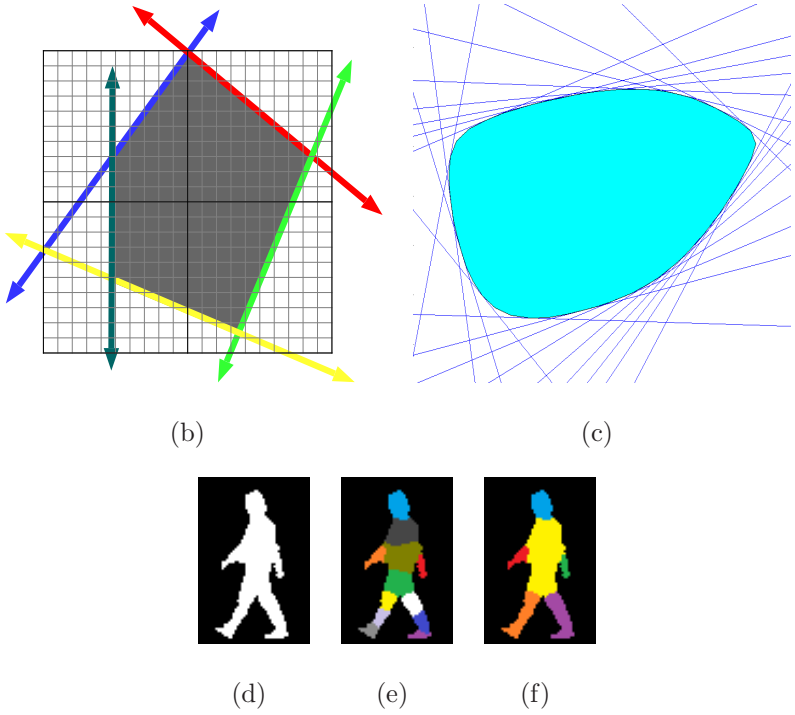


Figure 5.4: DNSM shape representation.

DNSM approximates the characteristic function of a shape as a union of convex polytopes which themselves are represented as intersections of half-spaces. Consider the characteristic function of a D-dimensional shape $f : \mathbf{R}^D \rightarrow B$ where $B = \{0, 1\}$. Let $\Omega^+ = \{\mathbf{x} \in \mathbf{R}^D : f(\mathbf{x}) = 1\}$ represent the foreground region. Ω^+ can be approximated as a union of N convex polytopes, $\Omega^+ \approx \bigcup_{i=1}^{N} P_i$. The i^{th} polytope is defined as the intersection $P_i = \bigcap_{j=1}^{M_i} H_{ij}$ of M_i half-spaces. The half-spaces are defined as $H_{ij} = \{\mathbf{x} \in \mathbf{R}^D : h_{ij}(\mathbf{x})\}$ where

$$h_{ij}(\mathbf{x}) = \begin{cases} 1, & \text{if } \sum_{k=1}^{D} \delta_{ijk} x_k + c_{ij} \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Therefore, Ω^+ is approximated by $\bigcup_{i=1}^{N} \bigcap_{j=1}^{M_i} H_{ij}$ and equivalently $f(\mathbf{x})$ is approximated by the disjunctive normal form $\bigvee_{i=1}^{N} \bigwedge_{j=1}^{M_i} h_{ij}(\mathbf{x})$. Converting the disjunctive normal form to a differentiable shape representation requires the following steps: First, De Morgan's rules are used to replace the disjunction with negations and conjunctions, which yields $f(\mathbf{x}) \approx \bigvee_{i=1}^{N} \bigwedge_{j=1}^{M_i} h_{ij}(\mathbf{x}) = \neg \bigwedge_{i=1}^{N} \neg \bigwedge_{j=1}^{M_i} h_{ij}(\mathbf{x})$. Since conjunctions of binary functions are equivalent to their product and negation is equivalent to subtraction from 1, $f(\mathbf{x})$ can also be approximated as $1 -$

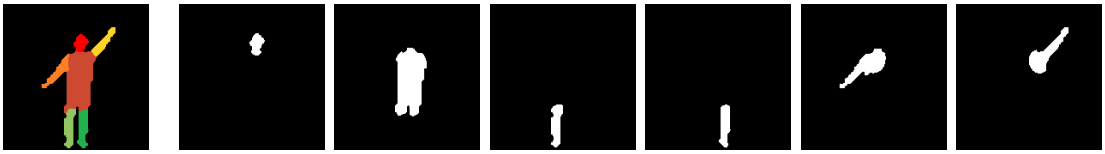
$\prod_{i=1}^{N}(1-\prod_{j=1}^{M_i} h_{ij}(\mathbf{x}))$. The final step for obtaining a differentiable representation is to relax the discriminants h_{ij} to sigmoid functions $\sigma_{ij} = 1/(1+e^{-(\sum_{k=1}^{D}\delta_{ijk}x_k+c_{ij})})$. The resulting approximation to the shape characteristic functions is then given by $f(\mathbf{x}) = 1-\prod_{i=1}^{N}\left(1-\prod_{j=1}^{M_i}\sigma_{ij}\right)$, where $\mathbf{x}=\{x,y\}$ for two-dimensional ($2D$) shapes and $\mathbf{x}=\{x,y,z\}$ for three-dimensional ($3D$) shapes [19].

The only free parameters of the model are δ_{ijk} and c_{ij}, which determine the orientation and location of the sigmoid functions (discriminants) that define the half-spaces. The level set $f(x)=0.5$ is taken to represent the interface between the foreground ($f(\mathbf{x})\geq 0.5$) and background ($f(\mathbf{x})<0.5$) regions.

The DNSM discriminant parameters, $\mathbf{\Delta}^t$, that represent the t^{th} training sample can be obtained by choosing the weights that minimize the following energy function

$$E(\mathbf{\Delta}^t) = \int_{\mathbf{x}\in\Omega}(f(\mathbf{x})-q_t(\mathbf{x}))^2 d\mathbf{x} + \eta\sum_{i}^{N}\sum_{r\neq i}^{N}\int_{\mathbf{x}\in\Omega} g_i(\mathbf{x})g_r(\mathbf{x})d\mathbf{x} \tag{5.3}$$

where, $g_i(\mathbf{x}) = \prod_{j=1}^{M_i}\sigma_{ij}$ represents the individual polytopes of $f(\mathbf{x})$. $q_t(\mathbf{x})$ is the t^{th} binary training image (1 for object and 0 for background) to be represented by DNSM and η is a constant that controls the allowed degree of overlap between polytopes. We find that having slightly overlapping polytopes is important to ensure shape continuity in the generated samples by DNSBM. We minimize Equation (5.3) using gradient descent to obtain $\mathbf{\Delta}^t$ which represents the t^{th} training sample. DNSM representation of the binary image in Fig. 5.4(d) is given in Fig. 5.4(e). Note that each polytope may not correspond to a local geometrically meaningful shape part since large number of convex polytopes are required for representing complex shapes. One can consider combining polytopes manually to obtain local shape parts when constructing the training set. We use the approach proposed in [97] that relaxes the convexity constraint of DNSM and represents complex shapes by a smaller number of approximately convex polytopes each corresponding to a geometrically meaningful local shape part. Fig. 5.4(f) shows the approximately convex polytopes obtained using the approach in [97].



(a) (b)

Figure 5.5: Decomposing a shape into polytopes. (a) A shape with DNSM representation. (b) Binary images corresponding to each physical shape part (polytope).

5.4.2 From DNSM to DNSBM

Our proposed approach, DNSBM is a type of Deep Boltzmann Machine having the structure shown in Fig. 5.4(a). In DNSBM, each pre-aligned binary training shape in an $X \times Y$ image is initially represented with N polytopes such that each polytope corresponds to a physically meaningful (local) shape part as explained in Section 5.4.1. Then, each shape is decomposed into N binary images where each binary image represents a single local shape part as shown in Fig. 5.5. Each red block in the visible layer $\mathbf{v}$ of DNSBM (see Fig. 5.4(a)) corresponds to a binary image that represents a particular local shape part. Therefore, there are N red blocks each containing $X \times Y$ units in the visible layer of DNSBM as exemplified by the binary images in Fig. 5.5(b). The first hidden layer $\mathbf{h^1}$ of DNSBM is composed of N blocks (shown in gray in Fig. 5.4(a)). The units in each block of $\mathbf{v}$ are fully connected with the units in the corresponding block of $\mathbf{h^1}$. Each unit of $\mathbf{h^1}$ is also connected to all units of $\mathbf{h^2}$. While the connections between $\mathbf{v}$ and $\mathbf{h^1}$ capture the dependencies between pixels, the connections between $\mathbf{h^1}$ and $\mathbf{h^2}$ capture the inter-relations of local shape parts.

Learning of the model involves maximizing $\log p(\mathbf{v};\theta)$ of the observed data $\mathbf{v}$ with respect to its parameters $\theta = \{b, W_1, W_2, c_1, c_2\}$. The work in [94] proposes a two-phase learning procedure. In the pre-training, the model is trained bottom up, one layer at a time, to find a good initial estimates of the model parameters. Once the parameters are initialized, parameters of the full model can be fine-tuned by backpropagation. In DNSBM, we follow the same procedure in [94] to learn the model parameters. Once the parameters of DNSBM are found, we generate samples from the model using block-Gibbs sampling.

5.5 Experimental results

In this section, we present experimental results of DNSBM on two data sets in which local shape parts play an important role for identifying shape densities when the training set is limited. We compare the performance of the DNSBM with SBM. The implementation of DNSBM and the data sets are available at `github.com/eerdil/dnsbm_icassp17`.

The first data set is the standing person data set [83]. The data set contains 50, 170×170 binary images of a standing person with varying arm postures. We construct a training set with 28 images by using shapes each containing a particular posture of either left or right arm as shown in Fig. 5.6. Each of the remaining 22 shapes in the data set contains arm postures of both left and right arms. Since each arm posture is contained for both left and right arms separately in the training set, the remaining 22 shapes can be explored by exploiting these local shape parts. Note that, exploitation of local shape parts is not done simply by combining all possible local shapes, it naturally emerges as a result of block-Gibbs sampling. We obtain local shape (head, left arm, right arm, etc.) representations of the standing person for each binary training shape using DNSM. When training DNSBM on this data set, we empirically set sizes of $\mathbf{h^1}$ and $\mathbf{h^2}$ to 2000 and 500, respectively. Increasing the size of $\mathbf{h^2}$ may cause overfitting whereas $\mathbf{h^1}$ should be large enough to capture pixel dependencies.

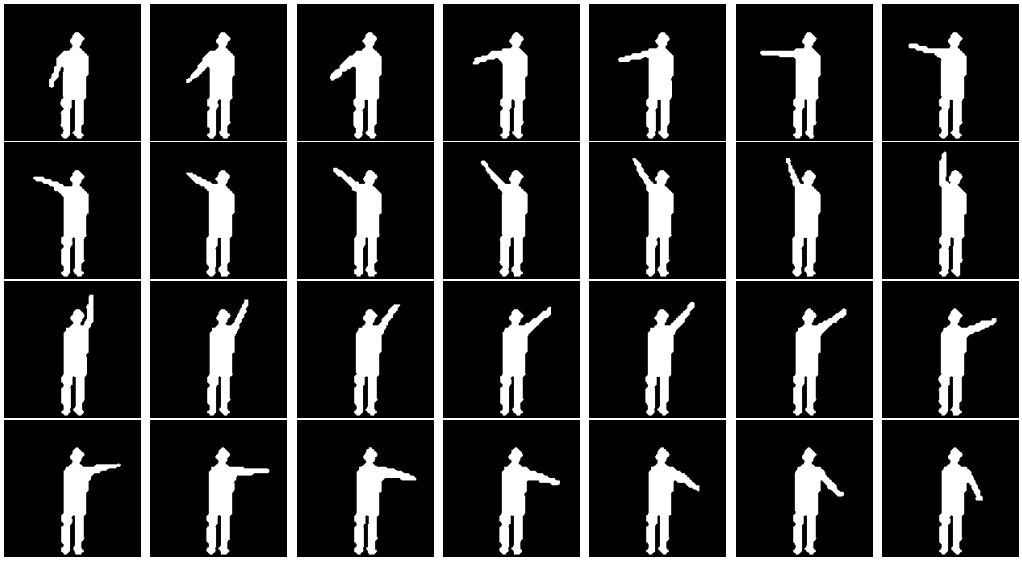


Figure 5.6: Training set of the standing person data set.

We design 3 test cases having different missing regions to be completed in our experiments as shown in the first column of Fig. 5.7. Image completion is established by generating samples from both DNSBM and SBM using the observed part of the shape. Some shape completion results of each approach are shown in Fig. 5.7. We also provide likelihood images in the first column for each approach in Fig. 5.7. These images are obtained by summing up all generated samples and normalizing with the total number of samples [77]. We further enhance the likelihood images in Fig. 5.7 for visualization purposes. Note that in the likelihood images, bright pixels indicate high occurrence of the corresponding pixel in foreground region of the generated samples. In this data set, all samples of DNSBM appear realistic, i.e., there is no sample that does not look like a standing person, whereas SBM generates some unrealistic samples (see the standing person samples in Fig. 5.10(b)).

The second data set is the walking silhouette data set [17]. The walking silhouette data set contains 150 binary images of a walking person. Similar to the experiments on the standing person data set, we choose a subset of 24 images (see Fig. 5.8) for training. We obtain the local shape parts of walking silhouettes using 6 polytopes with DNSM. We train the DNSBM on this data set using 1000 units for $\mathbf{h^1}$ and 50 units for $\mathbf{h^2}$ for 78×52 images.

Figure 5.8: Training set of the walking silhouette data set.

We design 5 test cases for shape completion using shapes not included in the training set and with different missing regions to be completed as shown in the first column of Fig. 5.9. We perform shape completion on these test images by generating samples from both DNSBM and SBM. Some completion results of each method together with the likelihood images for the corresponding input shape are shown in Fig. 5.9. The walking silhouette data set is a more challenging data set than the previous one since it contains more local shape parts that change their posture. In this data set, DNSBM produces better results than the SBM in terms of

the number of realistic samples, as well as its generalization capability to generate valid and diverse shapes, as shown particularly in the 2^{nd}, 3^{rd}, and 5^{th} rows of Fig. 5.9. Some unrealistic samples generated by both DNSBM and SBM on the walking silhouette data set are given in Fig. 5.10. The patch-based local shape representation of SBM is not a good representation for this data set, since almost each physical shape part, especially legs of the silhouette, appears in more than one patch. This leads SBM to generate a large number of unrealistic samples in this data set.

Figure 5.9: Samples generated by DNSBM and SBM for completion of the shapes in the first column. Pixels in the red region are missing.

(a) DNSBM (b) SBM

Figure 5.10: Some unrealistic samples generated by DNSBM and SBM.

Table 5.1: Comparison of DNSBM and SBM using Dice score.

	DNSBM	SBM
Walking silhouette	**0.6526**	0.6112
Standing Person	**0.5935**	0.5825

Quantitative evaluation of sampling-based approaches is not a trivial task and requires considering different metrics. First, we compute the similarity between the ground truth and the completion results using Dice score [70], since it is expected that a sampling-based approach generates many samples that are similar to the ground truth. The average Dice score results of all test cases for both data sets are shown in Table 5.1. Note that, high values of Dice score indicate higher similarity with the ground truth. Second, we expect to obtain realistic samples. We measure this by computing the probability of sampling the completed region given the observed data using the imputation score [22]. The average of all imputation scores in all test cases of both data sets are 0.085 for DNSBM and 0.014 for SBM where higher is better. Finally, a good sampling approach is expected to generate diverse samples. We demonstrate the diversity of samples by plotting the precision-recall (PR) values of all samples generated in all test cases in the walking silhouette data set as shown in Fig. 5.11. The results demonstrate that the samples of DNSBM spread in the precision-recall space more than the samples of SBM. Note that a large number of blue crosses in Fig. 5.11 correspond to unrealistic samples produced by SBM. Therefore, the superiority of the DNSBM over SBM in terms of diversity becomes more evident if we consider Fig. 5.11 without such samples.

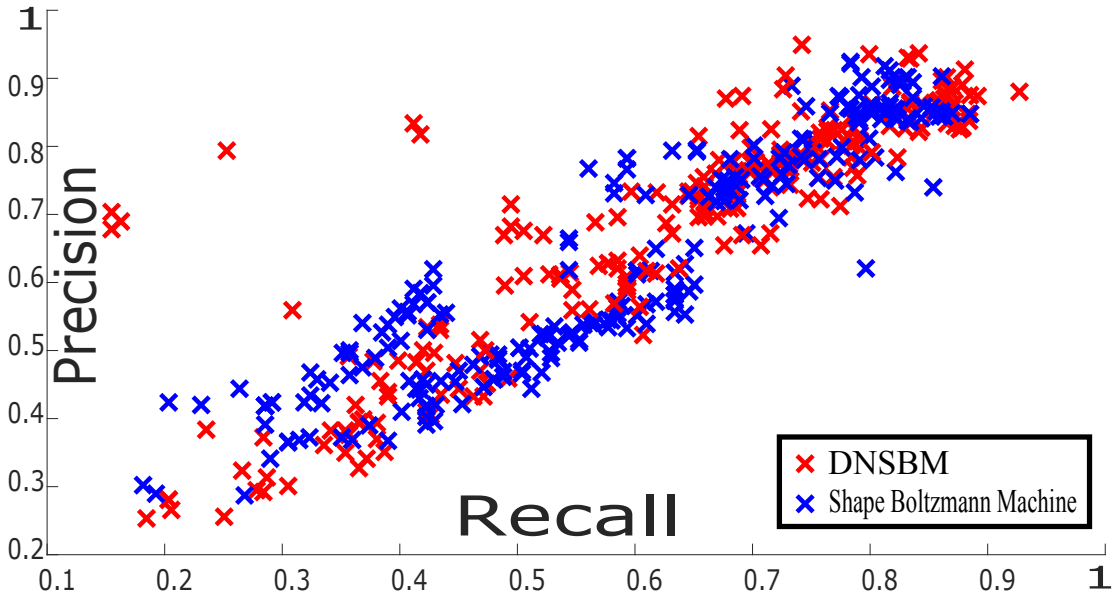


Figure 5.11: PR values of the samples generated using the walking silhouette data set.

Since DNSBM uses a representation of each physical local part individually by a single polytope, it does not suffer from having multiple pieces for a single local part in the generated samples. However, in some cases, exploiting different local shape parts in the training set does not yield a visually appealing sample as shown in Fig. 5.10. This problem originates at places where local shape parts are connected to each other. Although we have solved this problem up to some level by generating overlapping polytopes, we can still encounter such samples in some rare cases. Some possible solutions of this problem might be incorporating information about tie locations of polytopes to the sampling process. One can also consider performing a local registration as a post-processing step.

5.6 Conclusion

We have presented a shape model, DNSBM, that is based on the SBM and the DNSM. DNSBM is able to represent physically meaningful local shape parts individually and exploits this representation when the training set size is limited. We have shown the performance of DNSBM on two data sets for shape completion. The proposed method exhibits better performance than SBM.

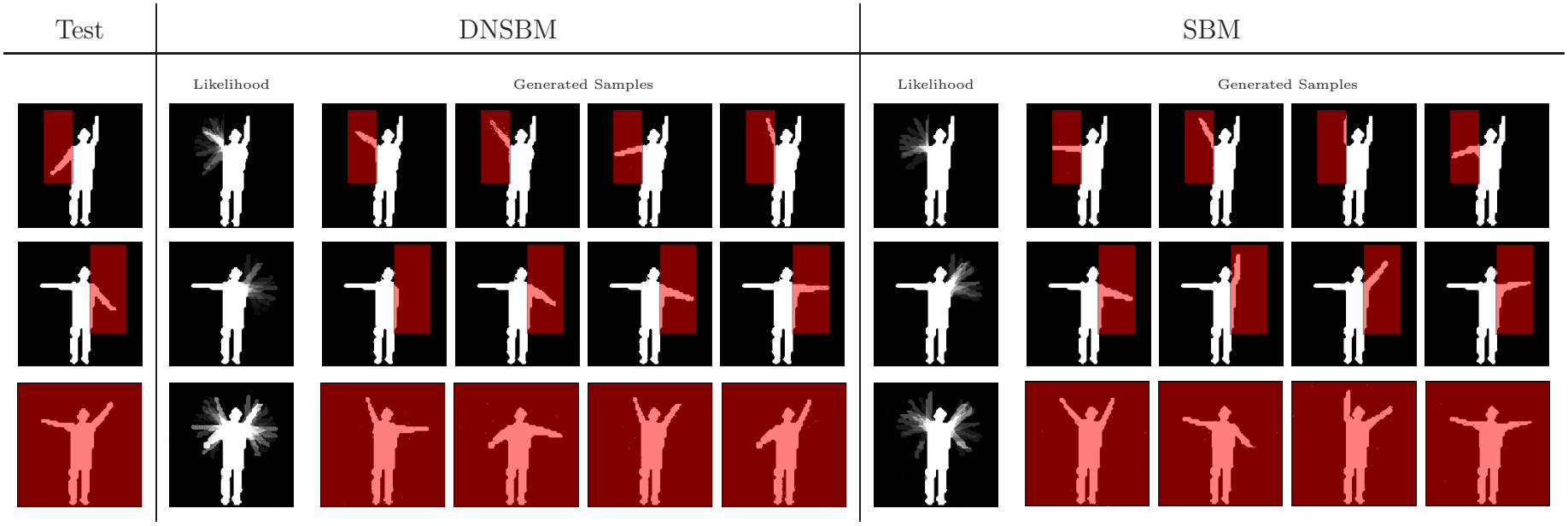


Figure 5.7: Samples generated by DNSBM and SBM for completion of the shapes in the first column. Pixels in the red region are missing.

Chapter 6

Conclusion and future work

In this chapter, we provide a summary of this thesis and possible future research directions.

6.1 Summary of this thesis

In this thesis, we propose Bayesian methods for image segmentation that exploits nonparametric shape priors. We approach the problem from two different perspectives: (1) MAP estimation, (2) Markov chain Monte Carlo sampling.

First, we propose is a segmentation method that exploits joint nonparametric shape and feature priors. The proposed method represents the segmentation problem in terms of the joint posterior density of shapes and features. Then, the resulting energy function is minimized using gradient descent and active contours. The role of using feature priors is to aid the evolving contour to converge to the correct mode of the posterior density. Conditioning on the extracted feature information helps improve the segmentation performance. Experimental results demonstrate the effectiveness of the proposed method when the posterior density is multimodal and data provides very limited information about the object to be segmented.

Second, we propose two different Markov chain Monte Carlo sampling based segmentation methods that exploits nonparametric shape priors. The first approach represents the segmentation problem in terms of the posterior probability density of the object boundaries given the observed data. Once the method finds the boundaries that can be segmented using a data fidelity term, it gives a probabilistic class decision about the object to be segmented. Then, samples from each selected class

are generated using Metropolis-Hastings. In order to generate samples from different modes, it is required to create multiple Markov chains each expected to generate samples from different modes. This brings a very high amount of computational cost to the approach which makes it really hard to use with large training sets. Moreover, this approach computes some probabilities in the Metropolis-Hastings ratio approximately. Although practical results demonstrate that the algorithm generates meaningful samples, it is not guaranteed to have samples from the target distribution as long as we do not compute these probabilities exactly. These shortcomings of the first MCMC sampling based approach motivate us to develop the second approach. In the second approach, we propose a pseudo-marginal MCMC sampling approach that uses nonparametric shape priors. The proposed approach represents the target posterior distribution as the joint distribution of segmenting curves and classes. Unlike the first MCMC sampling based approach, the second approach creates a single Markov chain that generates samples from different modes of the posterior density. The computation cost of the proposed approach does not depend on the size of the training set thanks to pseudo-marginal sampling. Moreover, in the proposed approach, probabilities are computed exactly when evaluating the Metropolis-Hastings ratio which guarantees having samples from the desired distribution.

Last, we propose a shape model that learns binary shape distributions called Disjunctive Normal Shape Boltzmann Machine (DNSBM). DNSBM combines the power of Shape Boltzmann Machine on learning shape distributions and Disjunctive Normal Shape Model on representing local shape parts. DNSBM can generate novel and realistic samples from the learned distribution. DNSBM has potential to be used in various applications such as image segmentation which is the main focus of this thesis.

6.2 Future research directions

Cremers et al. [62] and Chan et al. [63] focus on the segmentation problem in which given a scene with multiple different types of objects, the problem is to segment a particular object that is included in the training set. However, the approach

may not be effective when the training set contains object that are similar to each other and/or data provide very limited information. These shortcomings are similar to the ones that we mentioned in Chapter 3 for the methods proposed in the existing nonparametric shape priors based methods such the ones proposed by Kim et al. [1] and Cremers et al. [17]. Therefore, the shortcomings of the methods in [62] and [63] can possible be addressed by using additional feature priors as we proposed in Chapter 3. Hence using our joint shape and feature prior-based approach in the kinds of segmentation problems posed in [62, 63] could be explored in the future.

Mesadi et al. [98] propose a method that uses local shape and appearance priors for object segmentation using disjunctive normal shape models [23]. Using local shape and appearance priors significantly improves the segmentation results because it allows learning shape and appearance distributions in local regions. One can consider developing a segmentation approach by exploiting local shape priors in level set representation. In this representation, local regions can be represented by grids or an arbitrary decomposition of shapes depending on the application. Using a local shape representation with level sets might improve the performance of the proposed approaches in this thesis since it helps to extract more information from limited data and training set.

As we mentioned several times throughout this thesis, a simple Bayesian formulation of the segmentation problem consists of two terms: prior distribution of shape and conditional distribution of data given shape. In this thesis, we mostly focused on designing the prior distribution of shapes and used very simple models for the data term. One powerful approach to estimate the conditional density of data given shapes is to use intensity priors from a training set [57]. The learned conditional density of data given shapes can be used in the approaches proposed in Chapters 3 and 4 to achieve better results in some applications.

Convolutional Neural Networks (CNNs) have become very popular in recent years and have been successfully applied to many different problems. The main power of CNNs comes from the ability of learning their own features for different types of objects. In CNNs, each convolutional layer carries different types of features for objects from different classes. Adapting these features into the active contour models can be an interesting future research direction which can possibly lead to

better segmentation performance.

Using shape priors in the segmentation process helps in completing boundaries in the regions with missing data and/or occlusions. One can expects to recover the intensities in a region that suffers from missing data and/or occlusion. Therefore, it can be interesting to represent the problem as a joint segmentation and inpainting problem.

DNSBM is an interesting shape model for learning binary shape distributions by exploiting local shape parts. The learned distribution can be used in the segmentation process. A segmentation approach that uses the learned shape distribution learned by deep Boltzmann machine can be found in [83]. As an extension of the work of this thesis, development of a segmentation approach utilizing a DNSBM-based shape model could be an interesting direction for future work.

Most of the segmentation problems considered in this thesis involved multi-modal shape densities. The modes or other well-defined components of such shape densities could be associated with classes of objects included in the space of objects of interest. Our primary objective in this thesis was to solve segmentation problems, where the classes or multi-modal nature of the densities appeared as complicating factors. However, one can of course be explicitly interested in inferring the class of the object in the scene as well as segmenting it. The machinery we have developed in Chapters 3 and 4 of this thesis could serve as the basis of a statistical framework for joint segmentation and recognition of objects. Posing such a problem and examining the tools developed in this thesis in the process of solving that problem could be another interesting direction for future work.

Given recent success of deep learning methods in a variety of image analysis problems, a major line of future research might involve establishing connections between the types of methods presented in this thesis and deep learning methods. In fact, a limited portion of our work, that presented in Chapter 5, already contains such a connection, through Deep Boltzmann Machines. However, all of the work in the earlier chapters fits within the more classical statistical framework of Bayesian methods. A natural question that might arise is the following: should we still be interested in such classical methods, given all the success of powerful deep learning methods? We argue the answer is yes, and the key is to observe how these differ-

ent lines of thought can be consolidated in a complementary way to produce even more powerful methods. While deep learning methods have revolutionized machine learning, one important limitation has been that most deep learning models cannot represent their uncertainty. Representation of uncertainty has been one of the main themes of this thesis. Interest in combining Bayesian approaches with deep learning methods has recently emerged (although there exist examples in earlier work as well), as exemplified by the Bayesian Deep Learning Workshop at NIPS 2017. We argue one interesting question for future work in this domain will be how Bayesian methods can be used to make deep learning more interpretable. We believe the kind of work presented in this thesis would play an important role in such a quest.

Chapter 7

Appendix

7.1 Gradient flow of joint shape and feature density

In this section, we provide the details on how we derive gradient of Equation (3.12) and obtain Equation (3.14). Note that the derivation is a straightforward extension of the derivation in [1].

Let us consider the log of the joint shape and feature prior density

$$\log p(\tilde{x}, \hat{f}) = \log \Big(\frac{1}{n} \sum_{i=1}^{n} k(d_T(\tilde{x}, x_i), d_{L_2}(\hat{f}, f_i), \sigma_x, \sigma_f) \Big). \tag{7.1}$$

Then, the derivative of $\log p(\tilde{x}, \hat{f})$ with respect to $\tilde{x}$ is written in the following form

$$\begin{aligned} \frac{\partial \log p(\tilde{x}, \hat{f})}{\partial \tilde{x}} = & -\frac{1}{p(\tilde{x}, \hat{f})} \times \frac{1}{n} \times \frac{1}{\sigma_x \times \sigma_y} \\ & \times \sum_{i=1}^{n} \Big(k(d_T(\tilde{x}, x_i), d_{L_2}(\hat{f}, f_i), \sigma_x, \sigma_f) \\ & \times d_T(\tilde{x}, x_i) \times (d_{L_2}(\hat{f}, f_i))^2 \times \frac{\partial d_T(\tilde{x}, x_i)}{\partial \tilde{x}} \Big). \end{aligned} \tag{7.2}$$

Now the task comes to computing $\frac{\partial d_T(\tilde{x}, x_i)}{\partial \tilde{x}}$. Consider the template distance metric $d_T(\phi_{\tilde{x}}, \phi_{x_i}) = Area(inside(\tilde{x}) \triangle inside(x_i))$ where $\triangle$ denotes the set symmetric

difference. This metric can be written in the form of region integrals as follows [1]

$$\begin{aligned} d_T(\phi_{\tilde{x}}, \phi_{x_i}) &= \int_{\Omega} (1 - H(\phi_{\tilde{x}}(x)))H(\phi_{x_i}(x))\mathrm{d}x \\ &+ \int_{\Omega} H(\phi_{\tilde{x}}(x))(1 - H(\phi_{x_i}(x)))\mathrm{d}x \\ &= \int_{inside(\tilde{x})} H(\phi_{x_i}(x))\mathrm{d}x \\ &+ \int_{outside(\tilde{x})} (1 - H(\phi_{x_i}(x)))\mathrm{d}x \end{aligned} \tag{7.3}$$

For the region integrals in Equation (7.3), the derivative is well known [99], which is given by

$$\frac{\partial d_T(\tilde{x}, x_i)}{\partial \tilde{x}} = (2H(\phi_{x_i}) - 1). \tag{7.4}$$

By plugging Equation (7.4) into Equation (7.2), we obtain the gradient flow of $\log p(\tilde{x}, \hat{f})$ with respect to $\tilde{C}$:

$$\begin{aligned} \frac{\partial \log p(\tilde{x}, \hat{f})}{\partial \tilde{x}} &= \frac{1}{p(\tilde{x}, \hat{f})} \times \frac{1}{n} \times \frac{1}{\sigma_x \times \sigma_y} \\ &\times \sum_{i=1}^{n} \Big(k(d_T(\tilde{x}, x_i), d_{L_2}(\hat{f}, f_i), \sigma_x, \sigma_f) \\ &\times d_T(\tilde{x}, x_i) \times (d_{L_2}(\hat{f}, f_i))^2 \times (1 - 2H(\phi_{x_i})) \Big). \end{aligned} \tag{7.5}$$

Bibliography

[1] J. Kim, M. Çetin, and A. S. Willsky, "Nonparametric shape priors for active contour-based image segmentation," *Signal Processing*, vol. 87, no. 12, pp. 3021–3044, 2007.

[2] A. Foulonneau, P. Charbonnier, and F. Heitz, "Multi-reference shape priors for active contours," *International journal of computer vision*, vol. 81, no. 1, pp. 68–81, 2009.

[3] S. Chen and R. J. Radke, "Level set segmentation with both shape and intensity priors," in *International Conference on Computer Vision*. IEEE, 2009, pp. 763–770.

[4] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[5] D. W. Jacobs, "Robust and efficient detection of salient convex groups," *IEEE transactions on pattern analysis and machine intelligence*, vol. 18, no. 1, pp. 23–37, 1996.

[6] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[7] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *International journal of computer vision*, vol. 43, no. 1, pp. 7–27, 2001.

[8] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[9] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[10] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *Journal of computational physics*, vol. 79, no. 1, pp. 12–49, 1988.

[11] J. A. Sethian, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge university press, 1999, vol. 3.

[12] T. F. Cootes and C. J. Taylor, "A mixture model for representing shape variation," *Image and Vision Computing*, vol. 17, no. 8, pp. 567–573, 1999.

[13] P. Etyngier, F. Segonne, and R. Keriven, "Shape priors using manifold learning techniques," in *International Conference on Computer Vision (ICCV)*. IEEE, 2007, pp. 1–8.

[14] M. Kirschner, M. Becker, and S. Wesarg, "3d active shape model segmentation with nonlinear shape priors," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2011, pp. 492–499.

[15] A. Tsai, A. Yezzi Jr, W. Wells, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky, "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE Transactions on Medical Imaging*, vol. 22, no. 2, pp. 137–154, 2003.

[16] X. Bresson, P. Vandergheynst, and J.-P. Thiran, "A variational model for object segmentation using boundary information and shape prior driven by the mumford-shah functional," *International Journal of Computer Vision*, vol. 68, no. 2, pp. 145–162, 2006.

[17] D. Cremers, S. J. Osher, and S. Soatto, "Kernel density estimation and intrinsic alignment for shape priors in level set segmentation," *International Journal of Computer Vision*, vol. 69, no. 3, pp. 335–351, 2006.

[18] M. G. Uzunbas, O. Soldea, D. Unay, M. Cetin, G. Unal, A. Erçil, and A. Ekin, "Coupled nonparametric shape and moment-based intershape pose priors for

multiple basal ganglia structure segmentation," *IEEE transactions on medical imaging*, vol. 29, no. 12, pp. 1959–1978, 2010.

[19] F. Mesadi, M. Cetin, and T. Tasdizen, "Disjunctive normal shape and appearance priors with applications to image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 703–710.

[20] ——, "Disjunctive normal parametric level set with application to image segmentation," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2618–2631, 2017.

[21] B. Wang, X. Gao, J. Li, X. Li, and D. Tao, "A level set method with shape priors by using locality preserving projections," *Neurocomputing*, vol. 170, pp. 188 – 200, 2015.

[22] S. A. Eslami, N. Heess, C. K. Williams, and J. Winn, "The shape Boltzmann machine: a strong model of object shape," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 155–176, 2014.

[23] N. Ramesh, F. Mesadi, M. Cetin, and T. Tasdizen, "Disjunctive normal shape models," in *IEEE International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2015, pp. 1535–1539.

[24] W. A. Strauss, *Partial differential equations*. Wiley New York, 1992, vol. 92.

[25] J. N. Tsitsiklis, "Efficient algorithms for globally optimal trajectories," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1528–1538, 1995.

[26] S. O. R. Fedkiw and S. Osher, "Level set methods and dynamic implicit surfaces," *Surfaces*, vol. 44, p. 77, 2002.

[27] D. L. Chopp, "Computing minimal surfaces via level set curvature flow," *Journal of computational physics*, vol. 106, no. 1, pp. 77–91, 1993.

[28] D. Adalsteinsson and J. A. Sethian, "A fast level set method for propagating interfaces," *Journal of computational physics*, vol. 118, no. 2, pp. 269–277, 1995.

[29] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[30] M. Rosenblatt *et al.*, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[31] T. Cacoullos, "Estimation of a multivariate density," *Annals of the Institute of Statistical Mathematics*, vol. 18, no. 1, pp. 179–189, 1966.

[32] B. W. Silverman, *Density estimation for statistics and data analysis*. CRC press, 1986, vol. 26.

[33] L. Devroye, L. Györfi, and G. Lugosi, *A probabilistic theory of pattern recognition*. Springer Science & Business Media, 2013, vol. 31.

[34] A. T. Ihler, "Maximally informative subspaces: Nonparametric estimation for dynamical systems," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[35] N. Metropolis and S. Ulam, "The monte carlo method," *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.

[36] R. Eckhardt, "Stan ulam, john von neumann, and the monte carlo method," *Los Alamos Science*, vol. 15, no. 131-136, p. 30, 1987.

[37] N. Metroplis, "The beginning of the monte carlo method," *Los Alamos Science Special Issue*, 1987.

[38] C. P. Robert, *Monte carlo methods*. Wiley Online Library, 2004.

[39] J. Van Neuman, "Various techniques used in connection with random digits, collected works, 765–770," 1963.

[40] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[41] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[42] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[43] A. E. Gelfand and A. F. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American statistical association*, vol. 85, no. 410, pp. 398–409, 1990.

[44] E. Erdil, M. U. Ghani, L. Rada, A. O. Argunsah, D. Unay, T. Tasdizen, and M. Cetin, "Nonparametric joint shape and feature priors for image segmentation," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5312–5323, 2017.

[45] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[46] B. Van Ginneken, A. F. Frangi, J. J. Staal, B. M. Romeny, and M. A. Viergever, "Active shape model segmentation with optimal features," *IEEE Transactions on Medical Imaging*, vol. 21, no. 8, pp. 924–933, 2002.

[47] S. Milborrow and F. Nicolls, "Locating facial features with an extended active shape model," in *European Conference on Computer Vision (ECCV)*. Springer, 2008, pp. 504–513.

[48] M. de Bruijne, B. van Ginneken, M. A. Viergever, and W. J. Niessen, "Adapting active shape models for 3d segmentation of tubular structures in medical images," in *Information Processing in Medical Imaging*. Springer, 2003, pp. 136–147.

[49] W. Wang, S. Shan, W. Gao, B. Cao, and B. Yin, "An improved active shape model for face alignment," in *IEEE International Conference on Multimodal Interfaces*, 2002, p. 523.

[50] T. Heimann and H.-P. Meinzer, "Statistical shape models for 3d medical image segmentation: a review," *Medical Image Analysis*, vol. 13, no. 4, pp. 543–563, 2009.

[51] S. Dambreville, Y. Rathi, and A. Tannenbaum, "A framework for image segmentation using shape models and kernel space shape priors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 8, pp. 1385–1399, 2008.

[52] S. Zhang, Y. Zhan, M. Dewan, J. Huang, D. N. Metaxas, and X. S. Zhou, "Towards robust and effective shape modeling: Sparse shape composition," *Medical Image Analysis*, vol. 16, no. 1, pp. 265–277, 2012.

[53] O. Amadieu, E. Debreuve, M. Barlaud, and G. Aubert, "Inward and outward curve evolution using level set method," in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, vol. 3. IEEE, 1999, pp. 188–192.

[54] S. Jehan-Besson, M. Barlaud, and G. Aubert, "Dream 2 s: Deformable regions driven by an eulerian accurate minimization method for image and video segmentation," *International Journal of Computer Vision*, vol. 53, no. 1, pp. 45–70, 2003.

[55] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[56] R. Yang, M. Mirmehdi, X. Xie, and D. Hall, "Shape and appearance priors for level set-based left ventricle segmentation," *IET Computer Vision*, vol. 7, no. 3, pp. 170–183, 2013.

[57] A. Soğanlı, M. G. Uzunbaş, and M. Çetin, "Combining learning-based intensity distributions with nonparametric shape priors for image segmentation," *Signal, Image and Video Processing*, vol. 8, no. 4, pp. 789–798, 2014.

[58] E. Erdil, S. Yildirim, M. Cetin, and T. Tasdizen, "Mcmc shape sampling for image segmentation with nonparametric shape priors," in *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 411–419.

[59] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape," *International journal of computer vision*, vol. 72, no. 2, pp. 195–215, 2007.

[60] Y. Kihara, M. Soloviev, and T. Chen, "In the shadows, shape priors shine: Using occlusion to improve multi-region segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 392–401.

[61] L. A. Royer, D. L. Richmond, C. Rother, B. Andres, and D. Kainmueller, "Convexity shape constraints for image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 402–410.

[62] D. Cremers, N. Sochen, and C. Schnörr, "A multiphase dynamic labeling model for variational recognition-driven image segmentation," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 67–81, 2006.

[63] T. Chan and W. Zhu, "Level set based shape prior segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2005, pp. 1164–1170.

[64] E. Erdil, A. O. Argunsah, T. Tasdizen, D. Unay, and M. Cetin, "A joint classification and segmentation approach for dendritic spine segmentation in 2-photon microscopy images," in *International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2015, pp. 797–800.

[65] E. Erdil, L. Rada, A. O. Argunsah, I. Israely, D. Unay, T. Tasdizen, and M. Cetin, "Nonparametric joint shape and feature priors for segmentation of dendritic spines," in *International Symposium on Biomedical Imaging (ISBI)*, April 2016, pp. 343–346.

[66] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[67] J. Kim, J. W. Fisher, A. Yezzi, M. Çetin, and A. S. Willsky, "A nonparametric statistical method for image segmentation using information theory and curve evolution," *IEEE Transactions on Image processing*, vol. 14, no. 10, pp. 1486–1502, 2005.

[68] N. Houhou, J.-P. Thiran, and X. Bresson, "Fast texture segmentation model based on the shape operator and active contour," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008, pp. 1–8.

[69] O. Michailovich, Y. Rathi, and A. Tannenbaum, "Image segmentation using active contours driven by the bhattacharyya gradient flow," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2787–2801, 2007.

[70] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[71] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009, vol. 317.

[72] O. Söderkvist, "Computer vision classification of leaves from swedish trees," 2001.

[73] N. Thakoor, J. Gao, and S. Jung, "Hidden markov model-based weighted likelihood discriminant for 2-d shape classification," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2707–2719, 2007.

[74] M. U. Ghani, S. D. Kanık, A. Ö. Argunşah, T. Taşdizen, D. Ünay, and M. Çetin, "Dendritic spine shape classification from two-photon microscopy images," in *Signal Processing and Communications Applications Conference*. IEEE, 2015, pp. 939–942.

[75] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[76] M. S. Hassouna and A. A. Farag, "Multistencils fast marching methods: A highly accurate solution to the eikonal equation on cartesian domains," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1563–1574, 2007.

[77] A. C. Fan, J. W. Fisher III, W. M. Wells III, J. J. Levitt, and A. S. Willsky, "Mcmc curve sampling for image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2007, pp. 477–485.

[78] J. Chang and J. Fisher, "Efficient mcmc sampling with implicit shape representations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 2081–2088.

[79] J. Chang and J. W. Fisher III, "Efficient topology-controlled sampling of implicit shapes," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Sept 2012.

[80] S. Chen and R. J. Radke, "Markov chain monte carlo shape sampling using level sets," in *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, 2009, pp. 296–303.

[81] M. De Bruijne and M. Nielsen, "Image segmentation by shape particle filtering," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, vol. 3. IEEE, 2004, pp. 722–725.

[82] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, "(1996). markov chain monte carlo in practice."

[83] F. Chen, H. Yu, R. Hu, and X. Zeng, "Deep learning shape priors for object segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 1870–1877.

[84] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[85] C. Andrieu and G. O. Roberts, "The pseudo-marginal approach for efficient monte carlo computations," *The Annals of Statistics*, pp. 697–725, 2009.

[86] N. J. Higham, "Computing a nearest symmetric positive semidefinite matrix," *Linear Algebra and its Applications*, vol. 103, pp. 103 – 118, 1988.

[87] E. A. Nimchinsky, B. L. Sabatini, and K. Svoboda, "Structure and function of dendritic spines," *Annual Review of Physiology*, vol. 64, no. 1, pp. 313–353, 2002.

[88] M. U. Ghani, E. Erdil, S. D. Kanık, A. Ö. Argunşah, A. F. Hobbiss, I. Israely, D. Ünay, T. Taşdizen, and M. Cetin, "Dendritic spine shape analysis: A clustering perspective," in *European Conference on Computer Vision*. Springer, 2016, pp. 256–273.

[89] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *IEEE International Conference on Computer Vision*, vol. 1. IEEE, 2001, pp. 105–112.

[90] S. Nowozin and C. H. Lampert, "Global connectivity potentials for random field models," in *IEEE Conference Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 818–825.

[91] C. Rother, P. Kohli, W. Feng, and J. Jia, "Minimizing sparse higher order energy functions of discrete variables," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1382–1389.

[92] Y. Freund and D. Haussler, "Unsupervised learning of distributions on binary vectors using two layer networks," in *Advances in Neural Information Processing Systems*, 1992, pp. 912–919.

[93] T. Tieleman, "Training restricted Boltzmann machines using approximations to the likelihood gradient," in *International Conference on Machine learning*. ACM, 2008, pp. 1064–1071.

[94] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann machines," in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 448–455.

[95] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[96] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[97] F. Mesadi and T. Tasdizen, "Convex decomposition and efficient shape representation using deformable convex polytopes," *arXiv preprint arXiv:1606.07509*, 2016.

[98] F. Mesadi, E. Erdil, M. Cetin, and T. Tasdizen, "Image segmentation using disjunctive normal bayesian shape and appearance models," *IEEE Transactions on Medical Imaging*, 2017.

[99] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 18, no. 9, pp. 884–900, 1996.