GRAPHICAL MODELS FOR OBJECT SEGMENTATION

BY RUI HUANG

A dissertation submitted to the
Graduate School—New Brunswick
Rutgers, The State University of New Jersey
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
Graduate Program in Computer Science

Written under the direction of
Professor Dimitris N. Metaxas
and approved by

New Brunswick, New Jersey

October, 2008

ABSTRACT OF THE DISSERTATION

Graphical Models for Object Segmentation

by Rui Huang

Dissertation Director: Professor Dimitris N. Metaxas

Object segmentation, a fundamental problem in computer vision, remains a challenging task after decades of research efforts. This task is made difficult by the intrinsic variability of the object's shape, appearance, and its surrounding. It is compounded by the uncertainties arising from mapping the 3D world to the image plane and the noise in the acquisition systems. However, the human visual system often effectively entails the segmentation of the object from its background by fusing the bottom-up image cues with the top-down context. In this thesis we propose a novel probabilistic graphical modeling framework for object segmentation that effectively and flexibly fuses different sources of information, top and bottom, to produce highly accurate segmentation of objects in a computationally efficient manner. The main contributions of our work are:

1) We present a graphical model representing the relationship of the observed image features, the true region labels, and the underlying object contour based on the integration of Markov Random Fields (MRF) and deformable models. We propose two different solutions to this otherwise intractable joint region-contour inference and learning problem in the graphical model.

2) We introduce a Profile Hidden Markov Model (PHMM) built on the shape curvature sequence descriptor to improve the segmentation of specific objects. The special states and structure of PHMMs allow considerable shape contour perturbations and provide efficient inference and learning algorithms for shape modeling. Further embedding of the PHMM parameters captures the long term spatial dependencies on a shape profile, hence the global characteristics of a shape class.

3) We incorporate the proposed methods in a spatio-temporal MRF model to solve the video-based object segmentation problem. Our new model is a simultaneous object segmentation, background modeling, and pose estimation framework, which combines the top-down high-level object shape constraints with the bottom-up low-level image cues, and features a flexible graph structure induced by the motion information for more reliable temporal smoothness.

We demonstrate the effectiveness and robustness of all our methods in a wide variety of thorough experiments.

Acknowledgements

I would like to thank my advisor, Professor Dimitris Metaxas, for his guidance, support and encouragement throughout my Ph.D. years. He has always directed me toward the exciting areas in our field, yet still given me great freedom to pursue independent work. I still remember what he once said to me when I was frustrated and felt lost, "You don't need to work on every deadline. You need to sit down, take it slow, and go deeper." None of the work in this thesis would have happened without him.

I want to thank my secondary advisor, Professor Vladimir Pavlovic, who has been working closely with me since the very beginning and contributed numerous ideas and insights to the work I have finished here (and lots that I haven't finished). Again, I quote what Professor Metaxas once said to me when reviewing my writing, "Now your writing reads like something written by someone who really knows English..." and I told him, "Yes, Professor Pavlovic has rewritten it for me..."

I also thank my thesis committee members, Professor Doug DeCarlo, Professor Andrew Laine, and my qualifying exam committee members, Professor Ahmed Elgammal, Professor Casimir Kulikowski, and Professor Barbara Ryder, for their valuable suggestions regarding my early proposal and this thesis. It is a privilege for me to have each of them serve in my committees.

Last but not least, I want to thank all my colleagues from the Center for Computational Biomedicine Imaging and Modeling (CBIM), the Sequence Analysis and Modeling (SEQAM) Lab, the Computer Science Department, and the Biomedical Engineering Department. Who says peer pressure is a bad thing?

Dedication

To my parents: Xuequn Shuai and Xiaolin Huang. Without their love and support, I cannot achieve anything.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Graphical models for mid-level vision problems

Graphical models are a marriage between probability theory and graph theory [51]. A graphical model is a probabilistic model defined in terms of a graph in which the nodes represent random variables and the edges describe the probabilistic relationships among these variables. In particular, these probabilistic relationships are usually defined by conditional probabilities among the related variables or potential functions on the cliques of the graph, depend on whether the graph is directed or undirected. The joint probability distribution of a set of variables or the whole system can then be computed by taking products over the conditional probabilities or the potential functions defined on relevant nodes. The graph theory side of graphical models provides both an intuitive and compact representation for the complex probabilistic system, and well-studied graph-theoretic data structures for efficient graph-based algorithms. Probability theory, on the other hand, ensures the consistency of the whole system, and provides various statistical inference and learning methods to relate models to data. The intuitive and compact graph representation and its ability to model complex probabilistic systems make graphical models a powerful modeling tool in various research areas.

Graphical models have recently received extensive attention from many different research communities, including artificial intelligence, machine learning, computer vision, etc. One of the most successful examples in the computer vision area is the application of the Markov Random Field (MRF) model, a special case of undirected graphical models, to the low-level vision problems [34]. MRFs have been used for decades to model these problems because of their ability to capture the context of the image (i.e., dependencies among the neighboring image pixels) and to deal with the noise. Even though such problems can be elegantly represented by Markov random fields, the resulting energy minimization problems have been widely viewed as intractable. The MRF model has recently been revived largely because of the theoretical and computational developments in the graphical model area, especially the developments of efficient energy minimization algorithms such as Loopy Belief Propagation (LBP) [95] and graph

cuts [9], among others [87].

As we mentioned above, the power of the MRF model lies in its ability to capture the contextual information in the images, that is, besides the data energy term that relates the model to the data, there is also a smoothness energy term that imposes spatial smoothness on the model variables. From a Bayesian point of view, the data energy term is the likelihood term, the smoothness energy term is the prior term, and the model variables are solved by Maximum A Posteriori (MAP) estimation. The smoothness term is usually an Ising/Potts model, which is defined on a discrete collection of pairs of variables, with an energy which has one value when the two variables are the same, and a second value when the two are different. In most cases the first energy value is lower than the second to encourage spatial smoothness. Such a simple prior term can drastically improve the results of many low-level vision problems, yet this is still a low-level prior given that it is imposed globally and uniformly on the model variables without considering any specific knowledge to the content of the images.

Throughout this study, we consider a general problem, that is, can we solve a high-level vision problem by imposing a higher-level prior to the MRF model, and how? The general idea is to add a new layer of nodes representing the higher-level prior to the graph representation of the MRF model, hence a new set of variables to the probabilistic system, which naturally leads to some additional terms in the energy minimization procedure. The newly developed efficient energy minimization algorithms allow us to deal with such otherwise intractable problems. In particular, we investigate the object segmentation problem in both static 2D images/3D volumes and video sequences. As a side quest, we developed a new shape model to impose specific shape prior in our object segmentation framework, which turns out to be a comprehensive shape modeling method and bears the same idea of adding a new layer of nodes to a traditional graphical model (e.g., hidden Markov model in this case) to impose higher-level constraints. The same idea can be also used for object recognition and other mid-level vision problems, which will be briefly discussed in the end for future directions.

1.2 Image-based object segmentation

1.2.1 Problem statement

Image segmentation is one of the fundamental problems in computer vision. It is important to other computer vision tasks such as image understanding and retrieval, object recognition and categorization, etc., and medical imaging applications such as computer-aided diagnosis and surgery, etc. The main goal of image segmentation is to partition an image into its constituent

segments that have strong homogeneities with respect to certain criteria (e.g., image features). In practice, all the regions in an image are not always equally important, and one or more of them usually belong to a meaningful object, i.e., region of interest that is desired by the user or critical to the following tasks. In this study, we mainly focus on the object segmentation problem. That is, the goal of our segmentation algorithms is to find one specific object (region of interest) surrounded by a smooth and closed boundary contour. A seed point or small region is manually specified and the region of interest containing it is then segmented automatically. Therefore, without significant loss of modeling generality, we simplify the model parameters and avoid possible problems caused by segmenting multiple regions simultaneously.

Besides the object segmentation on 2D images, we also generalize the problem and our methods to 3D volumes. In the 3D case, accordingly, a 3D object (volume of interest) with smooth and closed surface is segmented. In the later part of this study, we further investigate the video-based settings, where the same object across all the frames is segmented.

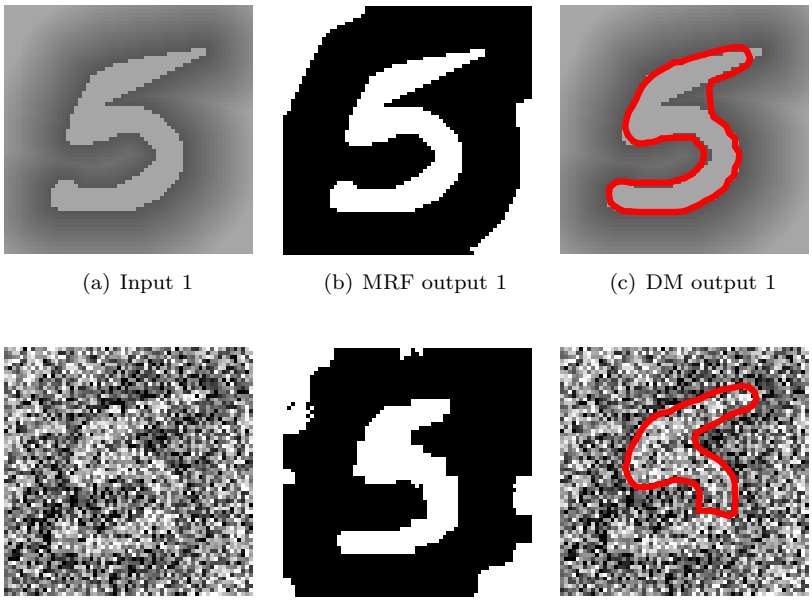
1.2.2 Previous work

There are two major categories of the segmentation methods: region-based segmentation and boundary-based segmentation. These two types of methods are naturally different in the way in which the image segments are represented (e.g., region labels vs. boundary locations), and also they are often different in the image features used for segmentation (e.g., region statistics vs. edges), therefore they have respective advantages and disadvantages.

Region-based methods assign image pixels to homogeneous regions according to the image features computed at individual pixels. Besides the classical region growing methods [44], this representation can naturally work with a wide variety of clustering and classification methods. The major disadvantages of these methods are that they do not explicitly model the region boundaries, hence it is hard to impose smoothness, topology, and shape constraints on the regions.

On the other hand, boundary-based methods extract region boundaries from the image. The boundary smoothness can be easily imposed because of the explicit modeling of the boundaries, although the oversmoothing sometimes may be hard to avoid. In the case of object segmentation, a priori knowledge of the object topology and shape can also be easily incorporated into boundary-based methods. Because these methods often rely on edge features, they are sometimes sensitive to image noise and initializations.

Figure 1.1 shows two example images and the segmentation results from a representative



(a) Input 1 (b) MRF output 1 (c) DM output 1

(d) Input 2 (e) MRF output 2 (f) DM output 2

Figure 1.1: Examples of 2D synthetic image segmentation

region-based method, a Markov Random Field model, and a representative boundary-based method, a Deformable Model (DM), respectively. The images were synthesized in a way similar to that in [26]. In [26] the 64×64 perfect image contains only 2 intensities in the total 256 intensity levels representing respectively the *object* (intensity level 160) and the *background* (intensity level 100), and Gaussian noise with mean 0 and standard deviation 60 is then added to the perfect image. In our example the background is made further complicated by introducing a non-uniform intensity level gradient. More precisely, the intensity level of the object is 160. The intensity levels of the background are increasing from 100 to 160 along the normal direction of the object contour (Figure 1.1(a)). This is motivated by the observation that in real world images, the background is often cluttered and contains overlapping image features of the foreground object, while in the local areas around the object boundaries the foreground and the background are usually still separable. Figure 1.1(b) shows the segmentation result of the MRF-based method using pixel intensities as image features. The object is segmented correctly, but some regions in the background are misclassified due to the overlapping features of the foreground and background. On the other hand, the deformable model (more precisely, the balloon model [21]) shows oversmoothing effect, i.e., either leaking from or not reaching the high-curvature parts of the object boundary, where the gradient in the normal direction is weak and unstable (Figure 1.1(c)).

The second example image (Figure 1.1(d)) is generated by adding Gaussian noise with mean 0 and standard deviation 60 to Figure 1.1(a). Note that the standard deviation of the noise is large enough to generate more overlapping image features and further confuse the foreground and the background. The segmentation result of the MRF-based method on this noisy image (Figure 1.1(e)) is somewhat similar to that in Figure 1.1(b), which shows that the MRF model can deal with image noise to some extent. But more significant misclassification occurs, especially in the background, because of the high noise level and more overlapping image features. The deformable model (Figure 1.1(f)) either sticks to the strong spurious edges caused by image noise or leaks from the weak true edges. We usually choose relatively conservative balloon forces to avoid leaking.

Throughout this study, we will show the experimental results in the same manner as Figure 1.1, that is, the results of MRFs are shown as label images, and those of DMs and our methods are shown as contours superimposed on the original images.

1.2.3 Motivation

A natural way to improve the segmentation results of a single methodology (either region-based or boundary-based) is to combine them and take advantages of both, especially when the two methodologies have complementary properties. In this study, we propose a graphical model framework to combine the region-based and the boundary-based segmentation methods, more specifically, the MRF model and the deformable model. To integrate these two fundamentally different traditional segmentation methods tightly, we need to construct a single graphical model to represent the relationship of the observed image pixels $\mathbf{y}$, the true region labels $\mathbf{x}$ and the underlying object contour $\mathbf{c}$.

It is natural to imagine that a real world image of an object is generated in such a way that the object shape is drawn first, and the image pixels *inside* the shape boundary is then considered the *object* and *outside* the *background*, and finally the object and the background appearances are rendered and noise is probably introduced by this rendering procedure. From a generative graphical model point of view, the object contour $\mathbf{c}$ is generated first with certain prior $P(\mathbf{c})$ (e.g., closed and smooth), and the region labels $\mathbf{x}$ are then generated conditioned on the object contour $\mathbf{c}$ with certain models $P(\mathbf{x}|\mathbf{c})$ (e.g., soft labeling or hard labeling), and finally the image pixels $\mathbf{y}$ are generated conditioned on the region labels $\mathbf{x}$ according to the appearance models of different regions $P(\mathbf{y}|\mathbf{x})$ (e.g., Gaussian models with different means and variances), as shown in Figure 1.2.

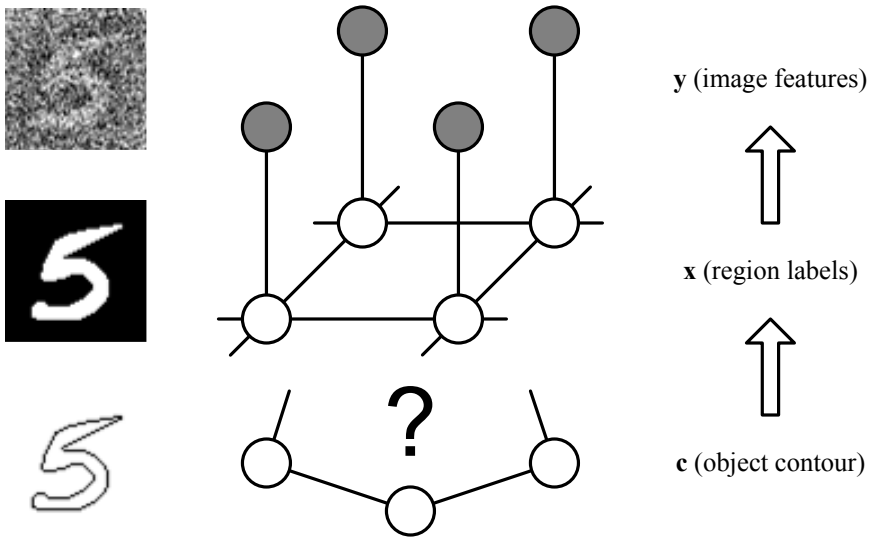


Figure 1.2: Generative graphical model for image segmentation

Another important reason to build our model in such a three-layer configuration is because the MRF model is a powerful tool to deal with the low-level vision problems, while the deformable model can be used to incorporate high-level cues into MRFs in addition to the simple Ising/Potts prior. It has been established that a combined top-down and bottom-up segmentation approach outperforms either of the one-direction methods [7, 59], because the top-down approaches can usually obtain a coarse segmentation efficiently using the high-level object-specific prior information, which can then be significantly refined by the bottom-up approaches using the low-level image cues. Our model bears the same idea of combining top-down and bottom-up segmentation.

1.2.4 Challenges and proposed solutions

The main goal of the first part of this study is to establish such a graphical model for the object segmentation problem. There are three challenges. The first challenge is the model structure problem. Though one can label regions according to boundaries or extract boundaries from regions, it is not easy to incorporate them in the same model, especially to have them take effect at the same time. This problem addresses how to fill in the missing part in Figure 1.2 and define the dependencies among the model variables. The second challenge is the inference problem. Once the model is defined, the object segmentation problem is considered an inference problem of the model. However, exact inference in such a model is usually intractable because of the huge state space and the couplings of model variables. The third challenge is the parameter

estimation problem. We also want to estimate the model parameters at the same time of performing inference (i.e., segmentation).

To tackle these problems we propose two different solutions to the integration of MRFs and DMs. The first solution uses a variational inference method to seemingly decouple the integrated model into two simpler models: one extended MRF model and one probabilistic deformable model, and the MAP estimation in the original model is obtained by solving the MAP problems of the two simpler models iteratively and incrementally. The second solution directly adapts the traditional deformable model formulation to explicitly model the relationship between the region labels and the contour nodes, and then solves the inference problem using the LBP algorithm. Both solutions also estimate the model parameters using the Expectation-Maximization (EM) algorithm. These two algorithms produce similar object segmentation results on 2D static images.

Besides the obvious implementation differences between the two solutions, the first solution is a tightly coupled model through variational inference in the seemingly decoupled submodules. The decoupling actually allows us to employ well-studied inference algorithms for both MRFs and DMs, making tasks such as the generalization to 3D object segmentation straightforward. The second solution, on the other hand, is a fully coupled probabilistic model that can be solved by a single statistical inference algorithm. It also allows us to investigate different inference methods and different message updating and passing schemes in LBP, which in a way justifies the decoupling in the first solution. The adaptation of the traditional deformable model contour representation also makes it easier to incorporate a more specific shape prior model instead of the global smoothing effect in the balloon model.

1.3 Video-based object segmentation

1.3.1 Problems and challenges

In this part of the study we address the problem of video-based object segmentation, that is, to segment the same object across all the frames in a video sequence. This task has become more and more important due to many applications such as human-computer interaction, video surveillance, video indexing and retrieval, etc., especially with the increasing availability of video data. Even though the modern static image segmentation algorithms have shown fairly good results with mild user interactions, it is still very difficult in practice to directly use these methods to segment video data either frame by frame or as 3D volumes. This is because the relatively low quality and high quantity of most video data often degrade the performances of

many static image segmentation algorithms, both in accuracy and running time, and sometimes forbid user interactions. On the other hand, the temporal dependencies carried in the video sequences usually provide many useful cues for potentially better and faster segmentation.

1.3.2 Previous work and our solution

We again employ the MRF model for the video-based object segmentation problem, because one particular advantage of the MRF model is that it can be easily extended to represent high dimensional data. For the video-based segmentation problem, a spatio-temporal MRF is usually constructed by adding to the regular MRFs one additional dimension that represents time. More precisely, a regular 2D MRF is used to model a single frame in the video sequence, and all the 2D MRFs can be stacked into one 3D MRF to model the whole sequence of 2D frames. A spatio-temporal MRF model naturally combines the spatial and temporal aspects of a video sequence and allows one to easily explore and integrate multiple cues for video-based object segmentation. In most previous work, however, only the dynamic information (e.g., the image differences) is used as the observation, and the appearance information is ignored. This is efficient for motion detection but not suitable for video-based object segmentation. Another potential problem of the traditional spatio-temporal MRF model is that the video sequences are usually treated as regular 3D volumes. Even though the smoothness constraint is imposed on the temporal dimension, due to the possible large movements of the object across different frames, the regular 3D structure does not correctly describe the temporal relationship between those model variables, hence the temporal smoothness is often incorrectly imposed.

We try to address these problems and also include the higher-level shape prior we used in the static image-based object segmentation framework in our new spatio-temporal MRF. There are four main modules in our model, which involve the four most important aspects of the video-based object segmentation problem. First, the bottom-up appearance model captures the low-level cues computed from the input data (e.g., intensity, color, texture, motion field, etc.). Second, the top-down prior model brings in the high-level object priors (e.g., topology, shape, color, texture, etc.) usually learned from the training data to guide the bottom-up approaches. Note that some features such as color and texture can be used as both low-level and high-level cues, but in different ways. In the bottom-up approaches, color and texture are only used for discovering homogeneous regions, while in the top-down approaches, specific distributions of color and texture learned from the training data are sought to separate different regions. Third, the spatial constraint term imposes spatial region smoothness on the segmentation in the image plane. This constraint greatly helps eliminate the inconsistencies in the segmentation caused by

the noise or other static imaging processes. Fourth, the temporal constraint term determined by the motion information (e.g., optical flow), instead of the regular 3D grid, imposes temporal smoothness on the otherwise unrelated static image frames. Again, we emphasize that even though the video-based object segmentation is performed on a stack of 2D images, i.e., a 3D volume, it is different from the 3D object segmentation we described in the first part of this study, and extra effort needs to be made to take into account the dynamic information among the 2D frames. All these four submodules are systematically incorporated into our framework. Furthermore, when some of these aspects are not known in advance (e.g., the shape prior), one can typically employ the Expectation Maximization (EM) algorithm to estimate the model parameters and perform segmentation simultaneously.

1.4 Shape analysis

1.4.1 Problems and challenges

As mentioned above, both static image-based and video-based object segmentation can be greatly improved by incorporating top-down shape priors, which leads to this part of the study, boundary-based 2D shape analysis. Shape analysis is also an important process for many computer vision and image processing applications, including image classification, recognition, retrieval, registration, segmentation, etc. An ideal shape model should be both invariant to global transformations (e.g., translation, rotation, scaling, etc.), and robust to local distortions (e.g., nonrigid transformations, occlusion, missing parts, etc.). To be used in our segmentation framework, the shape model also needs to provide both efficient and accurate matching algorithm. Here we present a new shape modeling framework that achieves all these goals. Application-wise, a comprehensive shape model should also be able to deal with various shape analysis tasks, e.g., shape matching, shape classification/recognition, shape segmentation, shape reconstruction, etc. Therefore we also investigate the effectiveness and robustness of our new shape model in all these different settings for completeness, even though the model is mainly motivated by the segmentation problem.

1.4.2 Previous work and our solution

Shape models can also be generally categorized into two classes: region-based shape models and boundary-based shape models. Region-based shape analysis methods make use of all the pixels within a shape region, hence are more robust to noise and suitable for shapes with complicated

internal structures and topologies. Boundary-based methods, on the other hand, mainly exploit the shape boundary information, which in many applications is both effective and efficient. For example, in image retrieval, the object contour is arguably the most convenient query that can be easily input by a user; in image segmentation, many boundary-based segmentation methods can be improved by a prior model of the boundary shape.

From the model perspective, there are two different levels in shape modeling. Adopting the terminology of [60], we use *shape description* to denote the numerical feature vector extracted from a given shape instance using a certain method (e.g., a curvature sequence), and *shape representation* the non-numerical, high-level representation of the shape (e.g., a graphical model) which preserves the important characteristics of the shape class.

To be consistent with the deformable model representation of the object contour used in our segmentation, in our shape model a shape instance is described by a curvature sequence descriptor. The curvature sequence descriptor has some attractive properties. First, it is invariant to the object translation. Second, the curvature computed at each contour point is rotationally invariant, so the descriptor is also invariant to the object rotation if the starting point is given. Otherwise, the object rotation causes a circular shift of the curvature sequence, which can be handled by the starting point detection procedure. Finally, the curvature sequence descriptor is not strictly invariant to the object scaling since a change of the contour length usually leads to a change of the curvature sequence length. One possible solution is to normalize all the shape contours to the same length or, equivalently, sample the contours to a fixed number of points. However, when there are nonrigid or local deformations or missing parts on the contour, the contour length may not be proportional to the actual object scale. Fortunately, the high-level representation of our shape model, a Profile Hidden Markov Model (PHMM) built on the low-level curvature sequence descriptions can address these problems and represent a class of similar shapes. PHMMs are a particular type of Hidden Markov Models (HMMs) with special states and architecture that can tolerate considerable shape contour perturbations, including rigid and non-rigid deformations, occlusions, and missing parts. PHMMs are more effective in the sequence matching task than ergodic HMMs because each observed salient feature in the observation sequence is modeled by a different state in the PHMM, instead of sharing a state with other observations as in the ergodic HMM. The sparseness of the PHMM structure provides efficient inference and learning algorithms for shape modeling and analysis. It is even more efficient than the ergodic HMM despite the larger number of states because of its strongly linear, left-to-right model structure. To capture the global characteristics of a class of shapes, the PHMM parameters are further embedded into a subspace that models long term spatial

dependencies. The new framework can be applied to a wide range of problems, such as shape matching/registration, classification/recognition, etc., and ultimately we can incorporate it into our object segmentation framework.

1.5 Organization

As outlined above, this study consists of three major contributions in the following three areas: static image-based object segmentation, video-based object segmentation, and 2D contour-based shape analysis. The remainder of this thesis, therefore, is organized as follows: Chapter 2 reviews the related work in all the three areas. This chapter also serves the purpose of defining the notations that will be used throughout the rest of the thesis. Chapter 3 describes the two different solutions to the image-based object segmentation problem. The 2D model is further generalized to 3D in this chapter. Since the shape analysis algorithms are extensively used in our video-based object segmentation method, we present our new shape model and its applications in Chapter 4, followed by Chapter 5, where the spatio-temporal framework for video-based object segmentation is presented. And finally we conclude and discuss the future work in Chapter 6.

Chapter 2
Related Work

2.1 Introduction

In this chapter, we review the related work in the three areas where we made our major contributions, which are image-based object segmentation, shape analysis, and video-based object segmentation. These topics are very broad and extensively studied, and we limit references to the work to which our work are most related, without meaning to slight the large body of other significant contributions. We also define the notations that will be used throughout the rest of the thesis.

2.2 Image-based object segmentation

As stated in Section 1.2.1, image-based object segmentation solves the problem of segmenting a specific object from the background in an image. We first review some general image segmentation methods, and then in greater detail the two representative methods, Markov random fields and deformable models, upon which our method is built. And finally some other object segmentation algorithms built on general image segmentation methods.

2.2.1 Image segmentation

Image segmentation is one of the fundamental problems in computer vision and has been extensively studied for decades. It is also one of the most difficult vision problems considering the wide variety of the image characteristics and contents. Numerous algorithms have been proposed over the years based on different assumptions to the input images and different approximations to the output results [83, 81, 32, 41], hence each method has its own advantages and disadvantages.

2.2.2 Region-based image segmentation and Markov random fields

Region-based methods assign image pixels to homogeneous regions according to the image features computed at individual pixels. The region growing method [44] and its variants are the early representatives in this category. MRFs are also an important region-based method, which has long been studied since 1980s [39, 5, 63]. MRFs and their *discriminative* counterpart, Conditional Random Fields (CRFs) or Discriminative Random Fields (DRFs) [57], have recently become the state-of-the-art segmentation methods due to the success of some newly developed energy minimization algorithms [87] such as BP [73] (more precisely, LBP [95]) and Graph Cuts [9]. These methods treat the segmentation problem as a classification problem (more specifically, a pixel-labeling problem), i.e., the image pixels are classified into different classes (regions) according to their features. Another representative group of region-based segmentation methods, graph partitioning methods such as Normalized Cuts [82], solve the segmentation problem in a flavor of clustering instead of classification, that is, the image pixels are grouped into clusters (regions) using clustering algorithms such as the spectral graph partitioning techniques. Since we focus on the MRF-based image segmentation method in this study, we briefly introduce MRFs in the rest of this section.

MRFs are a special case of undirected graphical models. See [52] for a comprehensive introduction to graphical models and [51] for more advanced topics. MRFs have been extensively used in many image analysis tasks, because of their ability to capture the context of the image (i.e., dependencies among the neighboring image pixels) and to deal with the noise. The MRF-based image segmentation methods are region-based methods, and the region labels are modeled by the MRF hidden variables and solved as a MAP inference problem.

A typical MRF model for image segmentation, as shown in Figure 2.1, is a graph with two types of nodes: observable nodes (shaded nodes in Figure 2.1, representing the image features computed at each pixel) and hidden nodes (clear nodes in Figure 2.1, representing the unknown region labels). The edges in the graph depict the probabilistic relationships among the nodes.

Let $n = w \times h$ be the number of the hidden/observable nodes (i.e., the number of pixels in a image of w by h). A configuration of the hidden layer is: $\mathbf{x} = (x_1, ..., x_n)$, where $x_i \in \mathcal{L}$, $i \in \mathcal{V}$. $\mathcal{L}$ is a set of region labels, e.g., $L = \{object, background\}$, and $\mathcal{V} = \{1, 2, ..., n\}$ is the pixel/node index set. Similarly, a configuration of the observable layer is: $\mathbf{y} = (y_1, ..., y_n)$, where $y_i \in \mathcal{R}^d$, $i \in \mathcal{V}$, and $\mathcal{R}^d$ is the d dimensional real space of the image features.

The probabilistic relationship between the hidden nodes and the observable nodes can be described by the *association* potential function: $\phi_i(x_i, y_i)$, which is usually a conditional model of

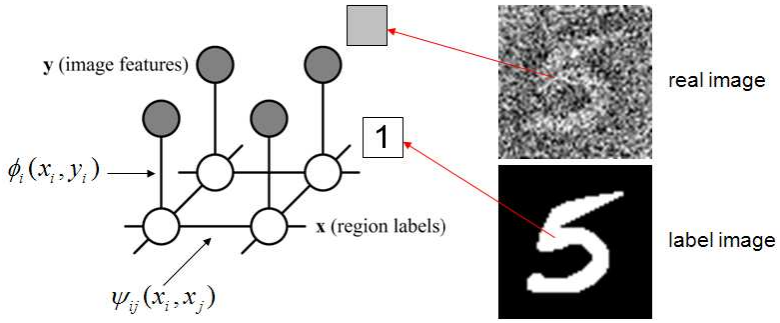


Figure 2.1: MRF model for image segmentation

the image feature given the region label at pixel i (e.g., a Gaussian or Gaussian mixture model); the pairwise relationship between the neighboring hidden nodes is described by the *interaction* potential function: $\psi_{ij}(x_i, x_j)$, which usually penalizes differences between the neighboring region labels to keep region smoothness (e.g., an Ising/Potts model). Note that the interaction potential functions can involve more hidden nodes if one divides the graph of the hidden layer into larger cliques. Here we mainly focus on the two node cliques, and accordingly, the *pairwise* interaction potential functions, which are adequate for our applications and simplify the inference problem. More details about these potential functions will be discussed later.

The image segmentation problem can be viewed as a problem of estimating the MAP solution of the MRF model:

$$\mathbf{x}_{\text{MAP}} = \arg\max_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}) \tag{2.1}$$

where

$$P(\mathbf{x}|\mathbf{y}) \propto P(\mathbf{y}|\mathbf{x})P(\mathbf{x}) \propto \prod_{i\in\mathcal{V}} \phi_i(x_i, y_i) \prod_{(i,j)\in\mathcal{N}} \psi_{ij}(x_i, x_j) \tag{2.2}$$

where $\mathcal{V}$, as mentioned above, is the pixel/node index set, and $\mathcal{N}$ is the edge set among the hidden nodes. The most commonly used edge set is defined by the regular grid on the image plane. Higher order neighborhood system can be used to capture the relationship among the nodes that are farther away.

The above MRF-MAP inference problem is equivalent to an optimization problem of minimizing the following energy function:

$$E(\mathbf{x}) = -log(P(\mathbf{x}|\mathbf{y})) = \sum_{i\in\mathcal{V}} -log(\phi_i(x_i, y_i)) + \sum_{(i,j)\in\mathcal{N}} -log(\psi_{ij}(x_i, x_j)) + Const \tag{2.3}$$

Note that we discarded $\mathbf{y}$ since it is known and can be encoded into the potential functions.

Because the exact MAP inference in the MRF model is computationally infeasible due to the large state space of $\mathbf{x}$ (i.e., $|\mathcal{L}|^n$), most MRF-based segmentation methods mainly differ in the means of approximating the MAP estimation or minimizing the corresponding energy function. These various techniques include Markov Chain Monte Carlo (MCMC) [39], iterated conditional modes (ICM) [5], maximizer of posterior marginals (MPM) [63], etc. Recently the BP algorithm [73] (more precisely, the loopy belief propagation algorithm [95]) and the Graph Cuts algorithm [9] have become the state of the art for the MRF-MAP inference problem. [26] presents a comparative analysis of some of the classical methods, while [88] focuses on the BP algorithm and the Graph Cuts algorithm. A more recent empirical study is presented in [87]. The MRF model parameters (i.e., the parameters in the potential functions) are often learned using the EM algorithm [98]. However, in the presence of multiple regions in the image, the automatic determination of the number of regions and the initial guess of the parameters could be difficult. More importantly, like other region-based methods, MRFs do not take account of object shape and topology, which means they may generate rough object boundaries and holes inside the objects.

2.2.3 Boundary-based image segmentation and deformable models

Boundary-based methods extract region boundaries from the image. These methods usually rely on edge detection to begin with, and the detected edges can then be linked together to form region boundaries [17]. Some methods directly extract higher-level geometric primitives such as lines and curves [27] using, e.g., Hough transform [45]. In this study we focus on another well-studied representative of boundary-based methods, deformable models, which fit a parameterized geometric primitive to the detected edges. We briefly review the DM-based image segmentation methods here.

A deformable model is usually a parameterized geometric primitive, whose deformation is determined by geometry, kinematics, dynamics, and even more sophisticated constraints (e.g., material properties, etc.) [66]. Snakes [54], a simple and widely used deformable model (also know as active contour model), are a parametric contour:

$$\begin{aligned} \Omega = [0,1] &\rightarrow \mathcal{R}^2 \\ s &\rightarrow \mathbf{c}(s) = (x(s), y(s)) \end{aligned} \tag{2.4}$$

where s is the parametric domain and x and y are the coordinate functions. The energy of the contour:

$$E(\mathbf{c}) = E_{\text{int}}(\mathbf{c}) + E_{\text{ext}}(\mathbf{c}) = \int_{\Omega} \omega_1(s) \left| \frac{\partial \mathbf{c}}{\partial s} \right|^2 + \omega_2(s) \left| \frac{\partial^2 \mathbf{c}}{\partial s^2} \right|^2 + F(\mathbf{c}(s))ds \tag{2.5}$$

where $\omega_1(s)$ and $\omega_2(s)$ control the "elasticity" and "rigidity" of the contour (i.e., the *internal* properties of the model), and F is the potential associated to the *external* forces determined by the image features or the user-desired model behaviors (e.g., expanding, shrinking, etc.). The final shape of the contour corresponds to the minimum of this energy.

To minimize the above energy term, one can use the discretized first order Lagrangian dynamics equation:

$$\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{f} \tag{2.6}$$

where $\mathbf{d}$ is discretized version of $\mathbf{c}$, $\mathbf{K}$ is the stiffness matrix determined by $\omega_1(s)$ and $\omega_2(s)$, and $\mathbf{f}$ is the generalized force vector determined by F.

Image gradient forces are most commonly used external forces to attract a deformable model to edges. However, when initialized far from the true boundary, the model often gets attracted to spurious image edges. Many variants of deformable models, such as Balloons [21] and Gradient Vector Flow (GVF) Snakes [94], have introduced different external forces to address this problem. In balloon models, namely, the deformable model is considered a balloon, which is inflated or deflated by additional forces along the normals of the deformable contour, and only stopped by stronger edges. The initial contour need no longer be close to the true boundary. Mathematically, a force along the normal direction to the curve at point $\mathbf{c}(s)$ with some appropriate amplitude f is added to the original forces:

$$\mathbf{f}' = \mathbf{f} + f\,\vec{\mathbf{n}}(s) \tag{2.7}$$

Deformable models can also be viewed in a probabilistic framework [65]. The internal energy $E_{\text{int}}(\mathbf{c})$ leads to a Gibbs prior distribution of the form:

$$P(\mathbf{c}) = \frac{1}{Z_i} \exp(-E_{\text{int}}(\mathbf{c})) \tag{2.8}$$

while the external energy $E_{\text{ext}}(\mathbf{c})$ can be converted to a sensor model with conditional probability:

$$P(\mathbf{I}|\mathbf{c}) = \frac{1}{Z_e} \exp(-E_{\text{ext}}(\mathbf{c})) \tag{2.9}$$

where $\mathbf{I}$ denotes the image, and $E_{\text{ext}}(\mathbf{c})$ is a function of the image $\mathbf{I}$.

The deformable models can then be fitted by solving the MAP problem:

$$\mathbf{c}_{\text{MAP}} = \arg\max_{\mathbf{c}} P(\mathbf{c}|\mathbf{I}) \tag{2.10}$$

where

$$P(\mathbf{c}|\mathbf{I}) \propto P(\mathbf{c})P(\mathbf{I}|\mathbf{c}) \tag{2.11}$$

One limitation of the DM-based methods is their sensitivity to the image noise, a common drawback of boundary-based methods. This may result in the deformable model being "stuck" in a local energy minimum of a noisy image. See [65] for a review of deformable models and [66] for further details about deformable models.

Besides the above mentioned *explicit* deformable models, the *implicit* deformable models such as level sets [70, 80, 71] are also widely used in image segmentation. The most obvious difference between these two types of deformable models is whether the boundary locations are modeled explicitly. Namely, in the explicit deformable models the boundary is modeled by a sequence of coordinates of the boundary points and evolved by moving the boundary points directly, while in the implicit deformable models the boundary is represented by the zero level set of a distance function and manipulated implicitly through the distance function. This intrinsic model difference leads to another difference between the two, that is, whether the region topology is strictly preserved. The level set methods make it easy to follow topology changes of the boundary (e.g., splitting, merging, developing holes, etc.), while the explicit deformable models are more suitable when the boundary topology is to be preserved (e.g., to impose strong shape constraints). It is worth noting, however, that there are both topologically adaptable snakes (explicit deformable models) [64] and topology preserving level sets (implicit deformable models) [42].

2.2.4 Hybrid image segmentation methods

Hybrid approaches attempt to combine region-based and boundary-based segmentations to alleviate deficiencies of the individual methods and improve the segmentation results. Many modern segmentation methods are formulated as an energy minimization procedure with both region-based and boundary-based features as energy terms, hence tend to blur the boundary between the two categories of segmentation methods. For example, many level set methods are, rather, considered region-based or hybrid methods. There are also various ways to incorporate region information into the explicit deformable models [76][49]. The most direct inspiration

of our work is [19], in which the authors proposes a way of integrating MRFs and deformable models. MRFs are used to initially estimate the region labels of the noisy images. Balloons are then used in the noise-reduced region maps to fit the object boundary. The result of the fitting is in turn used to update the MRF parameters and region label estimation. Final segmentation is achieved by iteratively integrating these processes. While this hybrid method attempted to take advantage of both MRFs and deformable models, the model coupling was loose. This may cause failure of deformable models if the initial estimation of the boundary by MRF is not closed, and it may also yield oversmoothed boundaries. Since our first attempts at building a fully coupled graphical model to combine MRFs and deformables [46][47], there have been other different choices of the combination. For example, the OBJ CUT algorithm [56] combines MRFs and the layered pictorial structures, the Pose Cut algorithm [11] combines MRFs and the stick figure model for human body segmentation, and the CRF-driven implicit deformable models [90] combine the Conditional random field model and the level set, etc.

2.3 Shape modeling

Many shape modeling techniques have been developed over years with different concerns and respective advantages [2, 60, 97]. We first review some HMM-based shape models that are most related to our work, and then introduce some successful and comprehensive shape models and their differences to our work.

2.3.1 Hidden Markov models for shape modeling

From the model perspective, our work is most similar to the HMM-based methods [43, 33, 3, 16, 6, 89]. Most HMM-based shape models are boundary-based shape models. The boundary of a shape is first extracted to form a shape contour, which can be further discretized into a set of landmark points. The shape description is then a sequence of shape attributes (e.g., curvature, radius, orientation, etc.) computed at these landmarks. HMMs are an ideal probabilistic sequence modeling method for the shape representation, because HMMs provide not only robust inference algorithms but also a probabilistic framework for training and building the model [74]. One of the earliest works on HMM-based shape model [43] used the autoregressive model parameters derived from the radius sequence of the contour points for shape description, and stationary or non-stationary HMMs with 2 to 6 states for shape representation. Promising results were presented. In [33], a shape was described by an 8-directional differential chain code.

Both fully connected and left-right HMMs were used for shape representations, and the left-right HMM achieved slightly higher classification rates than the fully connected HMM. A similar chain code description was also used in [3], but with a topologically different HMM for representation. This circular HMM-based shape model is insensitive to scaling and rotation to some extent. [16] used a Fourier spectral feature descriptor, and proposed a specially designed HMM topology and parameter re-estimation procedure to directly deal with these type of features. Recently, Bicego et al. [6] combined a curvature descriptor with an ergodic HMM for shape classification. Curvatures are treated as mixtures of Gaussians, and the Bayesian inference criterion was applied to select optimum number of the HMM states. This work was improved in [89], which used similar representation and description while focuses more on the design of the classification function by combining HMMs with generalized probabilistic descent method. However, most of these HMM-based methods concentrated on shape classification/recognition, while ignoring another important aspect of shape modeling: shape matching/registration [91].

In this study, we propose a new framework for shape modeling. Our model is easier to train and incorporates the global shape constraints that is critical for the shape matching, thus can be applied to more different and difficult tasks. The new model is a combination of the curvature sequence descriptor and Profile Hidden Markov Models (PHMMs) [28, 29]. PHMMs are strongly linear, left-right HMMs. A PHMM can model the entire shape profile more specifically than a general ergodic HMM, because each observed salient feature in the observation sequence is modeled by a different state in the PHMM, instead of sharing a state with other observations as in the ergodic HMM. This is crucial for shape matching. The special architecture of PHMMs contains *insert* and *delete* states, in addition to the regular *match* states, resulting in robustness to considerable shape contour perturbations, including rigid and non-rigid deformations, occlusions and missing contour parts. The adopted framework also leads to a computationally efficient set of algorithms for shape analysis. A further embedding of the model parameters can capture more global characteristics of a class of shapes. The embedding idea is similar to that of the statistical shape models such as the active shape models [23], except that we are embedding and constraining the global dependencies of the 1D curvature sequences, instead of the 2D spatial landmarks. Unlike previous HMM-based methods, our new framework is more comprehensive and comparable in performance to the state-of-the-art approaches. We will point out the differences between our model and those works in the next section.

2.3.2 Other shape models

In recent years, several comprehensive shape models have been proposed and widely used. Active shape models [23] represent similar shapes as variants of a mean shape. The "legal" variations are parameterized using statistical analysis, which requires the same number of corresponding sample points on each shape instance. For 2D contour shapes, a minimum description length approach [25] was proposed to find the correspondences before one can build the statistical models. The model embedding part of our model is similar to the shape variance parameterization part of the active shape models, however our model is more flexible without requiring the same number of sample points across all the shape instances, and the 1D nature of our model makes it more efficient to handle large rotation and occlusions. [1] presents an automatic learning framework for shape modeling using a differential-geometric treatment of planar shapes. It is also a statistical framework like our work, but no results on handling occlusions is reported. From the application perspective, our work is more similar to [37], which is based on syntactic matching. Our work is different in that we can build fully probabilistic models over classes of shapes. Shape context [4] also proves to be a successful descriptor for many shape analysis problems but it is not a high level representation. While such contour-less shape models using 2D descriptors can handle topologically more difficult shapes, our contour-based shape modeling approach can be both computationally efficient and robust to significant deformations, and it is easier to incorporate into contour-based segmentation methods. Disjoint region connecting methods such as [15] also allow our model to be applied to a wider range of 2D shapes.

2.4 Video-based object segmentation

As stated in Section 1.3.2, we address the video-based object segmentation problem with an improved spatio-temporal MRF model, which allows us to combine four different cues into the same framework. We first review the traditional spatio-temporal MRF model, and then the individual modules that we incorporate into it.

2.4.1 Spatio-temporal Markov random fields

One particular advantage of the MRF model is that it can be easily extended to represent high dimensional data. A spatio-temporal MRF model is constructed by stacking the regular MRFs that are used to model the data at different times to form a one dimensional higher MRF model. More precisely, for the video-based object segmentation problem, a regular 2D MRF is used to model a single frame in the video sequence, and all the 2D MRFs can be stacked

into one 3D MRF to model the whole sequence of 2D frames [62, 53, 92, 96]. In this setting, the spatio-temporal MRF model is three dimensional, though it is possible to have even higher dimensional models (e.g., a series of 3D volumes forming a 4D model).

A spatio-temporal MRF model naturally combines the spatial and temporal aspects of a video sequence and allows one to easily explore and integrate multiple cues for video-based object segmentation. In most previous work, however, only the dynamic information (e.g., the image differences) is used as the observation, and the appearance information is ignored. This is efficient for motion detection but not suitable for video-based object segmentation. Another potential problem of the traditional spatio-temporal MRF model is that the video sequences are usually treated as regular 3D volumes. Even though the smoothness constraint is imposed on the temporal dimension, due to the possible large movements of the object across different frames, the regular 3D structure does not correctly describe the temporal relationship between those model variables, hence the temporal smoothness is often incorrectly imposed.

Our model, on the other hand, tries to address these problems and also includes the higher-level shape prior we used in the static image-based object segmentation framework. There are four main modules in our model, which involve the four most important aspects of the video-based object segmentation problem. These are top-down high-level priors, bottom-up low-level features, spatial smoothness and temporal smoothness.

2.4.2 Combining top-down and bottom-up approaches

It has been established that a combined top-down and bottom-up segmentation approach outperforms either of the one-direction methods [7, 59]. The top-down approaches can usually obtain a coarse segmentation efficiently using the high-level object-specific prior information, which can then be significantly refined by the bottom-up approaches using the low-level image cues. MRFs are a powerful tool to deal with the low-level vision problems [34], and many efforts have been made to incorporate high-level cues into MRFs in addition to the simple Ising/Potts prior. These high-level priors can be generic shape and topology constraints, e.g., deformable models used in [46], or more object-specific shape models such as layered pictorial structures used in [56] and the stick figure model for human body used in [11]. Many of these shape prior models are formulated in the form of distance maps, commonly used in the level-set literature [78], for a proper probabilistic interpretation. In most cases, the shape prior is not given in advance, but has to be estimated at the same time of performing segmentation. This is usually solved iteratively using the EM algorithm.

Background subtraction is an effective approach to detect moving regions in image sequences. The usually low-level features (e.g., color) of each pixel in the background scene are modeled by a mixture of Gaussian distributions [36, 85], or a non-parametric model [30]. During the background subtraction process, false detection may occur due to the random noise or small movements of the background scene or the camera which are not captured by the background model. This can be suppressed by an additional stage of processing using spatial contextual information [30].

2.4.3 Combining spatial and temporal constraints

The other line of research addresses the noise problem using the spatio-temporal MRF model [62, 53, 92, 96]. The spatial smoothness is intrinsically improved due to the Ising/Potts prior in MRFs. Most of these models, however, are defined on a regular 3D grid neighborhood system, i.e., each node in the current frame is connected to the nodes with the same image coordinates in the previous and next frames, which may belong to different regions due to the possible large motions of the object. Therefore the regular Ising/Potts prior along the time dimension may oversmooth different regions. In [53] the image frames are divided into small patches and each patch is connected to the corresponding patches, determined by patch matching, in the neighboring frames. A different idea to determine real temporal neighbors is suggested in [58], where the optical flow is used to detect pixel correspondences in the time dimension, and the model nodes are connected to their real temporal neighbors defined by these correspondences instead of the 3D grid.

Another problem of the traditional spatio-temporal MRFs is that only the temporal information (e.g., the image differences between consecutive frames) is taken into account as the MRF observation model (i.e., the bottom-up features), and the appearance information is ignored, that is, they only model the motion information instead of the object or background appearance information. Hence these methods are more suitable for motion detection instead of object segmentation.

2.4.4 Relation to our work

Our goal is to build a spatio-temporal MRF model for video-based object segmentation. Instead of using the motion information between the image frames for the observations as in the traditional spatio-temporal MRF models, we use such dynamic information (e.g., optical

flow) to generate our model structure. Therefore our model is defined on a more flexible structure instead of the regular 3D grid, allowing the nodes in the model to be connected to more reliable temporal neighbors. The observation model of our framework is based on the foreground/background appearances, similar to the static image segmentation methods and the background modeling methods mentioned above, hence our model is more suitable for object segmentation. Furthermore, we incorporate a shape model into the otherwise bottom-up-only approach to improve the performance. To the best of the authors' knowledge, this is the first work to integrate all these different aspects for video-based object segmentation.

Chapter 3

Image-based Object Segmentation Using Graphical Models

3.1 Introduction

In this chapter we present two solutions to the image-based object segmentation problem. Recall the statement in Section 1.2.4, i.e., the main goal of this chapter is to establish a graphical model for the image-based object segmentation problem. There are three challenges. The first challenge is the model structure problem. Though one can label regions according to boundaries or extract boundaries from regions, it is not easy to incorporate them in the same model, especially to have them take effect at the same time (i.e., how to fill in the missing part in Figure 1.2) and define the dependencies among the model variables. The second challenge is the inference problem. Once the model is defined, the object segmentation problem is considered an inference problem of the model. However, exact inference in such a model is usually intractable because of the huge state space and the couplings of model variables. The third challenge is the parameter estimation problem. We also want to estimate the model parameters at the same time of performing segmentation.

To tackle these problems we propose two different solutions to the integration of MRFs and DMs. The first solution uses a variational inference method to seemingly decouple the integrated model into two simpler models: one extended MRF model and one probabilistic deformable model, and the MAP estimation in the original model is obtained by solving the MAP problems of the two simpler models iteratively and incrementally. The second solution directly adapts the traditional deformable model formulation to explicitly model the relationship between the region labels and the contour nodes, and then solves the inference problem using the loopy belief propagation algorithm. Both solutions also estimate the model parameters using the Expectation-Maximization (EM) algorithm. These two algorithms produce similar object segmentation results on 2D static images, yet have respective advantages, as we show in the experiments. The first solution can be straightforwardly extended to 3D object segmentation, while the second solution can be easily incorporated with a specific shape prior model instead

of the generally smoothed balloon model. We show results of 3D object segmentation in this chapter, and the results of object segmentation with specific shape prior are shown later after we introduce the embedded PHMM-based shape model in Chapter 4.

3.2 Solution I: Integration via variational inference

3.2.1 Integrated model I

As shown in Figure 1.2, the integrated model consists of three layers: the image feature layer $\mathbf{y}$, the region label layer $\mathbf{x}$, and the object contour layer $\mathbf{c}$.

The segmentation problem can be viewed as a *joint* MAP estimation problem:

$$(\mathbf{c}, \mathbf{x})_{\text{MAP}} = \arg\max_{\mathbf{c},\mathbf{x}} P(\mathbf{c}, \mathbf{x}|\mathbf{y}) \tag{3.1}$$

where

$$P(\mathbf{c}, \mathbf{x}|\mathbf{y}) \propto P(\mathbf{y}|\mathbf{x})P(\mathbf{x}|\mathbf{c})P(\mathbf{c}) \tag{3.2}$$

according to our generative model formulation.

One of the challenges is the huge state space of the model variables (i.e., all possible binary labelings $\mathbf{x}$ of the image on top of all possible contour configurations $\mathbf{c}$ in the image plane). However, as shown in Section 2.2.2 and Section 2.2.3, both the MRF-based and the deformable model-based segmentation methods have been extensively studied and both offer effective and efficient inference algorithms. To utilize these algorithms, one can loosely couple these two models as in the previous work [19]. More precisely, the MRF model is solved first, which gets rid of most noise. The binary label image is then used to fit the deformable model, which generates the smooth object boundaries. The MRF model parameters are updated according to the deformable model segmentation. This procedure repeats until convergence. However, in this loosely coupled model, suboptimal results from one model can cause serious problem to the other model.

Our first solution to the integrated model also tries to solve the two submodules separately but the structured variational inference technique is used to ensure the combination of the solutions to the two submodules is as close as possible to the original integrated model. Since one can solve the object contour $\mathbf{c}$ by the physics-based deformable model approaches, we do not need to model the object contour explicitly. This is achieved, as depicted in Figure 3.1, by having a single hidden node $\mathbf{c}$ in the object contour layer to represent the whole contour. This compact contour representation makes the model structure between the region label layer

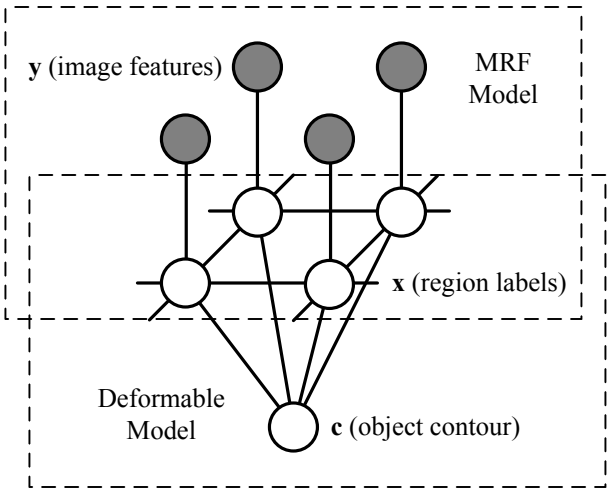


Figure 3.1: Graphical model I: integrated model

$\mathbf{x}$ and the object contour layer $\mathbf{c}$ rather simple (i.e., the contour node $\mathbf{c}$ is connected to every region label node x_i).

Under this model structure, Equation (3.2) can be factorized as

$$\begin{aligned} P(\mathbf{c},\mathbf{x}|\mathbf{y}) &\propto P(\mathbf{y}|\mathbf{x})P(\mathbf{x}|\mathbf{c})P(\mathbf{c}) \\ &\propto \prod_{i\in\mathcal{V}} \phi_i(x_i,y_i) \prod_{(i,j)\in\mathcal{N}} \psi_{ij}(x_i,x_j) \prod_{i\in\mathcal{V}} P(x_i|\mathbf{c})P(\mathbf{c}) \end{aligned} \tag{3.3}$$

$\phi_i(x_i,y_i)$ is the same as the association potential function in the traditional MRF model. If we assume the image features are simply the pixel intensities and are corrupted by white Gaussian noise:

$$\phi_i(x_i,y_i) = \frac{1}{\sqrt{2\pi\sigma_{x_i}^2}} \exp\left(-\frac{(y_i-\mu_{x_i})^2}{2\sigma_{x_i}^2}\right) \tag{3.4}$$

One can easily replace the Gaussian model with multivariate Gaussian if the image features have higher dimensions, and it is also straightforward to use more sophisticated observation models such as Gaussian mixture models or even non-parametric models.

$\phi_{ij}(x_i,x_j)$ is the interaction potential function to penalize differences between the neighboring labels (i.e., to keep local region smoothness), e.g.,

$$\phi_{ij}(x_i,x_j) = \exp\left(-\frac{(x_i-x_j)^2}{2\sigma^2}\right) \tag{3.5}$$

where σ controls the desired smoothness of neighboring hidden states. Data-dependent terms have been used in [8, 56], which assign different σ values depend on the image locations and

the local features. In our experience, this is not as critical of an issue as one is led to believe. It is clear that inference in a contrast-insensitive edge MRF captures the contrast (in appearance) between neighboring pixels. The data-dependent smoothing term is different from the Ising/Potts model (to some extent it is a more refined Ising/Potts model) in that it reinforces the standard observation model, at the expense of: 1) further departure from a generative MRF and 2) increased complexity and sensitivity in model estimation. Similar effects can often be accomplished using the differential (contrast-sensitive) observation features. Contrast-sensitive edges could be useful when the image is of high resolution and boundary details are important (e.g. in GrabCut [77] for image matting), here we choose the simple term to keep the clean structure of a generative model.

The dependency between the region labels $\mathbf{x}$ and the object contour $\mathbf{c}$ is modeled by the softmax function:

$$\begin{aligned} P(x_i = object|\mathbf{c}) &= 1/\left(1 + \exp(-\lambda dist(i, \mathbf{c}))\right) \\ P(x_i = background|\mathbf{c}) &= 1 - P(x_i = object|\mathbf{c}) \end{aligned} \tag{3.6}$$

induced by the signed distance of pixel i from contour $\mathbf{c}$:

$$dist(i, \mathbf{c}) = sign(i) \min_{s \in \Omega} \|loc(i) - \mathbf{c}(s)\| \tag{3.7}$$

where $sign(i) = 1$ if pixel i is inside contour $\mathbf{c}$ (i.e., belongs to the object), $sign(i) = -1$ when it is outside (i.e., belongs to the background), and $loc(i)$ denotes the spatial coordinates of pixel i. This can be achieved efficiently by computing the Euclidean distance transform, both in 2D [12] and 3D [35].

Note that the dependency between the region labels and the object contour is defined probabilistically, and the strength of the dependency is controlled by the parameter λ. As shown in Figure 3.2, the larger λ is, the more deterministic the dependency is. In particular, when $\lambda = inf$, the region labels and the contour are completely interdependent, and when $\lambda = 0$, they are completely independent. The uncertainty in region labels for a given contour allows us to process the two variables separately, and also arise as an attempt to model, e.g., image aliasing and changes in region appearance at boundaries.

Lastly, the prior term $P(\mathbf{c})$, as in Equation (2.8), can be represented as a Gibbs distribution when the shape prior is given by a parametric contour $\mathbf{c}$. Despite the compact graphical representation of the integrated model, the exact inference in the model is computationally

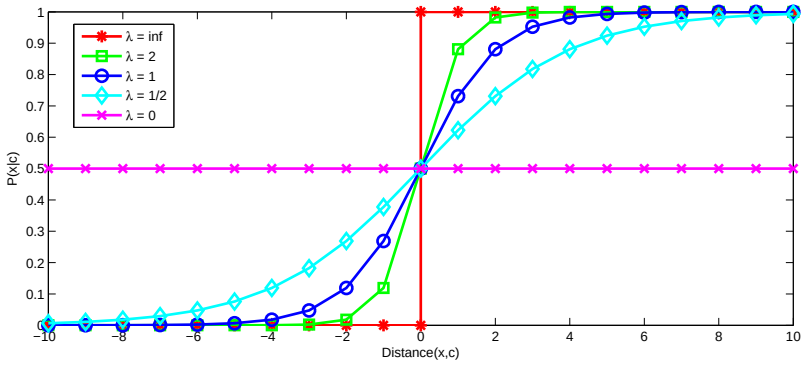


Figure 3.2: Dependency between the region labels and the object contour.

intractable due to the large state space of the model variables and the existence of loops in the graphical model, which preclude polynomial-time inference. To deal with these problems we propose an approximate, yet tractable, solution based on structured variational inference.

3.2.2 Approximate inference using structured variational inference

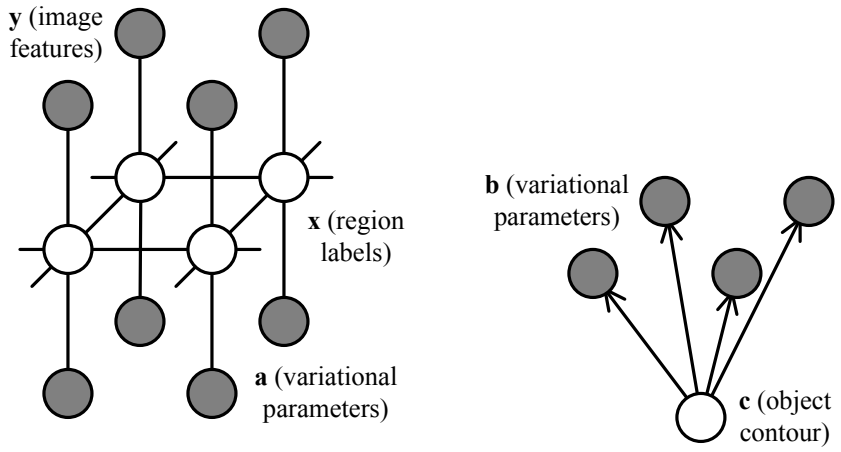
Structured variational inference techniques [50, 72] consider parameterized distribution which is in some sense close to the desired posterior distribution, but is easier to compute. Namely, for a given image $\mathbf{y}$, a distribution $Q(\mathbf{c}, \mathbf{x}|\mathbf{y}, \theta)$ with an additional set of *variational parameters* θ is defined such that the Kullback-Leibler (KL) divergence between $Q(\mathbf{c}, \mathbf{x}|\mathbf{y}, \theta)$ and $P(\mathbf{c}, \mathbf{x}|\mathbf{y})$ is minimized with respect to θ:

$$\theta^* = \arg\min_{\theta} \sum_{\mathbf{c},\mathbf{x}} Q(\mathbf{c}, \mathbf{x}|\mathbf{y}, \theta) \log \frac{P(\mathbf{c}, \mathbf{x}|\mathbf{y})}{Q(\mathbf{c}, \mathbf{x}|\mathbf{y}, \theta)} \tag{3.8}$$

The dependency structure of Q is chosen such that it closely resembles the dependency structure of the original distribution P. However, unlike P the dependency structure of Q *must* allow a computationally efficient inference.

In our case we define Q by decoupling the MRF model and the deformable model components of the original integrated model in Figure 3.1. The original distribution is factorized into two independent distributions: an extended MRF model Q_M with variational parameter $\mathbf{a}$ (Figure 3.3(a)) and a probabilistic deformable model Q_D with variational parameter $\mathbf{b}$ (Figure 3.3(b)). The *extended* MRF model means we have an additional layer to the traditional MRF model to deal with the shape prior, and the *probabilistic* deformable model means the contour is not fitted to the image directly, but to the probabilistic label image.

Because Q_M and Q_D are independent,



(a) Extended MRF model (b) Probabilistic deformable model

Figure 3.3: Graphical model I: decoupled models

$$Q(\mathbf{c}, \mathbf{x}|\mathbf{y}, \mathbf{a}, \mathbf{b}) = Q_M(\mathbf{x}|\mathbf{y}, \mathbf{a})Q_D(\mathbf{c}|\mathbf{b}) \tag{3.9}$$

According to the extended MRF model, we have:

$$\begin{aligned} Q_M(\mathbf{x}|\mathbf{y}, \mathbf{a}) &\propto Q_M(\mathbf{y}|\mathbf{x})Q_M(\mathbf{x}|\mathbf{a}) \\ &\propto \prod_{i\in\mathcal{V}} \phi_i(x_i, y_i) \prod_{(i,j)\in\mathcal{N}} \psi_{ij}(x_i, x_j) \prod_{i\in\mathcal{V}} P(x_i|a_i) \end{aligned} \tag{3.10}$$

On the other hand, the probabilistic deformable model yields:

$$\begin{aligned} Q_D(\mathbf{c}|\mathbf{b}) &\propto Q_D(\mathbf{b}|\mathbf{c})Q_D(\mathbf{c}) \\ &\propto \prod_i P(b_i|\mathbf{c})Q_D(\mathbf{c}) \end{aligned} \tag{3.11}$$

The optimal values of the variational parameters $\theta = (\mathbf{a}, \mathbf{b})$ are obtained by minimizing the KL-divergence. It can be shown, using e.g., [40], that the optimal parameters $\theta^* = (\mathbf{a}^*, \mathbf{b}^*)$ should satisfy the following equations:

$$\log P(x_i|a_i^*) = \sum_{\mathbf{c}} Q_D(\mathbf{c}|\mathbf{b}^*) \log P(x_i|\mathbf{c}) \tag{3.12}$$

$$\log P(b_i^*|\mathbf{c}) = \sum_{x_i\in\mathcal{L}} Q_M(x_i|\mathbf{y}, \mathbf{a}^*) \log P(x_i|\mathbf{c}) \tag{3.13}$$

Notice that the inference solutions, Equation (3.10) and Equation (3.11), together with the parameter optimizations, Equation (3.12) and Equation (3.13), form a set of *fixed-point equations*. Solution of this fixed-point set yields a tractable approximation to the intractable original posterior.

Since the state space of $\mathbf{c}$ (all possible contour configurations in the image plane) is too large, Equation (3.12) is still intractable. We simply use the winner-take-all strategy and approximate $Q_D(\mathbf{c}|\mathbf{b})$ as a delta function:

$$Q'_D(\mathbf{c}|\mathbf{b}) = \begin{cases} 1 & \text{if } \mathbf{c} = \arg\max_{\mathbf{c}} Q_D(\mathbf{c}|\mathbf{b}) \\ 0 & \text{else} \end{cases} \tag{3.14}$$

and Equation (3.12) can be simplified to:

$$P(x_i|a_i) = P(x_i|\mathbf{c}) \tag{3.15}$$

where $\mathbf{c} = \arg\max_{\mathbf{c}} Q_D(\mathbf{c}|\mathbf{b})$.

3.2.3 Algorithm description

The variational inference algorithm for the integrated MRF-DM model can now be summarized as:

Table 3.1: Image-based object segmentation algorithm I

Initialize contour $\mathbf{c}$;
while $(error > maxError)$ {
1. Calculate a band area $\mathcal{B}$ around $\mathbf{c}$. Perform the remaining steps inside $\mathcal{B}$;
2. Calculate $P(x_i|a_i)$ based on Equation (3.15) using $\mathbf{c}$;
3. Estimate MRF-MAP solution $Q_M(x_i|\mathbf{y}, \mathbf{a})$ based on Equation (3.10) using $P(x_i|a_i)$;
4. Calculate $\log P(b_i|\mathbf{c})$ based on Equation (3.13) using $Q_M(x_i|\mathbf{y}, \mathbf{a})$;
5. Estimate DM-MAP solution $Q_D(\mathbf{c}|\mathbf{b})$ based on Equation (3.11) using $\log P(b_i|\mathbf{c})$;

}

Simply put, in the extended MRF model, the true region labels are estimated using the BP algorithm in a band area around the estimated contour from the probabilistic deformable model, and the result in turn guides the probabilistic deformable model to an improved estimation of the contour. Steps 2 and 4 follow directly from Equation (3.15) and Equation (3.13). The details of steps 3 and 5, i.e., the inference in the two component models, are discussed in the following sections.

3.2.4 Extended Markov random field mdel

In step 3, the EM algorithm is used to estimate both the MAP solution of region labels $\mathbf{x}$ and the parameters of the model. In this study we primarily focus on the MRF association potential function parameters (we dropped the dependency on these parameters in Equation (3.2) for clarity).

Particularly, in E step, the MAP solution of region labels $\mathbf{x}$ is estimated based on current model parameters using BP. This is similar to the BP inference in the traditional MRFs, except that in our extended MRF model the association function is now extended to:

$$\Phi_i(x_i) = \phi_i(x_i, y_i)P(x_i|a_i) \tag{3.16}$$

We again note the difference from a traditional MRF model, due to the incorporated shape prior $P(x_i|a_i)$, which is calculated in step 2 of the algorithm. $\phi_i(x_i, y_i)$ and $\psi(x_i, x_j)$can be calculated using current MRF parameters.

BP is an inference method proposed by Pearl [73] to efficiently estimate Bayesian beliefs in the network by the way of iteratively passing messages between neighbors. It is an exact inference method in a network without loops. Even in a network with loops, the method often leads to good approximate of the otherwise intractable problems [93]. We hereby use the Loopy Belief Propagation algorithm [95].

As shown in step 1, in our algorithm belief propagation is again restricted to a single *band* of model variables around the current contour estimates because the region labels of the pixels far from the current contour are mostly determined by Equation (3.6) to be strongly either "object" or "background", hence there is no need to do inference there. Moreover, the banded inference significantly speeds up the whole algorithm. Although convergence of the banded algorithm is not guaranteed, in our experiments, the LBP algorithm does converge, usually in only several iterations.

In M step, the MRF association potential function parameters are updated based on the MAP solution of the region labels $\mathbf{x}$ using the following equations:

$$\mu_l = \sum_i Q_M(x_i = l|y_i, a_i)y_i / \sum_i Q_M(x_i = l|y_i, a_i) \tag{3.17}$$

$$\sigma_l^2 = \sum_i Q_M(x_i = l|y_i, a_i)(y_i - \mu_l)^2 / \sum_i Q_M(x_i = l|y_i, a_i) \tag{3.18}$$

where $l \in \mathcal{L}$.

3.2.5 Probabilistic deformable model

In step 5, according to Equation (2.9), we use the negative log term, $-\log P(\mathbf{b}|\mathbf{c})$, as the external energy in the deformable model. Given this "label image" energy landscape, the image force is simply $\nabla(\log P(\mathbf{b}|\mathbf{c}))$. With the additional balloon forces, this leads to the discretized first order Lagrangian dynamics equation:

$$\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \nabla(\log P(\mathbf{b}|\mathbf{c})) + k\vec{\mathbf{n}}(s) \tag{3.19}$$

We note that this formulation is different from that of [19] where the deformable model is fitted to a *binary* label image obtained from the MAP configuration of $\mathbf{x}$. In our method, we use a *probabilistic* measurement of the label of each pixel as specified in Equation (3.13).

Finally, following the definition in Equation (3.6) and Equation (3.7), we note that the gradient of the coupling energy at pixel i, $\nabla(\log P(\mathbf{b}|\mathbf{c}))$, can be shown to be:

$$\frac{\partial \log P(\mathbf{b}|\mathbf{c})}{\partial \mathbf{c}} = -\frac{\partial \log P(\mathbf{b}|\mathbf{c})}{\partial loc(i)} \tag{3.20}$$

3.2.6 Experiments - Model I

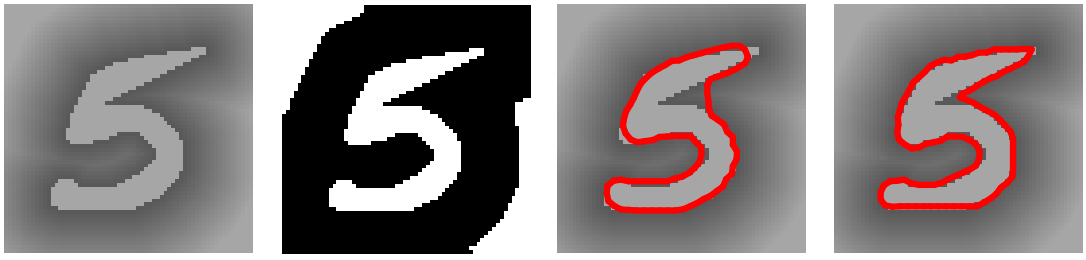
Our algorithm was implemented in MATLAB/C, and all the experiments were tested on a normal PC. Most of the experiments took a time varying from dozens of seconds to several minutes depending on the size of the images and the objects.

Experiments - Model I: synthetic images

The initial study of properties and utility of our method was conducted on the set of synthetic images (Figure 1.1(a) and Figure 1.1(d)) introduced in Section 1.2.1.

For the clean image (Figure 3.4(a)), Figure 3.4(b) shows the result of the traditional MRF-based method. The object is segmented correctly, however some regions in the background are misclassified due to the varying background. On the other hand, the deformable model fails because of the leaking from the high-curvature part of the object contour, where the gradient in the normal direction is too weak or unstable (Figure 3.4(c)). Our hybrid method, shown in Figure 3.4(d), results in a significantly improved segmentation.

On the noisy version (Figure 3.4(e)), Figure 3.4(f) shows more significant misclassification because of the high noise level and more overlapping image features, even though MRFs can usually deal with noise to some extent. The deformable model either sticks to spurious edges



(a) Input (b) MRF (c) DM (d) GM I

(e) Noisy (f) MRF (g) DM (h) GM I

Figure 3.4: Experiments on synthetic images

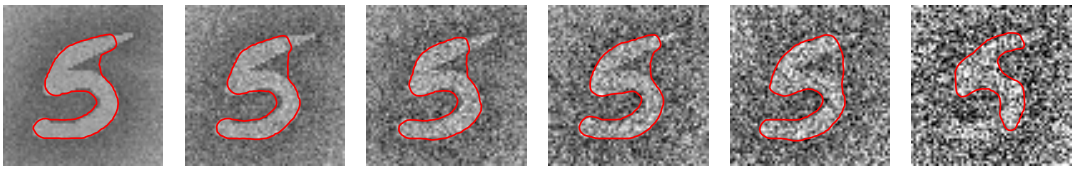
caused by image noise or leaks because of the weakness of the true edges (Figure 3.4(g)). Unlike the two traditional methods, our hybrid algorithm, depicted in Figure 3.4(h), correctly identifies the object boundaries despite the excessive image noise.

We further investigated the impact of different noise levels to these segmentation methods, and show the quantitative results in Figure 3.5. We added Gaussian noise with mean 0 to the clear image (Figure 3.4(a)), and increased the standard deviation of the noise from 0 to 70 by 10 each time, and tested the performances of the four methods (the second graphical model will be introduced next in Section 3.3). Note that the noise level 0 and 60 have been shown in(Figure 3.4)

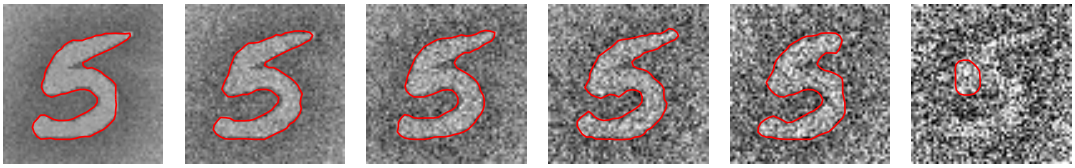
The first observation is that both of our hybrid methods are more robust to the initializations than the MRF model and the deformable model, as shown in Figure 3.6. For the MRF model, we initialize both the object model and the background model at multiple positions (shown as small circles in Figure 3.6), and use the best results for the quantitative comparison in Figure 3.5. For the deformable model, we also initialize the Balloons from different positions inside the object and keep the best results. For our own models, the position of the seed point does not affect the results substantially, so we use the same initialization position (i.e., the second case in Figure 3.6) for the comparison in Figure 3.5.

The second observation is that the results from the MRF model are noticeably worse than

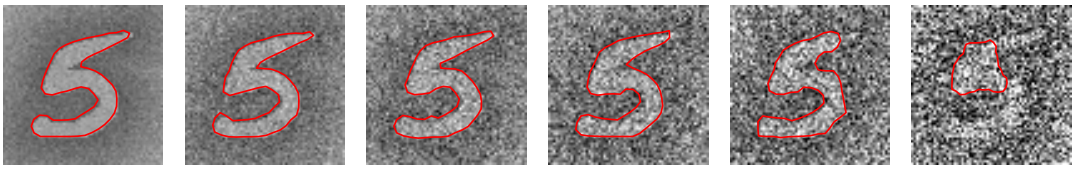
(a) $\sigma = 10$ (b) $\sigma = 20$ (c) $\sigma = 30$ (d) $\sigma = 40$ (e) $\sigma = 50$ (f) $\sigma = 70$



(g) $\sigma = 10$ (h) $\sigma = 20$ (i) $\sigma = 30$ (j) $\sigma = 40$ (k) $\sigma = 50$ (l) $\sigma = 70$



(m) $\sigma = 10$ (n) $\sigma = 20$ (o) $\sigma = 30$ (p) $\sigma = 40$ (q) $\sigma = 50$ (r) $\sigma = 70$



(s) $\sigma = 10$ (t) $\sigma = 20$ (u) $\sigma = 30$ (v) $\sigma = 40$ (w) $\sigma = 50$ (x) $\sigma = 70$

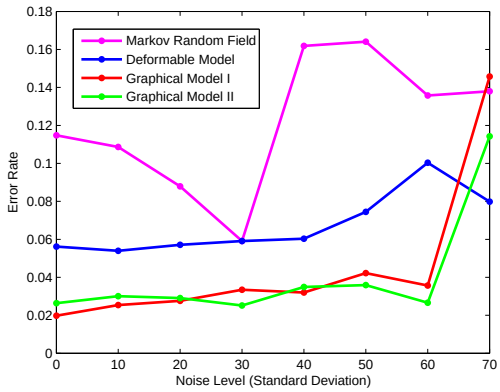
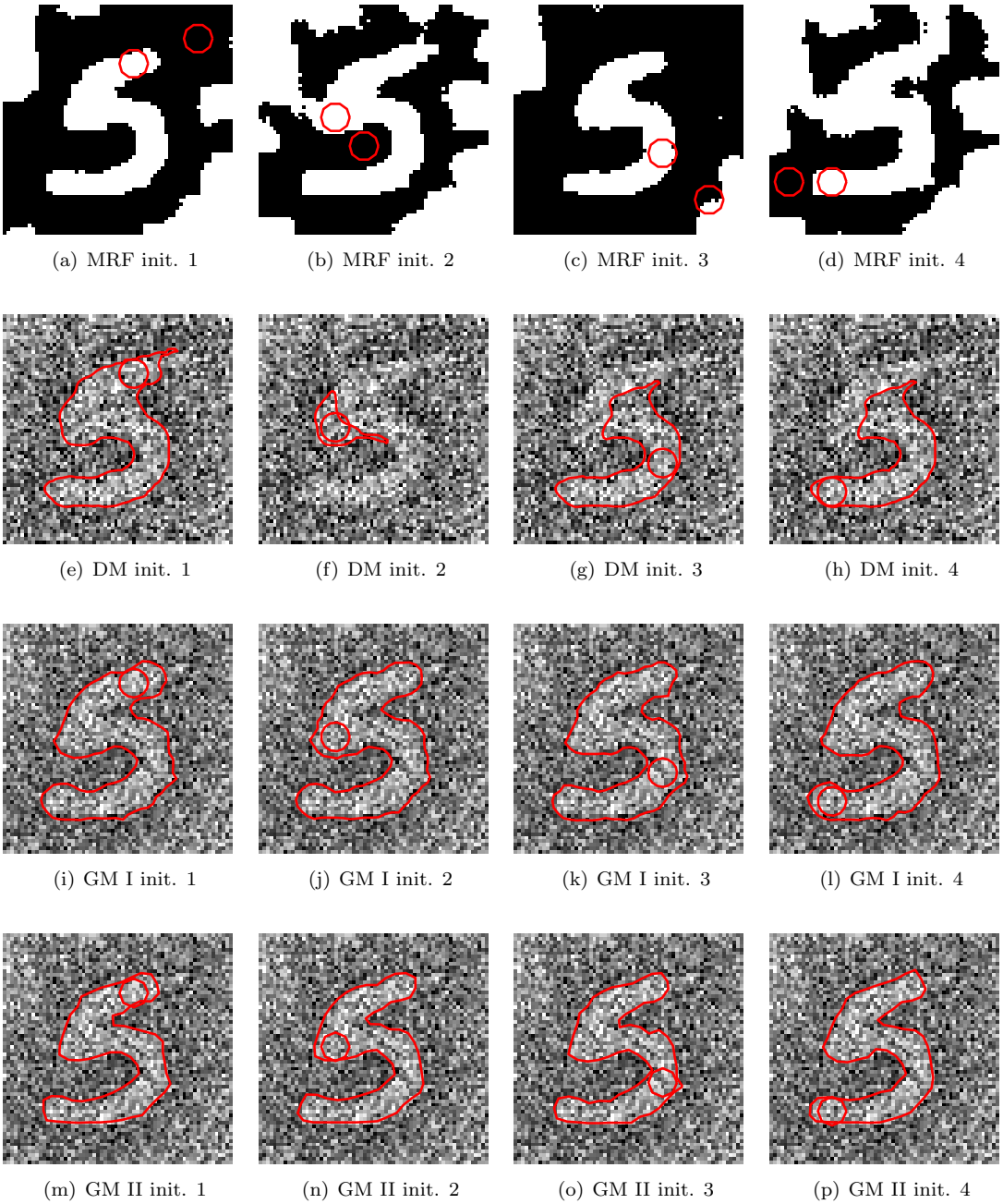


Figure 3.5: Comparison of different segmentation methods on different noise levels



(a) MRF init. 1 (b) MRF init. 2 (c) MRF init. 3 (d) MRF init. 4

(e) DM init. 1 (f) DM init. 2 (g) DM init. 3 (h) DM init. 4

(i) GM I init. 1 (j) GM I init. 2 (k) GM I init. 3 (l) GM I init. 4

(m) GM II init. 1 (n) GM II init. 2 (o) GM II init. 3 (p) GM II init. 4

Figure 3.6: Comparison of different segmentation methods on different initializations

the other methods. However, one important reason is that the way the error rates are computed favors the other object-oriented methods. We count the misclassified pixels in both the foreground (false negatives) and the background (false positives), and the MRF model often has significant error in the background, even though the segmentation of the object is relatively good. This shows that the MRF model does not focus on the object, but treats all the regions in the image equally. On the other hand, the deformable model either leaks from the high curvature parts of the object contour or sticks to the false edges inside the object. Since we always use relatively conservative balloon forces to avoid leaks, the error of the deformable model appears to be smaller than that of the MRF model due to significantly less false positives, but the segmented objects by the deformable model are often not as accurate as the ones by the MRF model due to more false negatives. This in some sense justifies why we need to combine these two models.

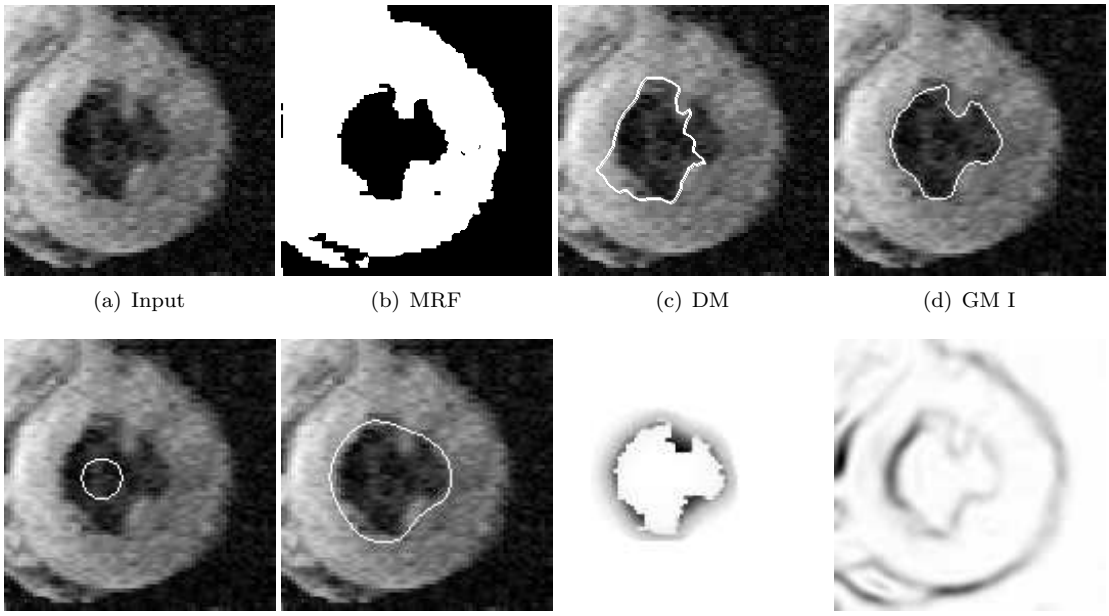
The third observation is that even though our hybrid methods clearly outperform the MRF model and the deformable model in most cases, their performance degrades severely when the noise level is extremely high. In this situation, due to the simple initialization method we use, the MRF component of our hybrid method cannot be estimated accurately.

Experiments - Model I: medical images

The above experiments with synthetic images outlined some of the benefits of our hybrid method. The real world images usually have significant, often non-white noise and contain multiple regions and objects, rendering the segmentation task a great deal more difficult. In this section we show results of applying our method to some medical images on which we can hardly get satisfying results with either the MRF-based or the deformable model-based methods alone.

In the following comparisons, we manually specified the inside/outside regions to get an initial guess of the parameters for the MRF-only method. For the deformable model method, we started the balloon model at several different initial positions and use the best results for the comparison. Again, our hybrid method is significantly less sensitive to the initialization of the parameters and the initial seed position.

Figure 3.7(a) shows a 2D MR image of the left ventricle of the human heart. Figure 3.7(b) is the result of the MRF-based method. While it is promising, the result still exhibits rough edges and holes. Figure 3.7(c) depicts the result of the DM-based method. Although we carefully chose the magnitude of the balloon forces, parts of the contour begin to leak while others stick to spurious edges. Our hybrid method, started from the initial contour shown in



(a) Input (b) MRF (c) DM (d) GM I

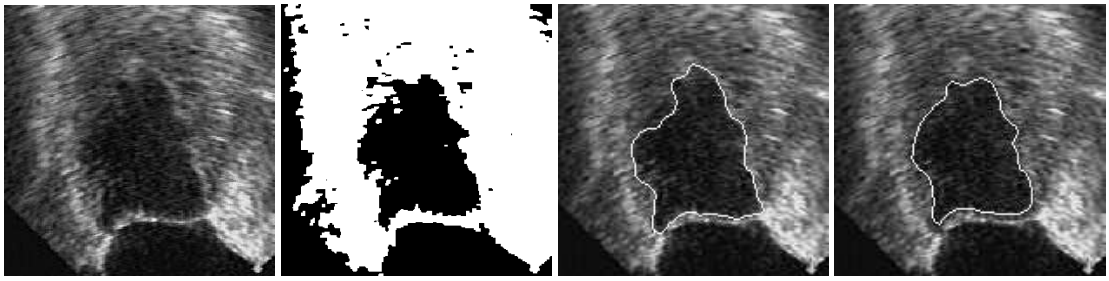
(e) Init. (f) Inter. (g) Energy (h) DM Energy

Figure 3.7: Experiments on medical images (1)

Figure 3.7(e), generated better result (Figure 3.7(d)). One of the intermediate iterations is shown in Figure 3.7(f). The corresponding external energy in the band area is depicted in Figure 3.7(g) (image intensities are proportional to the magnitude of the energy), showing a more useful profile than the traditional edge energy $-|\nabla(G_\sigma * I)|^2$ shown in Figure 3.7(h).

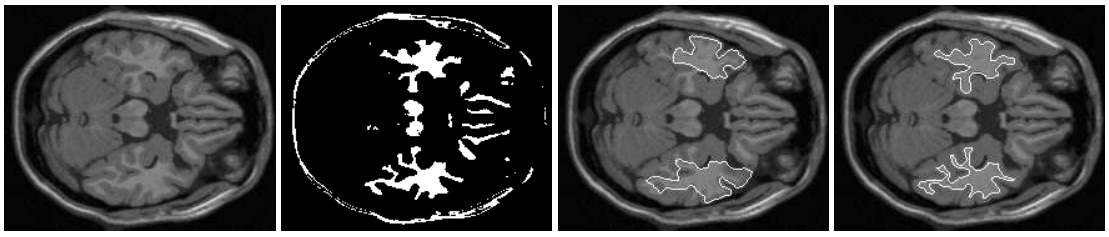
Figure 3.8(a) is an ultrasound image. The MRF gets rough edges and holes in the objects (Figure 3.8(b)) while the deformable model cannot escape a local minimum (Figure 3.8(c)) without leaking from other locations. Our hybrid method eliminates the rough edges and holes caused by the MRF while outlining the region more accurately than the deformable model (Figure 3.8(d)).

Figure 3.9(a) is an example of difficult images with complicated global properties, which makes it hard for the MRF-based method to automatically determine the number of regions and the initial values of the parameters. Figure 3.9(b) is obtained by manually specifying the inside/outside regions to get an initial guess of the parameters for the MRF model. Our method avoids this problem by creating and updating an MRF model locally and incrementally. Another problem with MRF-based method is that we cannot get a good representation of the segmented object directly from the model (e.g., extra efforts often need be made to extract the boundaries). The image is also difficult for deformable models because the boundaries of the



(a) Input (b) MRF (c) DM (d) GM I

Figure 3.8: Experiments on medical images (2)



(a) Input (b) MRF (c) DM (d) GM I

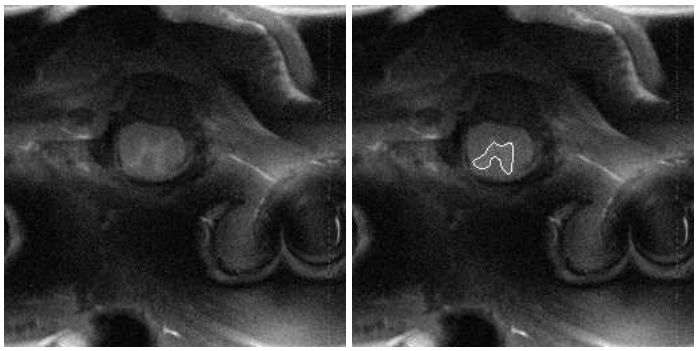
Figure 3.9: Experiments on medical images (3)

objects to be segmented have many high-curvature parts. Figure 3.9(c) exemplifies the over-smoothed deformable models. Our method's results, shown in Figure 3.9(d) does not suffer from the problems. For the deformable model method, we started the balloon model at several different initial positions and use the best results for the comparison. On the other hand, our hybrid method is significantly less sensitive to the initialization of the parameters and the initial seed position.

Finally, we show an image with both low contrast and low gradient (Figure 3.10(a)). Neither MRFs nor DMs can generate satisfying results, while our method's result, shown in Figure 3.10(b) does not suffer from either of the problems.

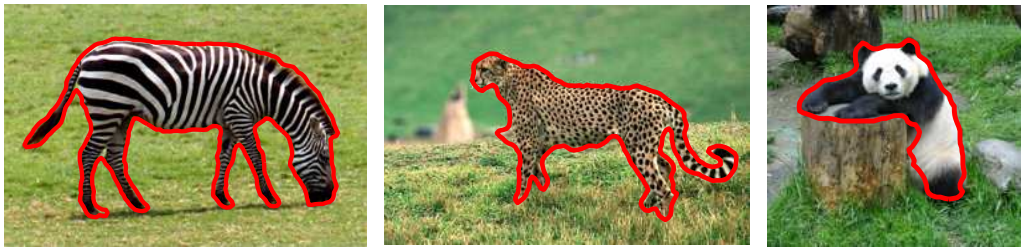
Experiments - Model I: natural images

We also applied our method to some natural images (Figure 3.11). In these experiments, the RGB color instead of gray-scale at each pixel is used as the image feature. A 5-component Gaussian mixture model is used as the object model and a 10-component Gaussian mixture as the background model. The adaptation is straightforward, i.e., only Equation (3.17) and Equation (3.18) need to be slightly modified.



(a) Input (b) GM I

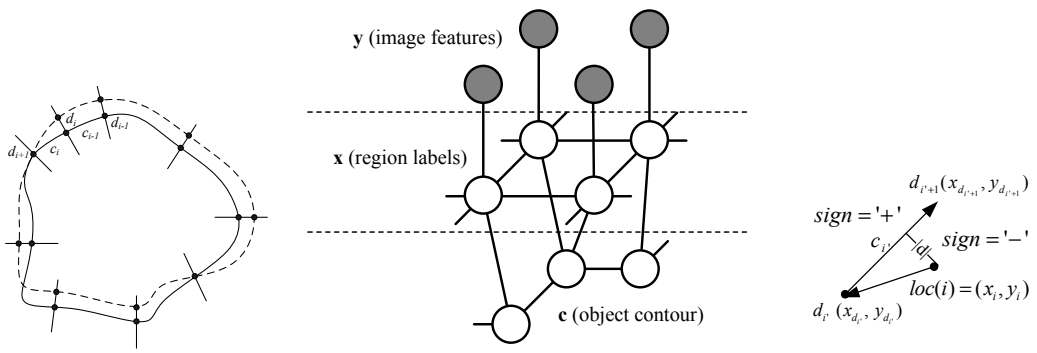
Figure 3.10: Experiments on medical images (4)



(a) Zebra (b) Cheetah (c) Panda

(d) Cow 1 (e) Cow 2 (f) Cow 3

Figure 3.11: Experiments on natural images



(a) Contour representation (b) Integrated model (c) Signed distance

Figure 3.12: Graphical model II

3.3 Solution II: Integration via contour factorization

3.3.1 Integrated model II

Recall that one of the challenges is the huge state space of the variables. However, as shown in Section 2.2.2, the MAP estimation can be approximated by algorithms such as BP and Graph Cuts if the model is properly factorized. Note that we can factorize the upper two layers of the model identical to the traditional MRF model. So the second solution we propose to solve this problem is to factorize the lower part of the model as well. We first discretize the object contour into a sequence of contour nodes $\mathbf{d} = (d_1, ..., d_m)$ where $d_i = (x_{d_i}, y_{d_i})$ are the contour node coordinates in the image plane, $i \in \mathcal{U}$, and $\mathcal{U} = \{1, 2, ..., m\}$. We then restrict the searching space of each contour node to a set of limited positions along the contour normals to reduce the state space of the contour model (Figure 3.12(a)). That is, the state space of each contour node d_i is restricted to a small number, e.g., k, of distinct locations.

Another challenge is to model the relationship between the contour nodes and the region labels. It is easy to define the region label of a pixel given the whole contour (e.g., region label is *object* if the pixel is inside the contour, and *background* outside), as shown in our first solution Equation (3.6), but not so given a single contour node. Thus we propose an adaptation to the traditional deformable model representation. We define $\mathbf{c} = (c_1, ..., c_m)$ where $c_i = (d_i, d_{i+1}), i \in \mathcal{U}$, and $d_{m+1} = d_1$. That is, each node c_i in the new representation is a segment of the object contour (Figure 3.12(a)). In turn, the state space of each contour segment c_i is of size k^2. The edges between the $\mathbf{x}$ layer and the $\mathbf{c}$ layer are determined based on the distance between the image pixels and the contour segments: each region label node x_i is connected to its nearest contour segment $c_{i'}$ where

$$i' = \arg\min_{j \in \mathcal{U}} dist(i, c_j) \tag{3.21}$$

This, again, is achieved efficiently by computing the Euclidean distance transform. Hence, the graph structure depends on the state of the contour nodes $c_{i'}$. A model of this type is often referred to as a *multinet* [38]. The new integrated model structure is depicted in Figure 3.12(b).

Now we can factorize Equation (3.2) as follows:

$$\begin{aligned} P(\mathbf{c}, \mathbf{x}|\mathbf{y}) &\propto P(\mathbf{y}|\mathbf{x})P(\mathbf{x}|\mathbf{c})P(\mathbf{c}) \\ &= \prod_{i \in \mathcal{V}} \phi_i^x(x_i, y_i) \prod_{(i,j) \in \mathcal{N}} \phi_{ij}^{xx}(x_i, x_j) \prod_{i \in \mathcal{V}} \phi_{ii'}^{xc}(x_i, c_{i'}) \prod_{i' \in \mathcal{U}} \phi_{(i'-1)i'}^{cc}(c_{i'-1}, c_{i'}) \prod_{i' \in \mathcal{U}} \phi_{i'}^{c}(c_{i'}) \end{aligned} \tag{3.22}$$

The association and interaction potential functions are identical to those defined before. $\phi_{ii'}^{xc}(x_i, c_{i'})$ models the dependency between the region labels $\mathbf{x}$ and the contour $\mathbf{c}$ using the softmax function:

$$\begin{aligned} \phi_{ii'}^{xc}(x_i = object|c_{i'}) &= 1/\left(1 + \exp(-\lambda dist(i, c_{i'}))\right) \\ \phi_{ii'}^{xc}(x_i = background|c_{i'}) &= 1 - \phi_{ii'}^{xc}(x_i = object|c_{i'}) \end{aligned} \tag{3.23}$$

induced by the signed distance of pixel i from the contour segment $c_{i'}$ (see Figure 3.12(c)):

$$dist(i, c_{i'}) = (d_{i'} - loc(i)) \times (d_{i'+1} - d_{i'})/|d_{i'+1} - d_{i'}| \tag{3.24}$$

where $loc(i)$ denotes the spatial coordinates of pixel i. This equation only holds when the pixel is close to the contour, which accords with our assumption. When the contour nodes are ordered counter-clockwise, the sign is positive when pixel i is inside the contour (i.e., belongs to the object) and negative when it is outside (i.e., belongs to the background).

$\phi_{(i'-1)i'}^{cc}(c_{i'-1}, c_{i'})$ simulates the discretized contour smoothness term:

$$\phi_{(i'-1)i'}^{cc}(c_{i'-1}, c_{i'}) = \exp\left(-\omega_1 \frac{|d_{i'-1} - d_{i'+1}|^2}{4h^2} - \omega_2 \frac{|d_{i'-1} + d_{i'+1} - 2d_{i'}|^2}{h^4}\right) \tag{3.25}$$

where $\omega_1(s)$ and $\omega_2(s)$ are the "elasticity" and "rigidity" parameters of the contour as in the traditional deformable models.

Finally $\phi_{i'}^{c}(c_{i'})$ can simulate the balloon force by defining

$$\phi^c_{i'}(c_{i'}, j, h) = p_{d_{i'}j} \times p_{d_{i'+1}h} \tag{3.26}$$

where $p_{d_{i'}j}$ is the prior of the state j at the contour node $d_{i'}$.

There are several desirable properties of the modified deformable segment model. It allows us to model the dependency between the region labels and the object contour, and the second order contour smoothness which involves three contour nodes is now conveniently modeled by a pairwise potential function.

3.3.2 Algorithm description

The inference algorithm for the integrated MRF-DM model can now be summarized as

Table 3.2: Image-based object segmentation algorithm II

Initialize contour $\mathbf{c}$;
while $(error > maxError)$ {
1. Calculate a band area $\mathcal{B}$ around $\mathbf{c}$. Perform the remaining steps inside $\mathcal{B}$;
2. Determine the links between the $\mathbf{x}$ layer and the $\mathbf{c}$ layer (Equation (3.21));
3. Calculate the k discretized states at each contour node along its normal;
4. Estimate the MAP solution $(\mathbf{c}, \mathbf{x})_{\text{MAP}}$ (Equation (3.1)) using BP with schedule $\mathcal{S}$;
5. Update model parameters θ_{MAP} (Equation (3.30)) and contour position $\mathbf{d}_{\text{MAP}}$;

}

3.3.3 Inference and learning

Again, the inference algorithm is carried out in a *band* area around the current contour. In this model, the primary reason for the band-limited BP update is that the definition of the signed distance requires the pixels to be close to the segments.

The implementation of BP in this model is slightly more difficult than the BP algorithm in a traditional MRF model, since the model structure is more complicated and keeps changing. One can solve this by converting the model into equivalent a factor graph, and use the BP algorithm for factor graphs [55, 95]. Here we give the straightforward modified algorithm for our specific model Figure 3.13.

Examples of the message passing rules are given in the following equations:

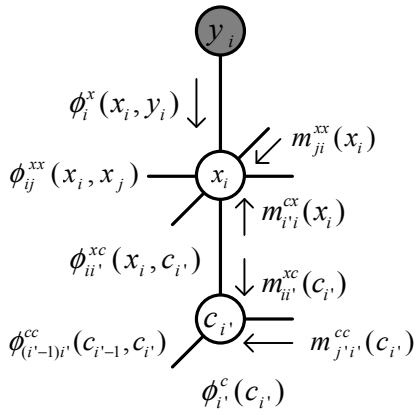


Figure 3.13: Message passing rules

$$
\begin{aligned}
m_{ji}^{xx}(x_i) &= \sum_{x_j}[\phi_j^x(x_j, y_j) \prod_{k \in \mathcal{N}(j) \setminus i} m_{kj}^{xx}(x_j) m_{j'j}^{cx}(x_j) \phi_{ij}^{xx}(x_i, x_j)] \\
m_{ii'}^{xc}(c_{i'}) &= \sum_{x_i}[\phi_i^x(x_i, y_i) \prod_{k \in \mathcal{N}(i)} m_{ki}^{xx}(x_i) \phi_{ii'}^{xc}(x_i, c_{i'})] \\
m_{i'i}^{cx}(x_i) &= \sum_{c_{i'}}[\phi_{i'}^c(c_{i'}) \prod_{k' \in \mathcal{U}(i')} m_{k'i'}^{cc}(c_{i'}) \prod_{k \in \mathcal{N}'(i') \setminus i} m_{ki'}^{xc}(c_{i'}) \phi_{ii'}^{xc}(x_i, c_{i'})] \\
m_{j'i'}^{cc}(c_{i'}) &= \sum_{c_{j'}}[\phi_{j'}^c(c_{j'}) \prod_{k' \in \mathcal{U}(j') \setminus i'} m_{k'j'}^{cc}(c_{j'}) \prod_{k \in \mathcal{N}'(j')} m_{kj'}^{xc}(c_{j'}) \phi_{j'i'}^{cc}(c_{j'}, c_{i'})]
\end{aligned} \tag{3.27}
$$

We have defined $\mathcal{N}$ as the set of neighbors in the region label layer, and here $\mathcal{N}'$ denotes the set of neighbors in the contour layer. These rules are based on the sum-product scheme. The max-product has analogous formula, with the marginalization replaced by the maximum operator. We empirically study these two message update schemes in Section 3.3.4. At convergence, the beliefs of the pixel labels and contour segments are

$$
b_i(x_i) = \phi_i^x(x_i) \prod_{k \in \mathcal{N}(i)} m_{ki}^{xx}(x_i) m_{i'i}^{xc}(x_i) \tag{3.28}
$$

$$
b_{i'}(c_{i'}) = \phi_{i'}^c(c_{i'}) \prod_{k' \in \mathcal{U}(i')} m_{k'i'}^{cc}(c_{i'}) \prod_{k \in \mathcal{N}'(i')} m_{ki'}^{xc}(c_{i'}) \tag{3.29}
$$

A crucial question in this BP process is that of the "right" message passing schedule [55, 95]. Different schedules may result in different stable/unstable configurations. For instance, it is widely accepted that short graph cycles deteriorate the performance of the BP algorithm. We empirically study this question in Section 3.3.4 and show that good schedules arise from understanding of the physical processes involved.

Lastly, the model parameter are estimated using the EM algorithm. More precisely, when the BP algorithm converges, the MRF association potential function parameters can be updated using following equations:

$$\begin{aligned} \mu_l &= \sum_i b(x_i = l) y_i / \sum_i b(x_i = l) \\ \sigma_l^2 &= \sum_i b(x_i = l)(y_i - \mu_l)^2 / \sum_i b(x_i = l) \end{aligned} \tag{3.30}$$

where $l \in \{object, background\}$ and $b(\cdot)$ denotes the current belief of the region labels.

In this model, the contour $\mathbf{c}$ is also updated according to the belief propagation (instead of the physics-based approach used in the previous model). Because the edges between the $\mathbf{x}$ layer and the $\mathbf{c}$ layer are determined by the distance between pixels and contour nodes, they also need to be updated after each iteration in the inference process. With the newly linked model, a new iteration can be executed.

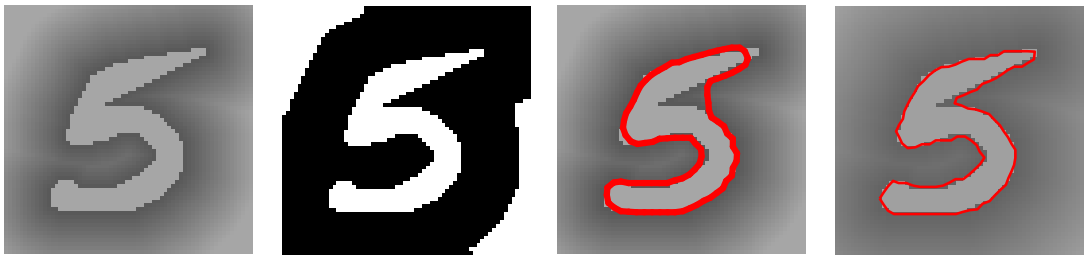
3.3.4 Experiments - Model II

Experiments - Model II: Comparison of different segmentation methods

Again, the initial study of properties and utility of our method was conducted on the set of synthetic images (Figure 1.1(a) and Figure 1.1(d)). The results (Figure 3.14(d) and Figure 3.14(h)) are similar to those of the first solution (Figure 3.4(d) and Figure 3.4(h)). A more detailed quantitative comparison was previously shown in Figure 3.5.

Figure 3.15 shows one of the intermediate iterations. Figure 3.15(a) depicts the linkage between the region label layer and the deformable contour (i.e., the Voronoi map) at the beginning of the iteration. The intensity of a Voronoi region is proportional to the index of a contour node, that is, all the pixels in one of the Voronoi regions are connected to the same contour node. The zero intensity region is outside the band and hence ignored. Figure 3.15(b) is the belief image of region labels after belief propagation, and the intensity of each pixel is proportional to the probability of that pixel being inside the contour. Figure 3.15(d) shows beliefs of contour nodes which imply their tendency of movement. We can clearly observe from the figure that most of the nodes have stopped moving (horizontal bright line in the middle of Figure 3.15(d)) while very few are still moving forward or backward. The contour position after that iteration is shown in Figure 3.15(c).

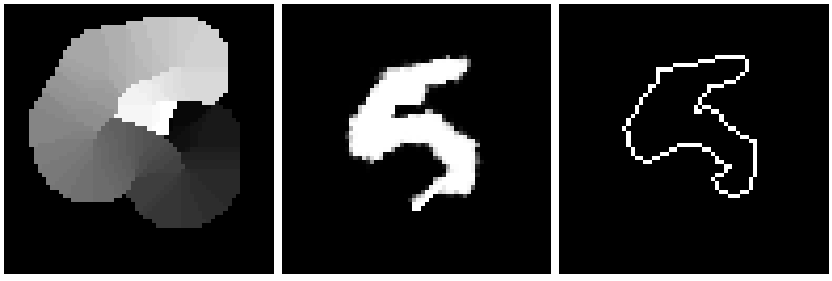
Another set of synthetic experiments are shown in Figure 3.16.



(a) Input (b) MRF (c) DM (d) GM II

(e) Noisy (f) MRF (g) DM (h) GM II

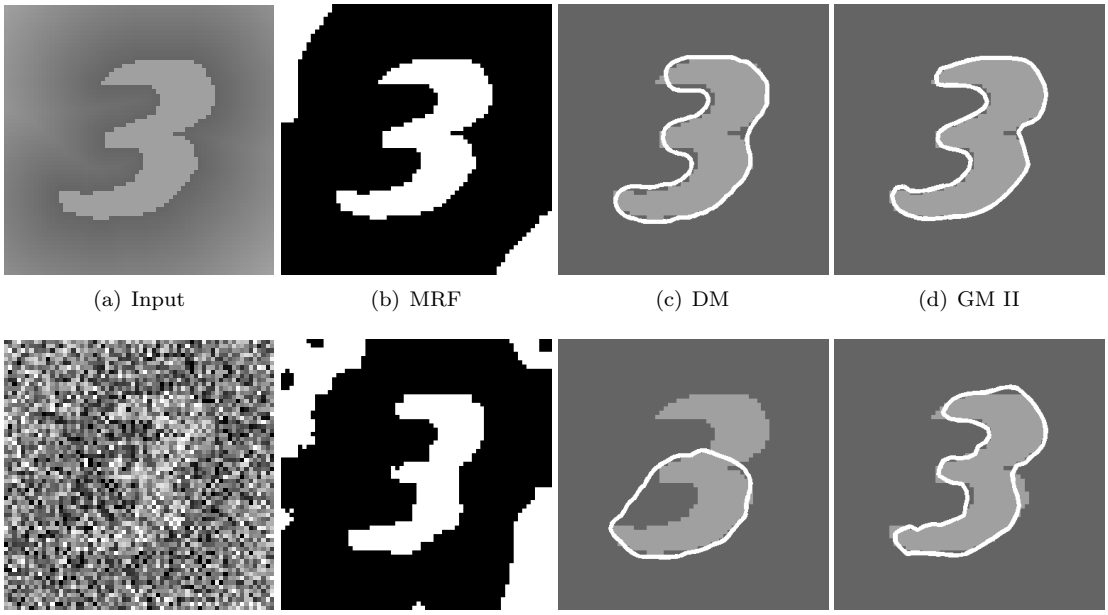
Figure 3.14: Experiments on synthetic images (1)



(a) V-map (b) $b(s)$ (c) c_{MAP}

(d) $b(c)$

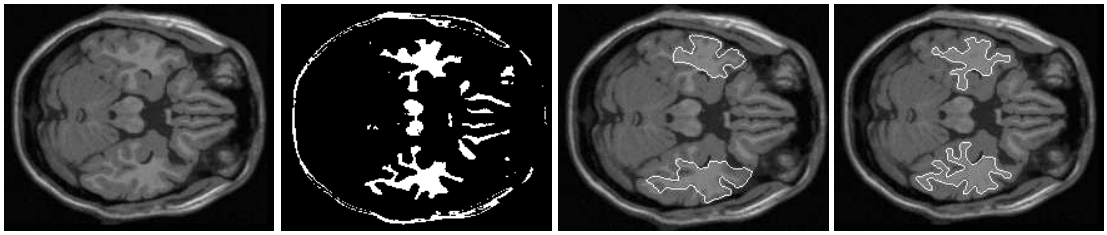
Figure 3.15: Intermediate results



(a) Input (b) MRF (c) DM (d) GM II

(e) Noisy (f) MRF (g) DM (h) GM II

Figure 3.16: Experiments on synthetic images (2)



(a) Input (b) MRF (c) DM (d) GM II

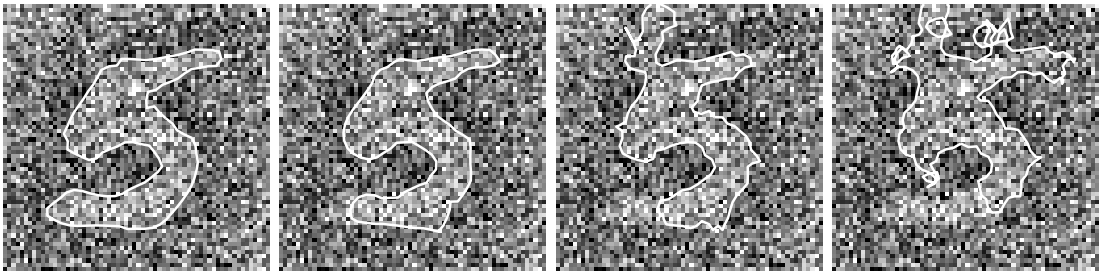
Figure 3.17: Experiments on medical images

Figure 3.17(a) is the same brain image that we experimented on in Section 3.2.6. We obtain, again, a similar result to the previous one using the second graphical model.

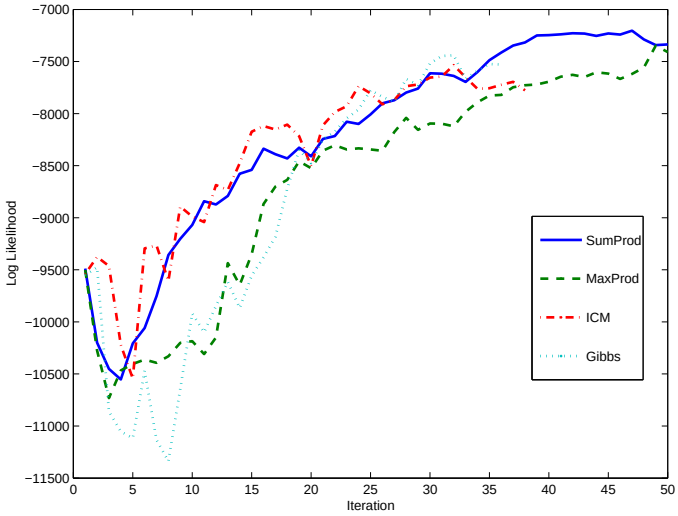
Experiments - Model II: Comparison of different inference methods

We compared the two variants of the BP algorithm (sum-product and max-product, see [95]) with two other similar methods: iterative conditional modes (ICM), and Gibbs sampling.

In all our experiments there was no substantial visible difference between the segmentation results of sum-product and max-product estimates, as exemplified by Figure 3.18(a) and Figure 3.18(b). Comparison of log likelihood profiles (Figure 3.18(e)) during inference revealed small differences – as expected, the sum-product inference outperforms the max-product in



(a) sum-product (b) max-product (c) ICM (d) Gibbs



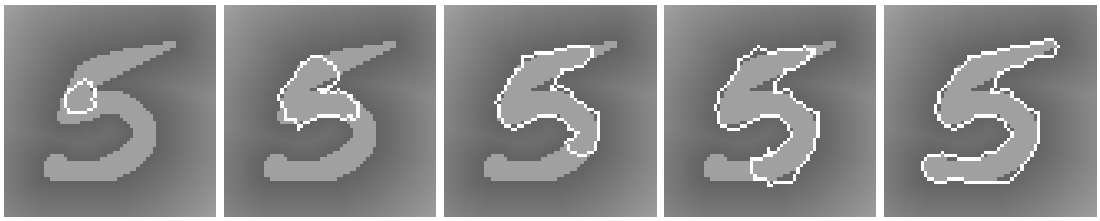
(e) Log-likelihood plot

Figure 3.18: Comparison of different inference methods

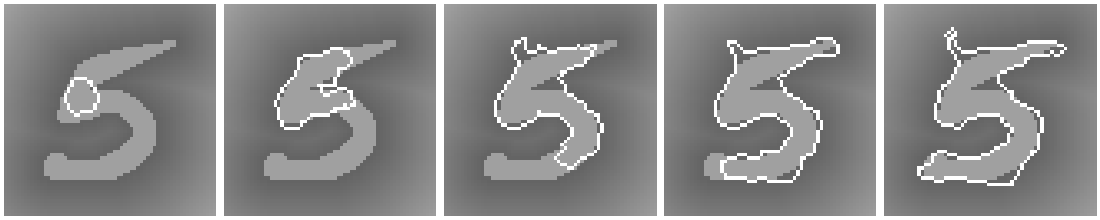
noisy situations. On the other hand, the use of ICM and Gibbs inference resulted in significantly worse final segmentation. For instance, ICM and Gibbs-driven segmentation lead to final estimates shown in Figure 3.18(c) and Figure 3.18(d). Surprisingly, the differences in the log likelihood estimates appear to be less indicative of this final performance. Note that the use of the two approximate inference methods (ICM and Gibbs) is only in the MRF layer. When used in the DM layer, they resulted in very poor segmentation (not shown here). Another difference is the running time, and the BP algorithm took seconds to converge while the other two took minutes without converging.

Experiments - Model II: Comparison of different message passing schedules

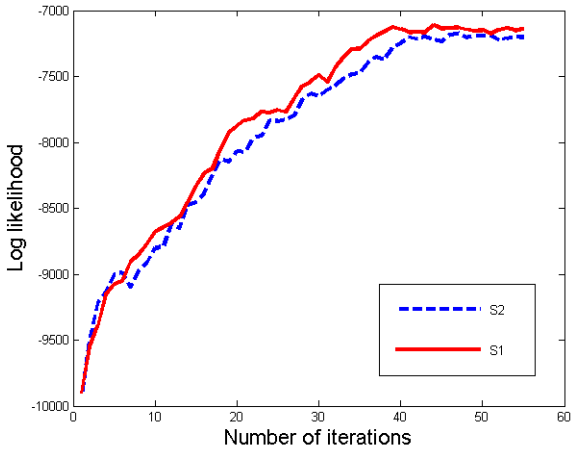
The choice of the message passing schedule in the BP algorithm is an interesting and still open problem. We experimented with two different schedules. The two message passing schedules



(a) $\mathcal{S}_1$



(b) $\mathcal{S}_2$



(c) Log-likelihood plot

Figure 3.19: Comparison of different message passing schedules

were chosen to study the importance of within-model (e.g., inside MRFs and DMs) local consistency and between-models (e.g., between MRF-layer and DM-layer) local consistency. In the first schedule $\mathcal{S}_1$ we first update messages $m_{ij}(x_j)$ in x-layer until convergence (this usually takes two or three iterations in our experiments), and then send messages $m_{ii'}(c'_i)$ once (which is essentially passing message from x-layer to c-layer). Next, we update messages $m_{i'j'}(c'_j)$ in c-layer until convergence (usually in one or two iterations), and finally update messages $m_{i'i}(x_i)$ once, i.e., send messages back from c-layer to x-layer. In the other schedule $\mathcal{S}_2$ we started from the top of the model, update all the messages $m_{ij}(x_j)$, $m_{ii'}(c'_i)$, $m_{i'j'}(c'_j)$, and $m_{i'i}(x_i)$ in this sequence exactly once and repeat, until convergence (Figure 3.13).

The former message passing schedule is intuitively more appealing considering the physical difference of the two models (MRFs and DMs) we are coupling. Traditional energy-based methods also point in the direction of this schedule; integration of forces is usually first computed in the individual horizontal layers. Moreover, $\mathcal{S}_1$ leads to better experimental results. Figure 3.19(a) and Figure 3.19(b) show the performance of the two schedules. The estimates of likelihood resulting from the two schedule, shown in Figure 3.19(c), also indicate that $\mathcal{S}_1$ is preferred to $\mathcal{S}_2$. Note that we did the experiments on Figure 1.1(d) and superimpose the contour on Figure 1.1(a) for visualization purposes.

3.4 Image-based object segmentation: from 2D to 3D

3.4.1 Method

Increasing availability of high-resolution 3D image data using modalities such as magnetic resonance (MR) and computed tomography (CT) has prompted the need for 3D segmentation approaches. However, 3D image segmentation remains an extremely difficult problem, due to the complex topology of 3D objects, the massive data, and demanding computational algorithms. Many 3D approaches are often 2D in nature (i.e., applying the 2D algorithm slice by slice to the 3D volume data [20]). The lack of interaction among individual slice solutions, however, leads to results that are inferior to true 3D-based solutions [22].

In this section, we generalize our framework to 3D image segmentation based on the integration of 3D MRFs and deformable surface models. The proposed method is a true 3D method that fully exploits the structure of the 3D data, resulting in improved object segmentation. The generalization is straightforward using the graphical model representation, and the variational inference in the graphical model also leads to computationally more efficient solutions, which, in the 3D case, is still of main concern.

A 3D MRF model is shown in Figure 3.20. The hidden nodes are positioned at the vertices of a regular 3D grid of the same size as the volume data (Figure 3.20 left). Each hidden node x_i is connected to 6 neighboring hidden nodes (more neighbors can be connected by adding diagonals in the grid) and one observable node y_i (Figure 3.20 right). Again, the observable nodes represent the voxel values of the 3D volume data and the hidden nodes represent the region labels of corresponding voxels.

As to the deformable models, Finite-Element Method (FEM)-based balloon models [22] and Polygonal Geometrically Deformed Model (GDM) [67] are commonly used for representation of 3D surfaces and segmentation of volume data.

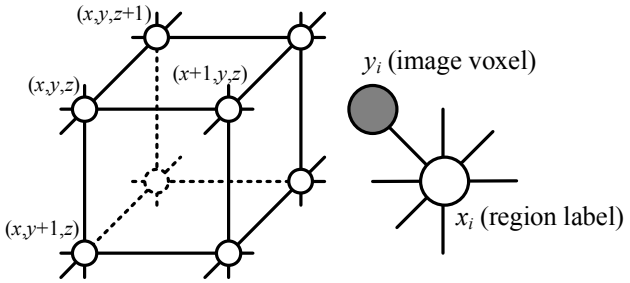


Figure 3.20: 3D MRF model

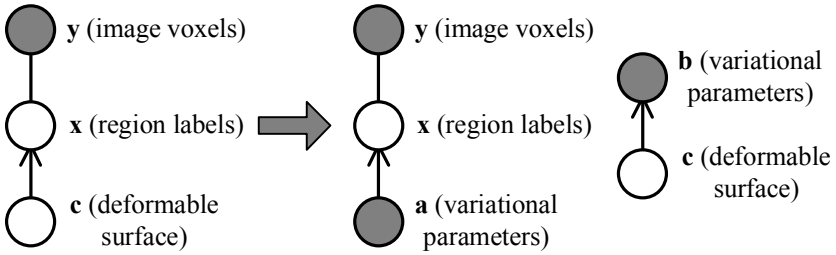


Figure 3.21: 3D graphical model: integrated and decoupled

Similar to the 2D case, a new hidden node representing the underlying boundary surface is added to the 3D MRF model (Figure 3.21 left, only one pair of voxel/label nodes is drawn for simplicity). We again use the structured variational inference technique to seemingly decouple the integrated model into two simpler models (Figure 3.21 right): one extended 3D MRF model with shape prior constraints and one probabilistic deformable surface model.

The 3D algorithm is similar to the 2D one. However, the expansion process of the 3D balloon model far away from the true boundary can be time-consuming and needs frequent reparametrization, and often suffers from local energy minima in noisy images. An interactive initialization procedure or a learned shape prior would be helpful. When there is no shape prior, one can use the 3D MRF segmentation algorithm alone to generate an initial region segmentation and apply the Marching Cubes algorithm [61] to the 3D belief image to generate an initial surface. Marching Cubes is an algorithm for constructing triangle models of constant density surfaces from discrete volume data. The resulting surface representation is suitable for the FEM-based balloon model. The rest of the 3D algorithm is a straightforward generalization of the 2D one.

3.4.2 Experiments - 3D

Experiments - 3D: synthetic images

Our 3D method was also first experimented on a set of synthetic images. The perfect image contains 2 gray levels representing the object (gray level is 160) and the background (gray level is 100) respectively. Gaussian noise with mean 0 and standard deviation 60 is added to the whole image to generate the test image.

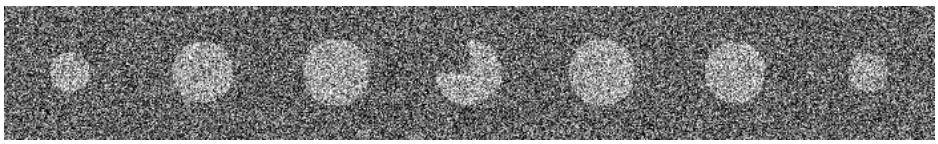
The first experiment intended to show the advantages of the true 3D method over the 2D slice-based method. In this experiment, we generated a $100 \times 100 \times 100$ 3D image containing a ball-like object. Figure 3.22a shows several slices of the perfect image. Our test image is generated by cutting out a part of the object in the middle frame (#50) and adding the Gaussian noise (Figure 3.22b). The segmentation results by 2D MRFs and 3D MRFs are shown in Figure 3.22c and Figure 3.22d. Both models handled noise successfully. The 3D MRF model obviously recovered the missing part of the object in the 50^{th} frame by retaining region smoothness in the direction perpendicular to the frame. Here the natural assumption is that the region smoothness should also be applied to the third dimension of the volume data. The 2D MRF model cannot achieve this due to the lack of interaction between neighboring frames. The boundary of the results from 3D MRFs also look smoother.

The second experiment was performed on a $64 \times 64 \times 64$ volume containing a "5"-like object similar to Figure 1.1(a). The thickness of the object is 8 (i.e., frames 29 to 36 contain the object). Besides the zero mean Gaussian noise, extra noise with mean 160 is also added to a part of the two successive frames 32 and 33. The test image slices are shown in Figure 3.23a. The results of 2D MRFs are shown in Figure 3.23b. Each slice is quite different from others, especially for the two frames with extra noise. The slices in Figure 3.23c (results of 3D MRFs), however, are smoother and similar to their neighbors, except for the first and last frames, which suffered more interference from the background. These two outermost frames are improved by coupling the DM with the 3D MRF model, and other frames are also slightly smoother (Figure 3.23d). The average error rates (misclassified voxels divided by the total volume) of the three methods are 3.98%, 2.79% and 1.62%.

Experiments - 3D: medical images

Experiments with synthetic images in the previous section outlined the advantages of both the 3D method over the 2D method and the hybrid method over the MRF-only method. In this section we show experimental results of applying our methods to 3D medical images. We do not

(a) Perfect image slices (Frame 30, 40, 49, 50, 51, 60, 70)



(b) Noisy corrupted slices

(c) 2D MRF segmentation slices

(d) 3D MRF segmentation slices

Figure 3.22: Experiments on 3D synthetic images (1)

show the results of the slice-based method with 2D MRFs as in previous experiments mainly because this method is sensitive to initialization and we cannot get satisfying results on these medical images. While our 3D method also needs manual initialization when the shape prior is not given, the slice-based method requires manual initialization for almost each single slice.

We first test our algorithms on simulated brain MRI data from BrainWeb [10]. The database contains simulated brain MRI data based on two anatomical models: normal and multiple sclerosis. For both of these, full 3D data volumes have been simulated using three sequences (T1-, T2-, and proton-density (PD)-weighted) and a variety of slice thicknesses, noise levels, and levels of intensity non-uniformity. We segmented the white matter from three different normal brain data volumes using the hybrid method. Figure 3.24(a) shows a slice from the ground truth data of the white matter. Figure 3.24(d) is the result from our hybrid method. Figure 3.24(e) shows the segmentation result on the T1 image without noise and intensity non-uniformity, i.e., RF inhomogeneity) (Figure 3.24(b)). The segmented white matter is slightly thicker than the result from the ground truth, because some of the grey matter is misclassified due to its similar grey value to the white matter. Same misclassification can be observed in Figure 3.24(f), which

(a) Noisy corrupted slices (Frame 29 to 36)

(b) 2D MRF segmentation slices (error = 3.98%)

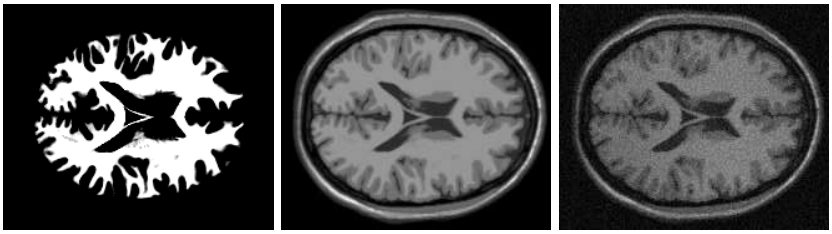
(c) 3D MRF segmentation slices (error = 2.79%)

(d) 3D MRF + DM segmentation slices (error = 1.62%)

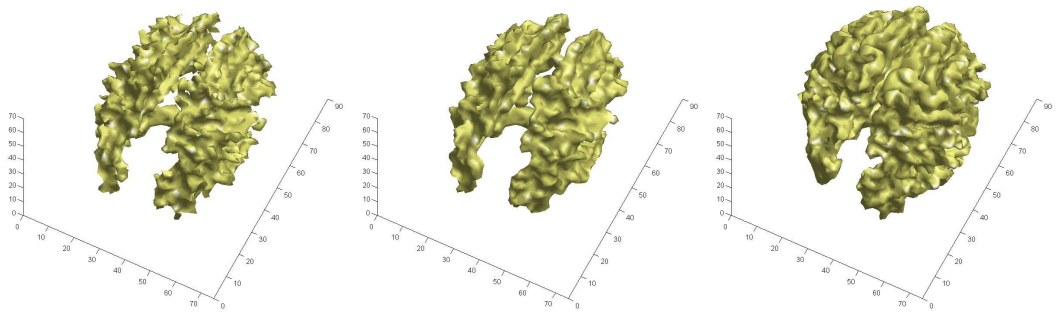
Figure 3.23: Experiments on 3D synthetic images (2)

is the segmentation result on the T1 image with 9% noise and 40% intensity non-uniformity (Figure 3.24(c)). One possible solution to the misclassification problem is using the 3D MRF-only algorithm to do a multi-region segmentation first.

Finally, we show some results on a real medical image [84], which is an MR image of a head with the skull partially removed to reveal the brain. Figure 3.25a is one of the slices from the volume. The results of our methods are shown in Figure 3.25b and Figure 3.25c. To show the difference between the two algorithms (i.e., the effect of adding deformable models), the upper-right parts of Figure 3.25b and Figure 3.25c are magnified in Figure 3.25d and Figure 3.25e. The arrows show that some incorrect patches are eliminated by the deformable fitting process. Surface smoothness can be easily controlled by tuning the parameters in the stiffness matrix. Because the white matter itself is a complicated object with high curvature, the parameters are usually chosen according to experts' opinion.

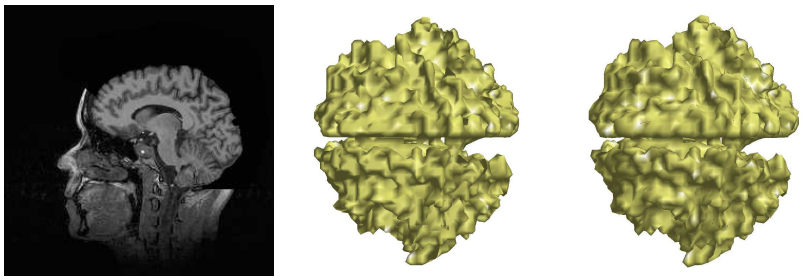


(a) Ground truth slice (b) T1 image slice (c) Noisy T1 image slice

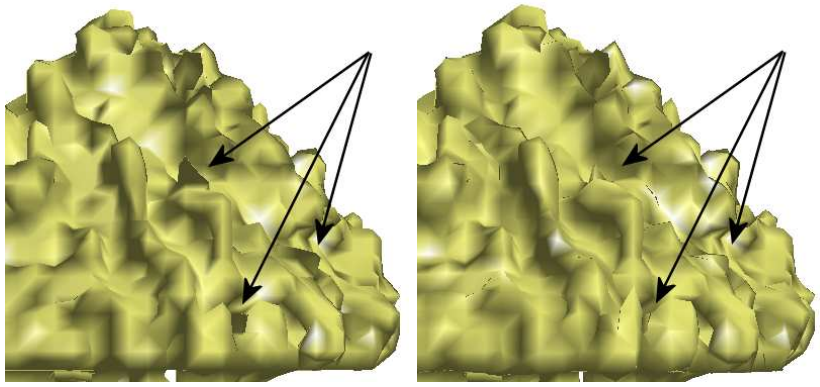


(d) Result on ground truth (e) Result on T1 image (f) Result on noisy T1 image

Figure 3.24: Experiments on 3D medical images (1)



(a) Test image slice (b) 3D MRFs only (c) 3D MRFs + DM



(d) Upper-right part of b (e) Upper-right part of c

Figure 3.25: Experiments on 3D medical images (2)

3.5 Summaries

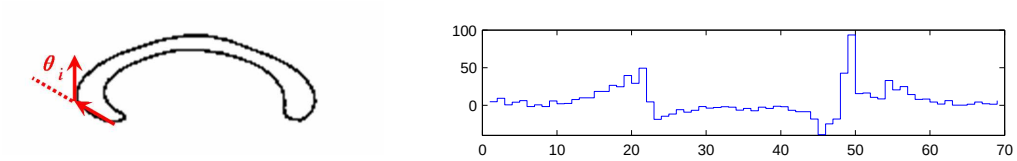
We proposed a new framework to combine the MRF-based and the DM-based segmentation methods. The framework was developed under the auspices of the graphical model theory allowing us to employ a well-founded set of statistical inference and learning techniques. In particular, we developed two solutions to integrate MRFs and DMs. The first model employed the variational inference method, which seemingly decouples the integrated MRF and deformable model. Both components can then be solved by the well-studied algorithms for MRFs and DMs, respectively. This also makes the generalization to 3D applications straightforward. The second model is a fully coupled probabilistic model that allows us to employ an approximate, computationally efficient solution (e.g., the BP algorithm) to the otherwise intractable inference of region boundaries. We have presented two different message passing schedules and pointed to their central role in the segmentation process. We did observe some over-smoothing effect in some of the results, which motivated us to investigate various shape modeling techniques to incorporate shape prior in our segmentation framework. In the next Chapter, we will introduce our work in this direction, and later a more challenging problem, the video-based object segmentation.

Chapter 4
Contour-based Shape Modeling Using Embedded PHMMs

4.1 Introduction

Shape modeling is an important process for many computer vision applications, e.g., image classification, recognition, retrieval, matching, registration, segmentation, etc. Many shape modeling techniques have been developed over the years [60, 91], and it still remains a challenging problem considering all the concerns about robustness, computational efficiency, scalability, interpretability, etc. We started out looking for a proper shape prior model for our segmentation framework and realized after investigating various shape models that there was still improvement to be made. In particular, we are looking for a contour-based shape model for its efficiency and its compatibility with our segmentation methods. Accurate matching of different shapes is most important for us but we also want to develop a general shape model that is as comprehensive as possible. An ideal shape model should be both invariant to global transformations (e.g., translation, rotation, scaling, etc.), and robust to local distortions (e.g., non-rigid transformations, occlusion, missing parts, etc.). Finally a good probabilistic interpretability of the model is also our desire.

In this chapter, we develop a new shape model for 2D shape analysis. A shape instance is described by a curvature-based shape descriptor. A Profile Hidden Markov Model (PHMM) is then built on such descriptors to represent a class of similar shapes. PHMMs are a particular type of Hidden Markov Models (HMMs) with special states and architecture that can tolerate considerable shape contour perturbations, including rigid and non-rigid deformations, occlusions, and missing parts. The sparseness of the PHMM structure provides efficient inference and learning algorithms for shape modeling and analysis. To capture the global characteristics of a class of shapes, the PHMM parameters are further embedded into a subspace that models long term spatial dependencies. The new framework can be applied to a wide range of problems, such as shape matching/registration, classification/recognition, etc. Our experimental results demonstrate the effectiveness and robustness of this new model in these different settings.



(a) Contour (b) Curvature

Figure 4.1: Curvature sequence descriptor

4.2 Low-level shape description

From the model perspective, there are two different levels in shape modeling. Adopting the terminology of [60], we use *shape description* to denote the low-level, numerical feature vector extracted from a given shape instance using a certain method (e.g., a curvature sequence), and *shape representation* the non-numerical, high-level representation of the shape (e.g., a graphical model) which preserves the important characteristics of the shape class. We introduce the shape description part of our model in this section and the shape representation next.

4.2.1 Feature extraction

In this work we employ the *curvature sequence* descriptor. Assuming that the shape contour has been extracted into an ordered list of equally spaced points, the shape can then be described by the sequence of the curvatures computed at these points. To compute the curvature accurately, one may need to upsample the point set by interpolation. A Gaussian filter may be applied to the point coordinates before computing the curvatures to reduce the noise effect. Given three consecutive points $\mathbf{x}_{i-1}$, $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ on the contour, we define $\vec{\mathbf{s}}_{i-1} = \overrightarrow{\mathbf{x}_{i-1}\mathbf{x}_i}$ and $\vec{\mathbf{s}}_i = \overrightarrow{\mathbf{x}_i\mathbf{x}_{i+1}}$, and the bending angle at $\mathbf{x}_i$ which represents the local curvature is

$$\theta_i = \text{sign}(\vec{\mathbf{s}}_{i-1} \times \vec{\mathbf{s}}_i) \arccos\left(\frac{\vec{\mathbf{s}}_{i-1} \cdot \vec{\mathbf{s}}_i}{|\vec{\mathbf{s}}_{i-1}||\vec{\mathbf{s}}_i|}\right) \tag{4.1}$$

Figure 4.1 shows an example of a shape contour and the extracted curvature sequence.

4.2.2 Feature selection

After computing the curvatures, one can downsample a dense curvature sequence to reduce the model complexity. Again note that one should only downsample the dense curvature sequence instead of the sample point sequence before the curvatures are computed, because sparse sample points degrade the accuracy of the curvature computation. A good strategy to downsample the

sequence is to keep all the local extremes in the sequence (because the high curvature parts of the shape contour are usually more informative, i.e., they are the salient features), and then choose equally spaced points in between. The spatially equal-distance sampling is also important to the reconstruction of the shape contour from the curvature sequence descriptor. It should be noted that not all objects have distinguished key points (think of a circle for instance), and using key points alone sacrifices the shape information available in smooth portions of object contours.

4.2.3 Shape reconstruction

One can reconstruct the 2D shape contour from a 1D curvature sequence descriptor. According to the definition of the bending angle (Equation (4.1)),

$$\mathbf{x}_{i+1} - \mathbf{x}_i = \vec{\mathbf{s}}_i = \mathbf{R}_i \vec{\mathbf{s}}_{i-1} = \mathbf{R}_i(\mathbf{x}_i - \mathbf{x}_{i-1}) \tag{4.2}$$

where

$$\mathbf{R}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \tag{4.3}$$

is the rotation matrix. We then have a linear system (for a closed contour)

$$\begin{cases} \mathbf{R}_1\mathbf{x}_n - (\mathbf{R}_1 + \mathbf{I})\mathbf{x}_1 + \mathbf{x}_2 = 0 \\ \mathbf{R}_i\mathbf{x}_{i-1} - (\mathbf{R}_i + \mathbf{I})\mathbf{x}_i + \mathbf{x}_{i+1} = 0, \ i = 2, \cdots, n-1 \\ \mathbf{R}_n\mathbf{x}_{n-1} - (\mathbf{R}_n + \mathbf{I})\mathbf{x}_n + \mathbf{x}_1 = 0 \end{cases} \tag{4.4}$$

Given a set of boundary conditions, e.g., $\mathbf{x}_1 = (x_1, y_1)^{\mathbf{T}}$, $\mathbf{x}_2 = (x_2, y_2)^{\mathbf{T}}$, one can solve the system using the least squares method. Note that the choice of the constants x_1, y_1, x_2, y_2 actually determines the translation, rotation and scaling of the reconstructed shape contour. This shows some attractive properties of the curvature sequence descriptor. First, it is invariant to the object translation. Second, the curvature computed at each contour point is rotationally invariant, so the descriptor is also invariant to the object rotation if the starting point is given. Otherwise, the object rotation causes a circular shift of the curvature sequence, which can be handled by the PHMM-based representation. Finally, the curvature sequence descriptor is not strictly invariant to the object scaling since a change of the contour length usually leads to a change of the curvature sequence length. One possible solution is to normalize all the shape contours to the same length or, equivalently, sample the contours to a fixed number of points.

However, when there are nonrigid or local deformations or missing parts on the contour, the contour length may not be proportional to the actual object scale. Fortunately, PHMMs can also address the scaling problem, as well as the nonrigid deformations and missing contour parts.

4.3 High-level shape representation

A curvature sequence descriptor can capture the characteristic of a given shape instance. However, two similar shapes of the same class can still have quite different curvature sequence descriptors. To model a class of shapes, one needs a higher-level model to take into consider all the variations within the class. As we pointed out in Section 2.3.1, HMMs are an ideal probabilistic sequence modeling method for the shape representation. However most previous HMM-based methods model the feature sequences with ergodic HMMs. As a consequence, several potential problems may arise. First, most of the states may be used to explain multiple observations along the shape profile. This potentially makes shape matching a complex problem. Second, the training procedure in general ergodic models is typically plagued by sensitivity to the model structure selection, model initialization, and local optima of parameter estimation. We show that these problems can be effectively address in the new PHMM framework.

4.3.1 Profile hidden Markov models

PHMMs are a particular type of HMMs well suited for describing general sequence profiles and sequence matching. PHMMs have shown outstanding success in computational molecular biology for modeling of DNA and protein sequences [28, 29].

As shown in Figure 4.2, a PHMM is a left-right HMM with three different types of states:

Match states $M_1, \cdots, M_n$ are regular states of a left-right HMM with emission models $e_{M_i}(O_j)$. Note that the match states can not be revisited, hence each match state is used to explain no more than one observation segment, a *salient* feature of the modeled shape. This significantly simplifies the matching problem.

Insert states $I_0, \cdots, I_n$ are used to model the portions of the observation sequences that do not correspond to any match states in the model (e.g., the stretched parts on the observed shape contour). They have emission distributions $e_{I_i}(O_j)$.

Delete states $D_1, \cdots, D_n$ are used to handle the portions of the model that do not appear in the observation sequences (e.g., occluded or missing parts in the observations). These situations can also be handled by forward jump transitions between non-neighboring match

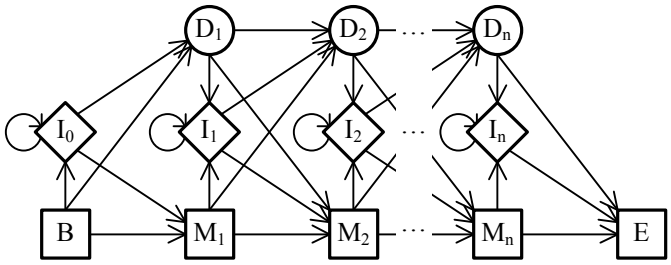


Figure 4.2: Profile hidden Markov model

states. However, to allow for arbitrary deletions the match states need to be completely forward connected. Introducing delete states is an alternative way to model transitions from any match state to any subsequent state with fewer transitions in the model. These states are silent states, which do not emit any observations. Another two silent states $B(egin)$ and $E(nd)$ are introduced for modeling both ends of a sequence.

A profile model may appear more complex than the alternative ergodic model as it typically contains more states (the number of match states is close to the number of salient features in a typical shape instance). However, the transitions of a PHMM are very sparse: there are at most three transitions to and from each state. This significantly reduces the complexity of its algorithms. More importantly, it also addresses a number of problems that the ergodic model may have. For example, with the model structure fixed, the number of states can be easily determined by the number of salient features in the curvature sequence descriptors, and the model parameters are much easier to learn (see Section 4.3.3).

4.3.2 Model inference

Even though PHMMs have different types of states from traditional HMMs, they inherit most of the HMM algorithms [74] with simple adaptations.

Forward algorithm

The *forward* variable of the PHMM is defined as the probability of the partial observation sequence $O_1 \cdots O_j$ and the state X at time j, i.e.,

$$F_X(j) = P(O_1 \cdots O_j, S_j = X|\Theta) \tag{4.5}$$

where Θ are the model parameters. It can be computed inductively:

$$
\begin{aligned}
F_{M_i}(j) &= [F_{M_{i-1}}(j-1)a_{M_{i-1}M_i} + F_{I_{i-1}}(j-1)a_{I_{i-1}M_i} \\
&\quad + F_{D_{i-1}}(j-1)a_{D_{i-1}M_i}]e_{M_i}(O_j) \\
F_{I_i}(j) &= [F_{M_i}(j-1)a_{M_iI_i} + F_{I_i}(j-1)a_{I_iI_i} \\
&\quad + F_{D_i}(j-1)a_{D_iI_i}]e_{I_i}(O_j) \\
F_{D_i}(j) &= F_{M_{i-1}}(j)a_{M_{i-1}D_i} + F_{I_{i-1}}(j)a_{I_{i-1}D_i} \\
&\quad + F_{D_{i-1}}(j)a_{D_{i-1}D_i}
\end{aligned} \tag{4.6}
$$

The forward variable can be used to compute the likelihood of a sequence given the model parameters:

$$
P(O_1 \cdots O_t|\Theta) = \sum\nolimits_X F_X(t) \tag{4.7}
$$

where t is the length of the observation sequence.

Backward algorithm

The *backward* variable is defined as the probability of the partial observation sequence $O_{j+1} \cdots O_t$ given state X at time j, i.e.,

$$
B_X(j) = P(O_{j+1} \cdots O_t|S_j = X, \Theta) \tag{4.8}
$$

It can be computed in the same manner as the forward variable, but in the opposite direction:

$$
\begin{aligned}
B_{M_i}(j) &= a_{M_iM_{i+1}}e_{M_{i+1}}(O_{j+1})B_{M_{i+1}}(j+1) \\
&\quad + a_{M_iI_{i+1}}e_{I_{i+1}}(O_{j+1})B_{I_{i+1}}(j+1) \\
&\quad + a_{M_iD_{i+1}}B_{D_{i+1}}(j) \\
B_{I_i}(j) &= a_{I_iM_{i+1}}e_{M_{i+1}}(O_{j+1})B_{M_{i+1}}(j+1) \\
&\quad + a_{I_iI_{i+1}}e_{I_{i+1}}(O_{j+1})B_{I_{i+1}}(j+1) \\
&\quad + a_{I_iD_{i+1}}B_{D_{i+1}}(j) \\
B_{D_i}(j) &= a_{D_iM_{i+1}}e_{M_{i+1}}(O_{j+1})B_{M_{i+1}}(j+1) \\
&\quad + a_{D_iI_{i+1}}e_{I_{i+1}}(O_{j+1})B_{I_{i+1}}(j+1) \\
&\quad + a_{D_iD_{i+1}}B_{D_{i+1}}(j)
\end{aligned} \tag{4.9}
$$

Combining the forward and backward variables, one can compute the probability of being in state X at time j, given the observation sequence O, i.e., the posterior state distributions:

$$P(S_j = X|O,\Theta) = \frac{F_X(j)B_X(j)}{\sum_X F_X(j)B_X(j)} \tag{4.10}$$

which can be used to measure the certainty of a specific matching.

Viterbi algorithm

Viterbi algorithm has similar recurrence equations to the forward algorithm, but with the sum operation replaced by max operation. This algorithm can be used to find the single best state sequence given the observation and the model parameters, i.e., the optimal state sequence $\arg\max_S P(S|O,\Theta)$:

$$\begin{aligned}
V_{M_i}(j) &= e_{M_i}(O_j) \times \max \begin{cases} V_{M_{i-1}}(j-1)a_{M_{i-1}M_i} \\ V_{I_{i-1}}(j-1)a_{I_{i-1}M_i} \\ V_{D_{i-1}}(j-1)a_{D_{i-1}M_i} \end{cases} \\
V_{I_i}(j) &= e_{I_i}(O_j) \times \max \begin{cases} V_{M_i}(j-1)a_{M_iI_i} \\ V_{I_i}(j-1)a_{I_iI_i} \\ V_{D_i}(j-1)a_{D_iI_i} \end{cases} \\
V_{D_i}(j) &= \max \begin{cases} V_{M_{i-1}}(j)a_{M_{i-1}D_i} \\ V_{I_{i-1}}(j)a_{I_{i-1}D_i} \\ V_{D_{i-1}}(j)a_{D_{i-1}D_i} \end{cases}
\end{aligned} \tag{4.11}$$

where $V_X(j)$ is the highest probability of the partial observation sequence $O_1, \cdots, O_j$, along a single path, ending at state X at time j. One can then trace backwards to find the optimal state sequence using these Viterbi variables.

It is important to note that the computational complexity of all three algorithms (i.e., Forward, Backward, and Viterbi) is only $O(nt)$ time (in contrast to $O(n^2t)$ of ergodic HMMs) and $O(nt)$ space for a model of n states and an observation sequence of length t. This may lead to significant computational savings when dealing with complex shapes. In practice, most of these algorithms take seconds to run on a normal PC, with a magnitude of 10^2 for n and t.

4.3.3 Model learning

As mentioned above, the building of the PHMM is much easier than that of the ergodic HMM. The model structure is fixed, we only need to specify the number of the match states, which can

be either a fixed number specified by the user or the number of the salient features in a typical training sequence (the rest of the features will be modeled by the insert states). One can also use more states for better performance, since the complexity of most algorithms only increase linearly. The model parameters are simple to estimate and the variance in their estimates does not significantly impact the model performance [28].

In our implementation, we consider a homogeneous transition model:

$$\begin{aligned} a_{X_i M_{i+1}} &= \alpha \\ a_{X_i I_i} &= \beta \\ a_{X_i D_{i+1}} &= \gamma = 1 - \alpha - \beta \end{aligned} \tag{4.12}$$

where $i = 1, \cdots, n, X \in \{M, I, D\}$, and α usually dominates β and γ, signifying the importance of match states for modeling the shape. One typical choice for the observation models $e_{M_i}(O_j)$ and $e_{I_i}(O_j)$ are Gaussian models. Insert states use a single zero-mean Gaussian model as they are usually used to model the smoothly stretched contour parts. This ultimately results in the small set of parameters, $\Theta = (\mu_1, \sigma_1, \cdots, \mu_n, \sigma_n, \sigma_I)$.

The PHMM parameters Θ can be estimated in the traditional EM formalism without the need for labeled state correspondences. However, training a PHMM from multiple initially unaligned shape sequences is a difficult problem, usually tackled with local optimizers [29]. Fortunately, unlike general ergodic models, PHMMs allow a viable strategy starting with a PHMM initialized from a single sequence. This model is subsequently reestimated by matching the remaining training sequences to the initial model, and finally refined with all the aligned sequences.

Estimation of ergodic shape models typically shows significant dependency on initial model estimates. In PHMMs this dependency is reduced due to a simpler model structure. Our initialization procedure relies on a set of general steps.

1. Given a curvature sequence, $\theta_1, \cdots, \theta_n$, the sequence may be downsampled to keep mostly the salient features. However, since we start building the model from a single sequence, we usually keep all to be match states.
2. The n match states in the PHMM are assigned Gaussian emission models $e_{M_i}(O_j) = N(O_j; \theta_i, \sigma_i)$. σ_i can be initialized uniformly to some constant, or more specifically according to our knowledge about the deformation capability of different parts of the contour. These parameters can easily control the flexibility of the model.

3. The insert states also have Gaussian emission distributions $e_{I_i}(O_j) = N(O_j; 0, \sigma_I)$. The zero mean suggests that the insert states are simply an extension of the current contour, which is useful for modeling the scaling effects and stretched shape parts. σ_I is chosen to control the rigidity of such extensions, and usually smaller than σ_i.

Once the initial model is built, the remaining training sequences are aligned to it, and the model parameters are fitted to the data. In this way, we avoid labeling the training data.

4.3.4 Model embedding

The regular PHMM is both efficient and effective for many shape analysis tasks, as we show in Section 4.4. However, inconsistent shape matching may occur because of the *lack* of global shape constraints in PHMMs. Such constraints impose global dependencies between spatially distant parameters (e.g., match states) in an otherwise local model.

One way to impose global constraints is to embed the PHMM parameters Θ into a lower dimensional subspace that spans the range of Θ. While there are numerous ways to achieve this embedding, both linear and nonlinear, we here adopt the embedding via Probabilistic Principal Component Analysis (PPCA). PPCA formalism matches our probabilistic formulation and is typically a first stage of many nonlinear embedding techniques, such as the nonlinear or kernel PCA. Unlike the nonlinear methods, PPCA is efficient and often produces satisfactory results in practice.

Let $\Theta = (\mu_1, \cdots, \mu_n)^{\mathbf{T}}$ be the vector of the model parameters whose embedding we seek, then

$$P(\Theta|h) = N(\Theta; Wh, \delta\mathbf{I}) \tag{4.13}$$

where W[1] is the principal component matrix, h is the principal factor, and $\delta\mathbf{I}$ is the covariance of the noise model. It is common to assume a prior distribution over the latent variable h, e.g., a Gaussian prior

$$P(h) = N(h; 0, \lambda\mathbf{I}) \tag{4.14}$$

where $\lambda\mathbf{I}$ is also learned from the training data.

The latent factors h correlate the otherwise, in the regular PHMM, uncorrelated match states. The complete model can now be expressed as

[1] W includes the offset term $\bar{h}$.

$$P(O, S, h|W) = \int_{\Theta} P(O|S, \Theta)P(S|\Theta)P(\Theta|W, h)P(h)d\Theta \tag{4.15}$$

This Embedded PHMM (EPHMM) is now parameterized by a new set of global shape parameters W, λ. Latent factor h represents the global deformation of the shape and needs to be estimated during the inference stage, in addition to hidden states S. We accomplish this using the following coordinate ascent fixed point equations that amounts to an approximate MAP inference:

$$\begin{aligned} S^* &= \arg\max_S P(O, S|\Theta^*, h^*, W) \\ (h^*, \Theta^*) &= \arg\max_{h,\Theta} P(O, S^*|\Theta, h, W) \end{aligned} \tag{4.16}$$

The first task is the PHMM Viterbi algorithm. The second is the PPCA inference (solved in the manner similar to active shape models). Given the above inference procedure, estimation of the embedding matrix W is carried out in the standard EM framework. λ is typically treated as a hyperparameter.

4.4 Applications and results

In this section, we show how our new framework can be applied to shape classification and matching. We demonstrated its effectiveness and robustness on several data sets. Note that in all the experiments, besides the parameters learned from the data, we used the same set of values for all the other parameters (e.g., $\alpha = .998, \beta = .001, \gamma = .001, \sigma_I = 1$, etc.). The number of samples in each sequence is usually controlled to be around 100.

4.4.1 Shape rotation (starting point) detection

As we mentioned before, the curvature sequence descriptor is not strictly rotation invariant when the starting point is not given. In most cases, we start extracting features from the leftmost point on the boundary and following the boundary in a clockwise manner, so the object rotation leads to a circular shift of the curvature sequence. Given model Θ and observation O, the starting point can be computed as follows, $j^* = \arg\max_j P(O_j O_{j+1} \cdots O_t O_1 O_2 \cdots O_{j-1}|\Theta)$. The brute force approach needs $O(nt^2)$ time to evaluate the likelihood of all the t sequences starting from $O_1, \cdots, O_t$ respectively using the Forward algorithm.

One approximate but efficient way of accomplishing the same task is to modify the model parameters involving the states I_0 and I_n with broad distributions of contour features (i.e., to

allow higher variances), which then act like two "don't-care" states. The Viterbi algorithm is then run on the sequence (O, O), a twice concatenated original sequence. Ideally, I_0 and I_n will absorb most of the repeated observations from the two ends, and the sub-sequence started from the actual starting point will be matched to the original model. In this manner we may reduce the complexity of starting point detection to $O(nt)$ in many cases.

We subsequently assume that the curvature sequences have been obtained and if necessary, downsampled according to Section 4.2.2 and aligned to the same starting point using one of the above methods.

4.4.2 Shape similarity measure

Similarity measure is fundamental to many pattern recognition problems. In this section, we define the shape similarity measure based on our framework, and apply it to several different problems. The similarity score between two shapes is defined as:

$$\begin{aligned} P(O^1, O^2) &= \sum_{\Theta} P(O^1|\Theta)P(O^2|\Theta)P(\Theta) \\ &\approx P(O^1|\Theta_1^*)P(O^2|\Theta_1^*)P(\Theta_1^*) + \\ &\quad P(O^1|\Theta_2^*)P(O^2|\Theta_2^*)P(\Theta_2^*) \end{aligned} \tag{4.17}$$

where $\Theta_i^* = \arg\max_\Theta P(\Theta|O^i) = \arg\max_\Theta P(O^i|\Theta)$ for uniformative model priors. This is solved by the Forward algorithm.

We first test this measure on a corpus callosum data set with the image query task. The data set contains 65 corpus callosum images [13, 75, 86]. Figure 4.3 shows a real corpus callosum image and five extracted contour images from the data set. Figure 4.4 shows the results of three different image queries in rows. In each of the three rows, the first image is the query image. We then show the three most *similar* and three most *dissimilar* images from the whole data set found by our algorithm (similarities decrease from left to right in each row).

We also performed classification on the shape data set created by Sebastian et al. [79], which consists of 9 classes of objects, each having 11 images, bearing all the variances we mentioned. Figure 4.5 shows some examples from the data set (top row: one shape from each class; bottom row: all the shapes in the class "hand"). In the most straightforward fashion, we measured the distance between each pair of shapes, and then used the nearest neighbor classifier and leave-one-out strategy to achieve a 100% classification rate. This is not trivial considering the large in-class variance and a general set of parameters were used for all the images. Instead of

Figure 4.3: Corpus callosum shape data set

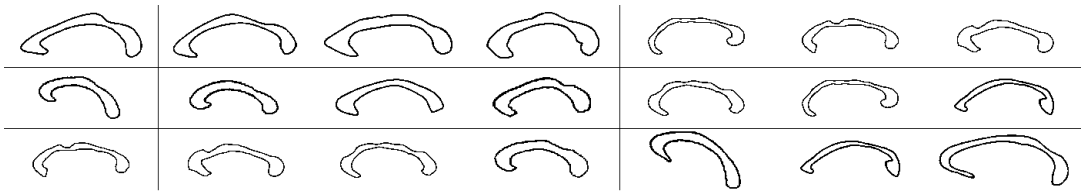


Figure 4.4: Shape similarity measure

computing distance between every pair of shapes, one can build a PHMM for each class, and then evaluate the likelihood of the test image given models of different classes. This usually requires more training images. More sophisticated methods can be used to build the PHMMs, e.g., [89], to further improve the classification rate.

4.4.3 Shape matching

Shape similarity measure is mainly related to a basic problem for HMMs, i.e., evaluating the likelihood of the observation sequence given the model. On the other hand, shape matching corresponds to another basic problem, finding the optimal explanation of the observation sequence. The input to the shape matching algorithm is two curvature sequences O^1 and O^2, and the output is the point correspondence.

First we build the PHMM model Θ of sequence O^1 (also known as the target) using the method described in Section 4.3.3. We then compute the optimal state path of the second sequence O^2 (the source) given this model as $S^* = \arg\max_S P(O^2, S|\Theta)$. Here S denotes the sequence of states under model Θ and it depicts an optimal correspondence between the two sequences. This is solved by the Viterbi algorithm straightforwardly.

Robustness

We first investigate the robustness of our algorithm to global transformations. Since the curvature sequence descriptor does not encode the location information, it is invariant to transition. It is not invariant to rotation unless the starting point is given. In the first experiment, we test the robustness of our algorithm to rotation. We rotate Figure 4.6(a) by 30, 60, ..., 330 degrees and then match the resulted shapes (Figure 4.6(b) to Figure 4.6(l)) to the original shape (Figure 4.6(a)). Each shape is sampled from the leftmost point (i.e., the starting point

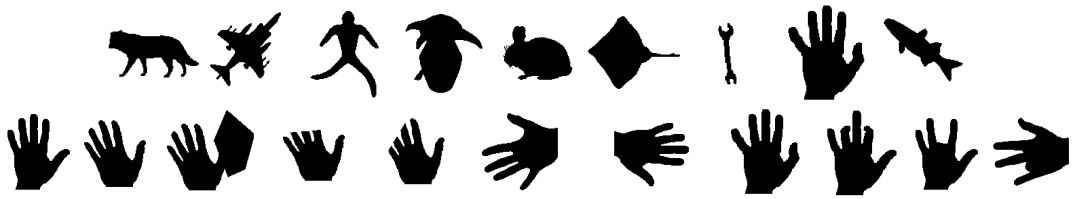


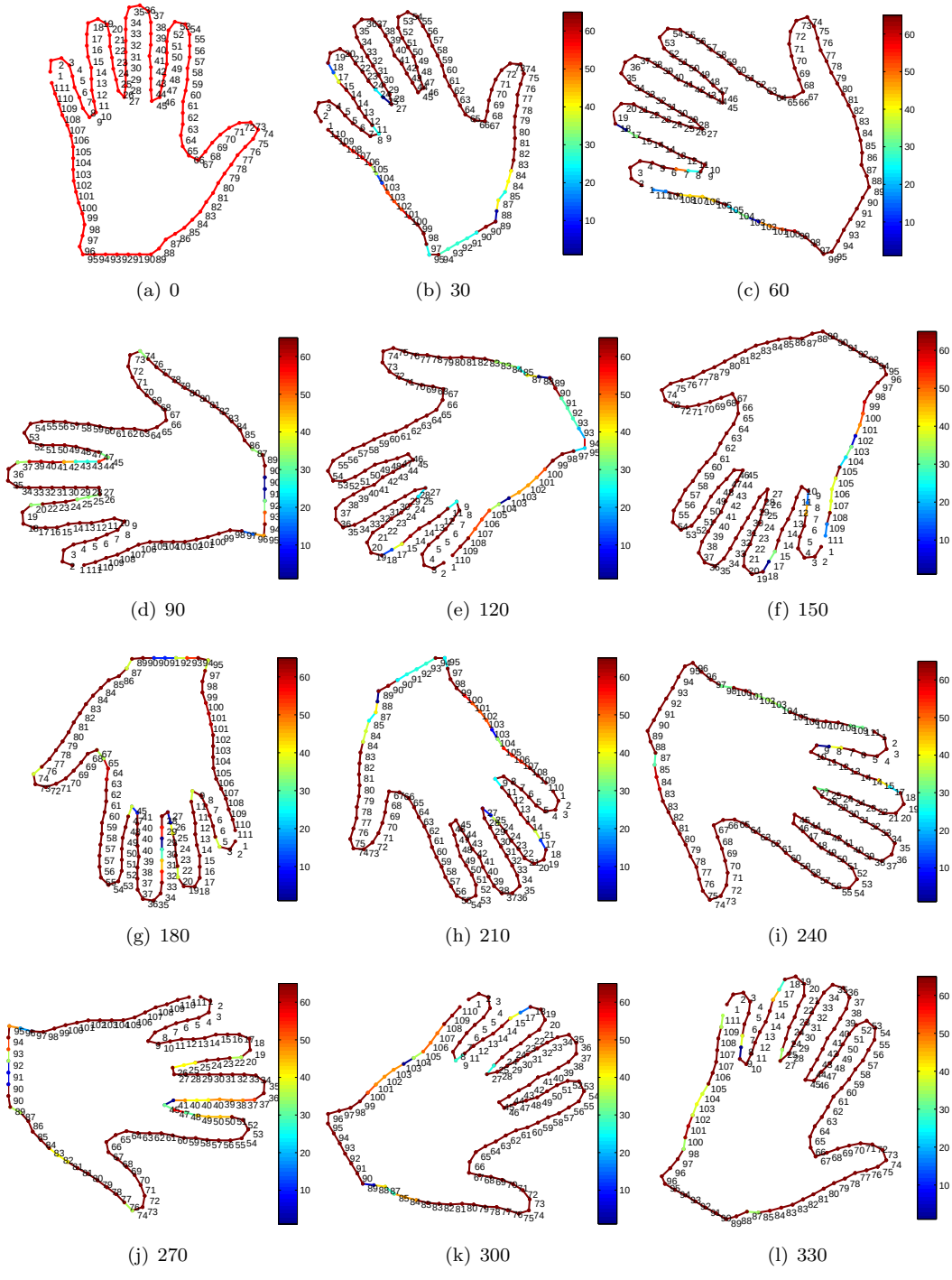
Figure 4.5: Sebastian shape data set

is not given, but detected automatically using the algorithm in Section 4.4.1). The numbers along the shape contour indicate the correspondences to the first shape. The color indicate the matching certainty for each observation computed according to Equation (4.10). In very rare cases some observations are not correctly matched, e.g., there are two consecutive observations are matched to state 5 in Figure 4.6(b). More precisely, the first 5 means *match* state 5 and the second means *insert* state 5. This mostly happens when the observation point is on a nearly straight line and is misclassified as an insert state. Also the matching certainty measure is often lower along the straight lines where the curvature features are not *salient*.

In the second experiment, we test the robustness of our algorithm to scaling. We resize Figure 4.7(a) by factors of s = 2, 3/2, 2/3, and 1/2, and then match the resulted shapes (Figure 4.7(b) to Figure 4.7(i)) to the original shape (Figure 4.7(a)). There are two ways to handle scaling, either using the curvature sequence descriptions or using the PHMM representation. Given two shapes with different scales, the first way is to sample the two shapes at different step-sizes to get the curvature sequences with roughly the same length (as shown in Figure 4.7(b), Figure 4.7(d), Figure 4.7(f), and Figure 4.7(h)). The second way is to sample at the same step-size and let the *insert* and the *delete* states in the PHMM to handle the scaling (as shown in Figure 4.7(c), Figure 4.7(e), Figure 4.7(g), and Figure 4.7(i)). Both strategy work well. Again, most salient feathers are matched perfectly, and the only ambiguity occurs when there are nearly straight lines along the shape contour.

The third experiment shows the robustness of our algorithm to local deformations. We take the class of 11 hand shapes in Sebastian's shape database [79], use the first shape (a normal hand shape) as the target and match all the other 10 shapes to it (Figure 4.8). The only problematic result is the third one because the large extra part on the shape contour. Surprisingly, this mismatch does not affect the classification results (recall the 100% classification rate we obtained on this database) since this shape is still closer to its own class than the other classes in the database. This also explains why the traditional ergodic HMMs were successful for shape classification tasks, even though they are not suitable for shape matching.

Figure 4.9 is another example from this database where we match two shapes from two



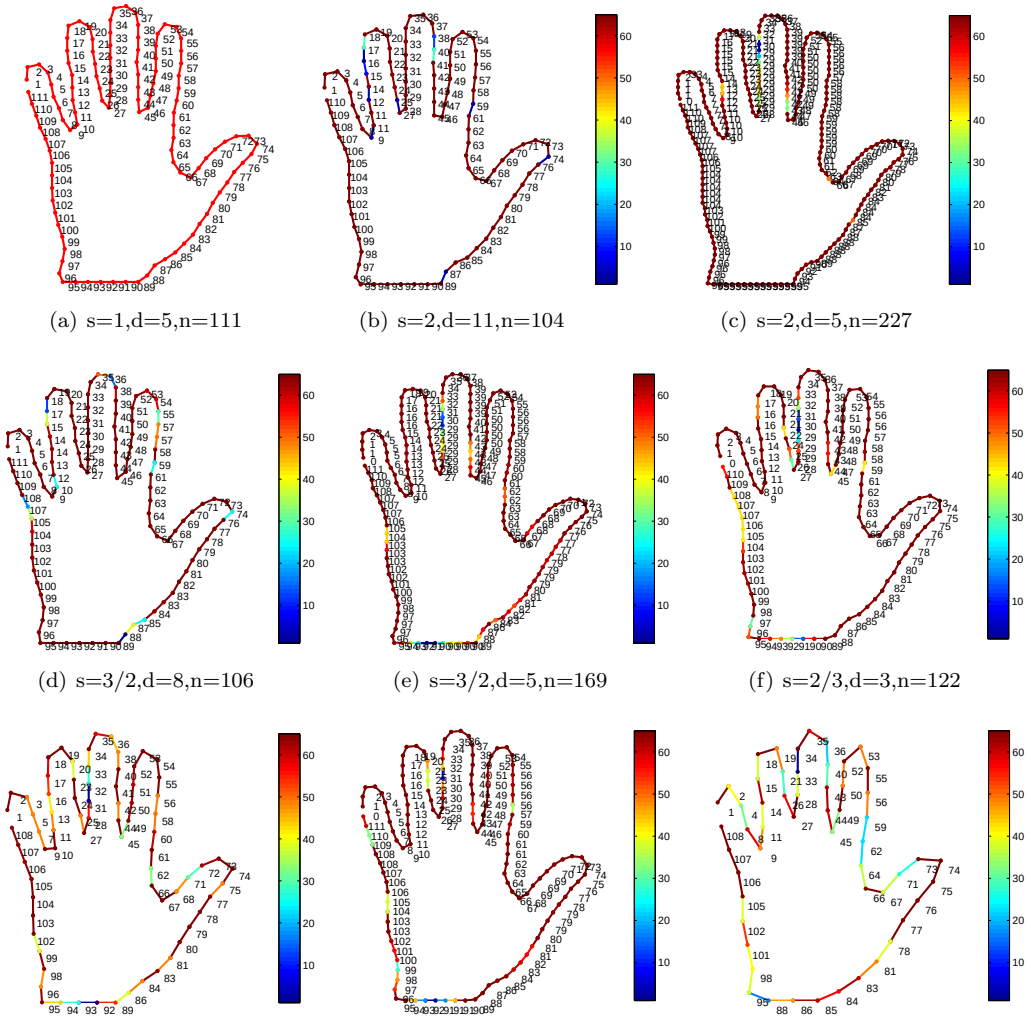
(a) 0 (b) 30 (c) 60

(d) 90 (e) 120 (f) 150

(g) 180 (h) 210 (i) 240

(j) 270 (k) 300 (l) 330

Figure 4.6: Robustness to rotation



(a) s=1,d=5,n=111 (b) s=2,d=11,n=104 (c) s=2,d=5,n=227

(d) s=3/2,d=8,n=106 (e) s=3/2,d=5,n=169 (f) s=2/3,d=3,n=122

(g) s=2/3,d=5,n=74 (h) s=1/2,d=2,n=132 (i) s=1/2,d=5,n=54

Figure 4.7: Robustness to scaling

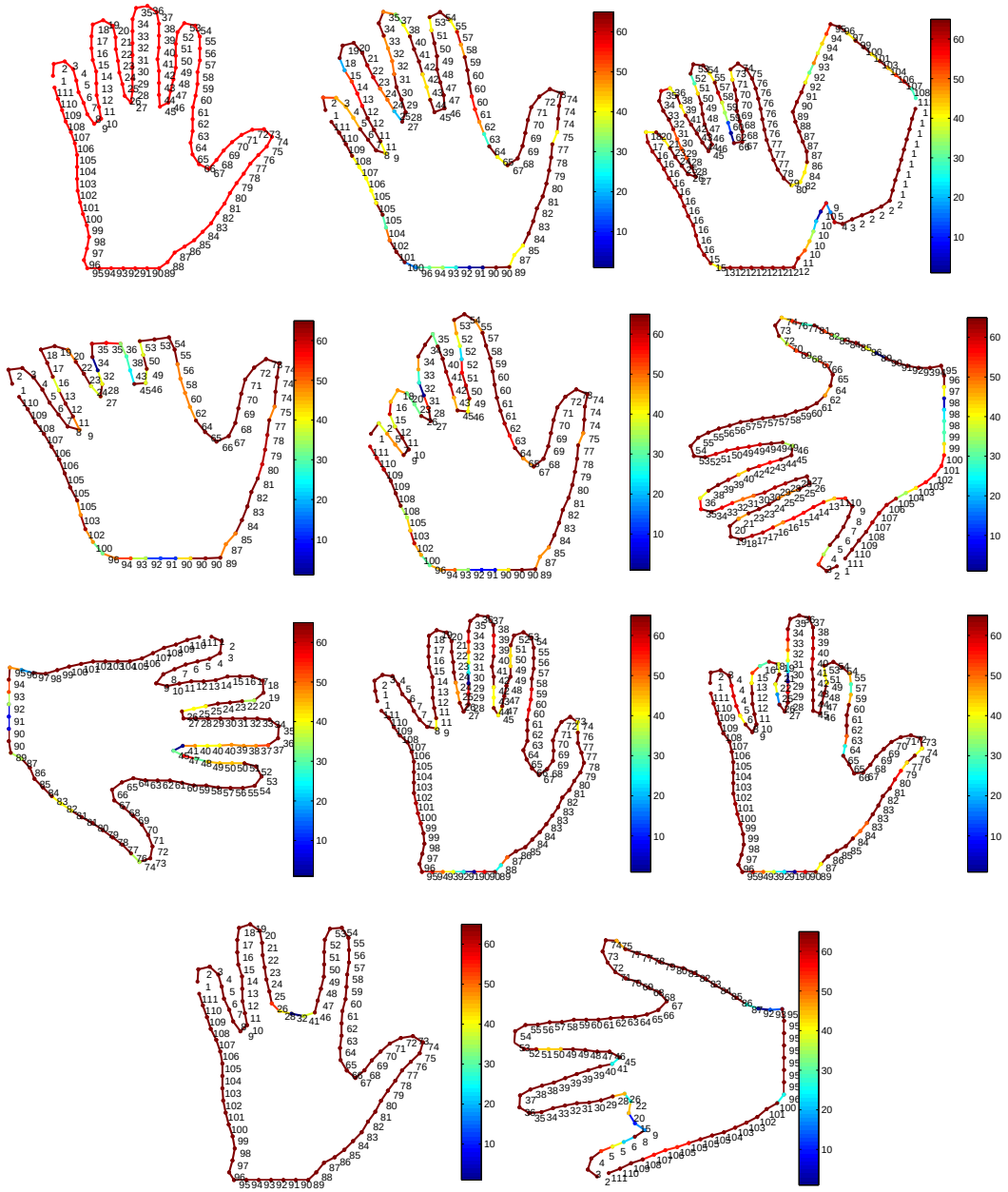


Figure 4.8: Robustness to local deformation

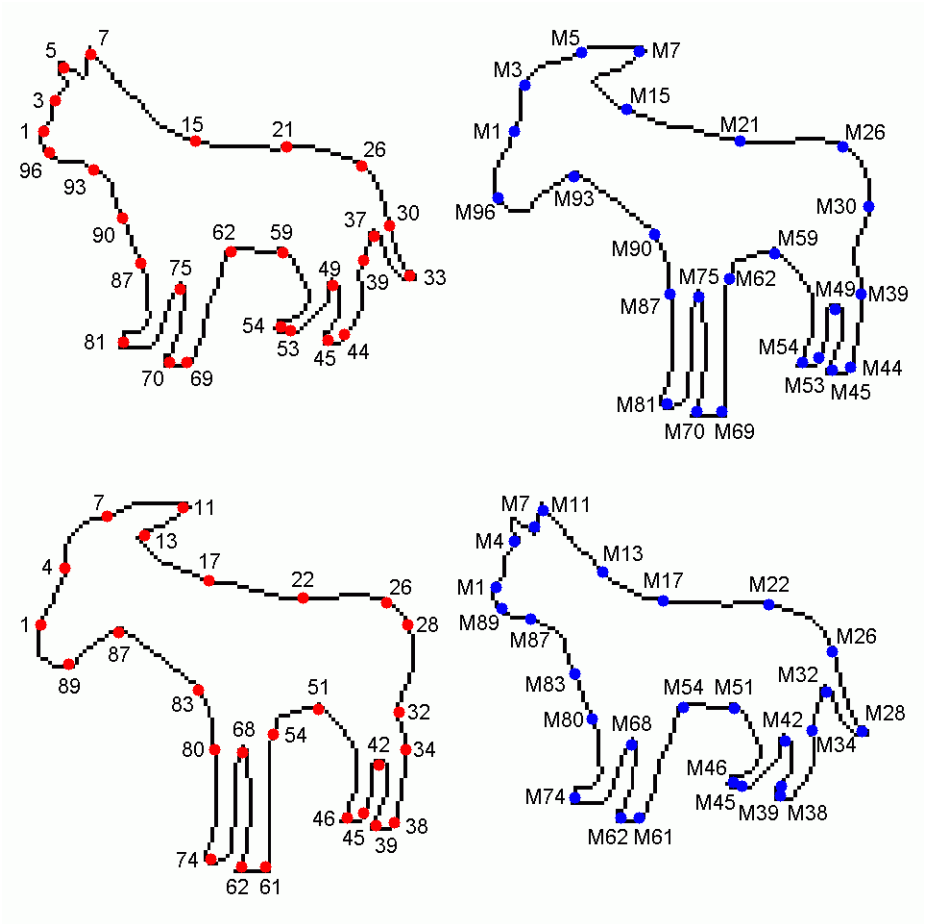


Figure 4.9: Matching of two animal shapes

different objects, which can also be considered non-rigid deformations and missing parts. In the top row, the cat shape is used to build the model and the donkey as the observation, and vice versa in the bottom row. Note the correct correspondence between the labeled points. While the tail and one of the ears of the donkey cannot be seen, they don't affect the correct matching of the rest parts.

Finally we show the robustness of our algorithm to both global transformations (i.e., rotation, scaling) and local deformations (i.e., distortions or missing parts), at the same time. The left image of Figure 4.10 is the shape used to build the model, and the right one is treated as observations. Some representative points are highlighted and labeled similar to previous figures. Both sequences start from the leftmost contour points ("1" and "M44" respectively). The algorithm successfully detected the corresponding start point on the observation. Note the insertions on the index finger (18 observations vs. 8 match states) and the deletions on the third finger, which is missing in the observations (5 observations vs. 16 match states).

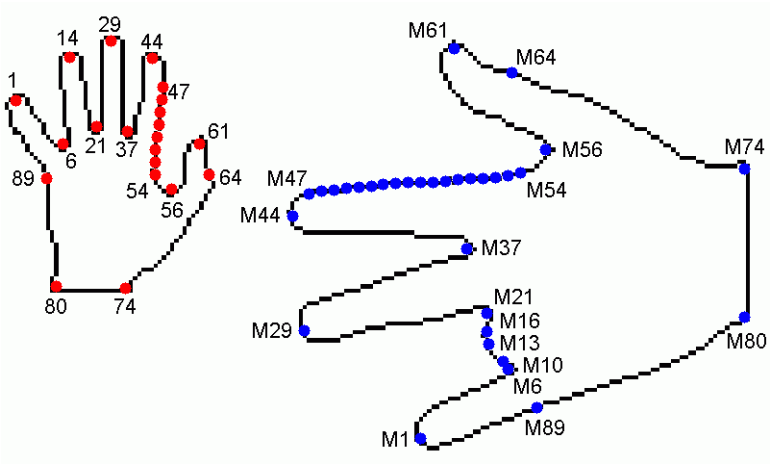


Figure 4.10: Matching of two hand shapes

One of the typical and difficult examples in the corpus callosum data set is shown in Figure 4.11. The upper image is the target shape used to build the model, where the numbers below the (red) points are the match state id's. The middle image is the source shape treated as the observation, where the numbers below the (blue) points indicate which match state this observation is matched to according to our algorithm (the repeated numbers are matched to corresponding insert states). We also show some of the observation id's (after the starting point detection) around the matching labels for reference purposes. Note that the randomly chosen starting point of the observation sequence is not the same as the model starting point, and the lengths of the two sequences are also different. These situations pose both the rotation and scaling problems, which are successfully solved by our algorithm. As for the non-rigid deformations, the splenium (to the right in the figure) of the observation sequence is larger than that of the model sequence, so there are insertions between some of the points (e.g., 24 and 25 are repeated, etc.). On the other hand, the genu of the observation sequence (to the left in the figure) is remarkably smaller than that of the model, where we observed deletions (e.g., 64 jumped to 67, 71 jumped to 75, etc.). In the lower graph of Figure 4.11, we show the matching certainty for each observation computed according to Equation (4.10). One can interpret it as a measurement of how good a specific match is, individually. For example, our algorithm accurately matches O_{83} to M_{64}. However, the local deformation on shape O around O_{83} causes it to be significantly different from the model shape M around M_{64}. The low matching certainty score $P(S_{83} = M_{64}|O, \Theta)$ points to this discrepancy. Similarly, other points of low matching score correspond to changes in local shape O away from the original shape M. This information can be particularly useful for detection of abnormalities in medical imaging applications.

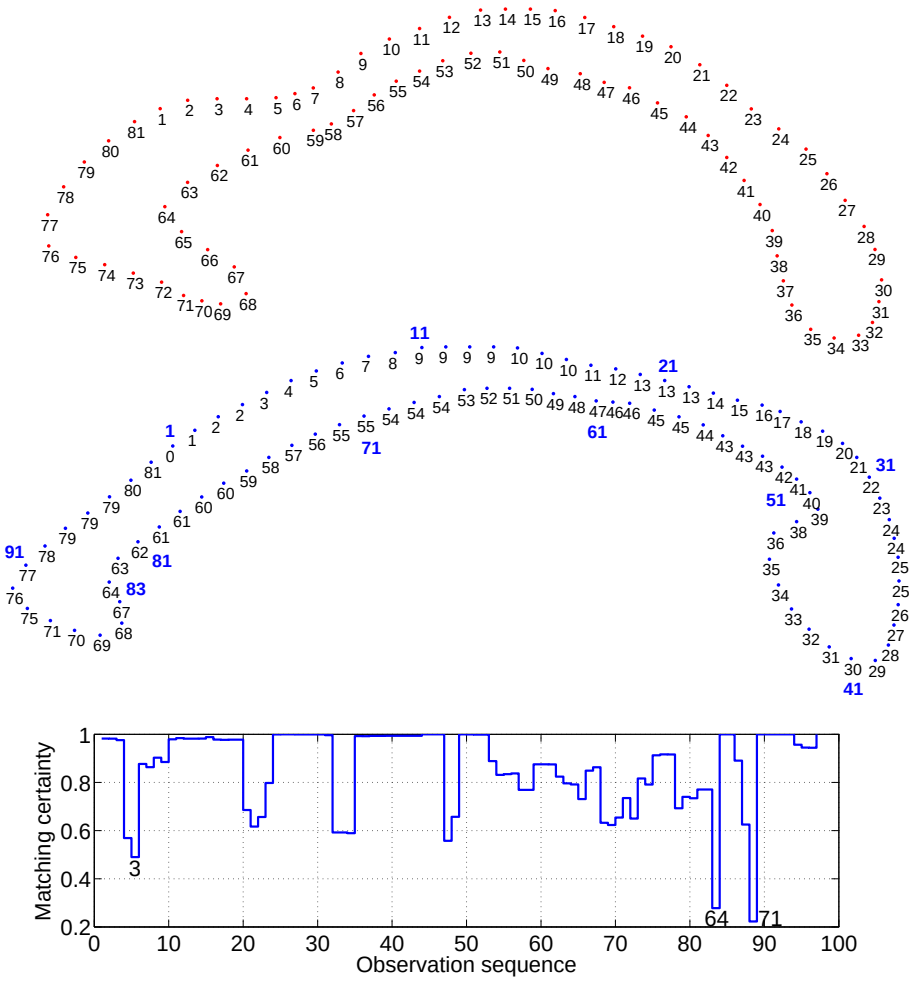


Figure 4.11: Matching of two corpus callosum shapes

The above experiments show the power of the curvature+PHMM-based shape model. However we did observe some mismatching in the "aircraft" class of the Sebastian data set. The reason is that the shape contour of an aircraft often has multiple high curvature parts with similar feature values, separated by nearly straight lines, and distributed evenly along the contour. The regular PHMM may fail due to the strongly ambiguous features. Instead, we trained an embedded PHMM as described in Section 4.3.4 and used the inference procedure outlined in Equation (4.16).

Role of embedding

Figure 4.12 shows examples of the aircraft shapes collected from NASA Dryden online gallery. Note that auto-alignment on this data set is not easy. To avoid manually labeling the data, we used only 25 shapes in the data set that are easy to be aligned automatically. Figure 4.13 shows

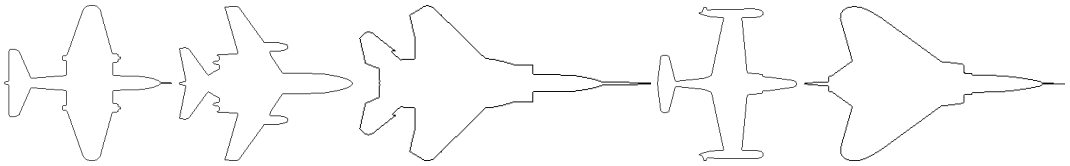


Figure 4.12: NASA Dryden aircraft shape data set

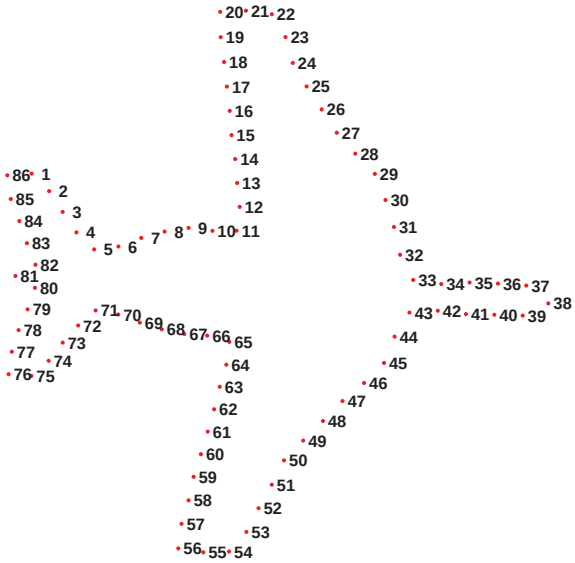


Figure 4.13: Reconstructed mean shape

the mean shape we trained from the 25 shapes according to Section 4.3.3 and Section 4.3.4 and reconstructed according to Section 4.2.3. The upper-left image of Figure 4.14 shows the initial matching using the regular PHMM, where it can hardly handle the strong ambiguities. The upper-right image is the reconstructed shape contour from the matched observations. It is far away from a reasonable shape of an aircraft, indicating a low probability of the latent variable h. After we regularize it using the PPCA, we get a new sets of parameters for the PHMM match states Θ^* (the reconstructed shape contour is shown in lower-right figure, which is reasonably like an aircraft), and rematch the observation sequence to the new adapted PHMM. This procedure usually converges in several iterations in our experiments. The lower-left image of Figure 4.14 is the final matching result. Figure 4.15 shows another example where the constrained PHMM outperforms the regular PHMM. It is worth noting that the seemingly simpler shapes actually causes more ambiguities and are often harder to match than others, and the matching of such shapes can be achieved by matching each of them to the mean shape using the embedded PHMM.

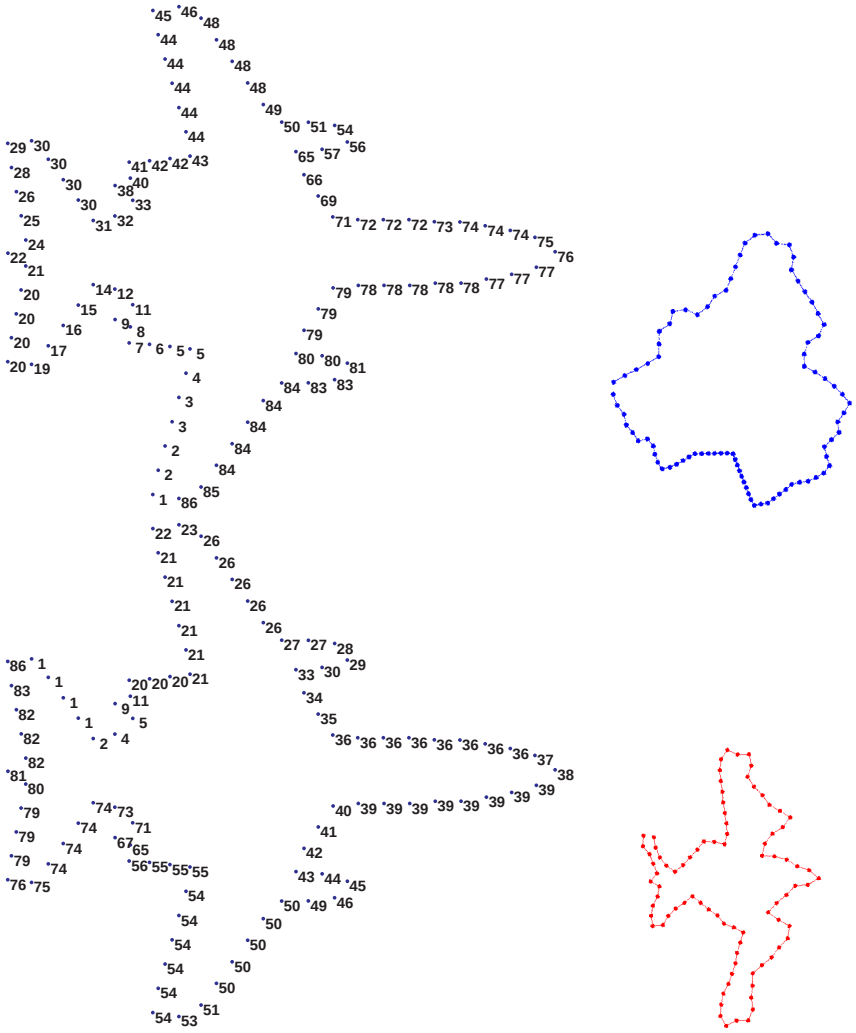


Figure 4.14: Shape matching using embedded PHMMs (1)

4.4.4 Shape segmentation

Another important application of the shape model is that of serving as the shape prior for image segmentation [68, 24]. For example, the traditional deformable model based segmentation often generates oversmooth boundaries, because the global internal energy term:

$$E_{\text{int}}(C) = \sum_i [\alpha_i |P_i - P_{i-1}|^2 / 2h^2 + \beta_i |P_{i-1} - 2P_i + P_{i+1}|^2 / 2h^4] \quad (4.18)$$

imposes homogeneous smoothing over the contour. To capture the high curvature parts of the boundaries, one has to increase the density of the contour points. Another way to solve this problem is to use a shape prior to impose locally different internal energy terms. In the following

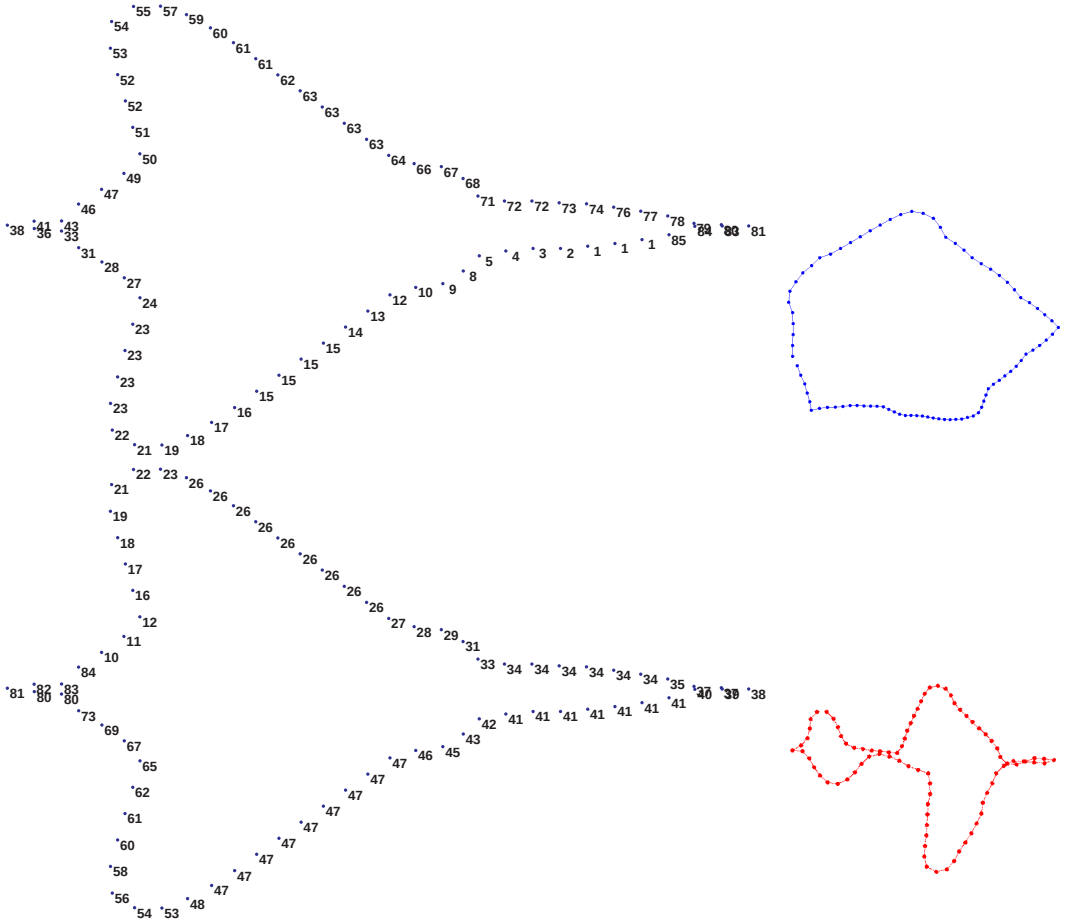
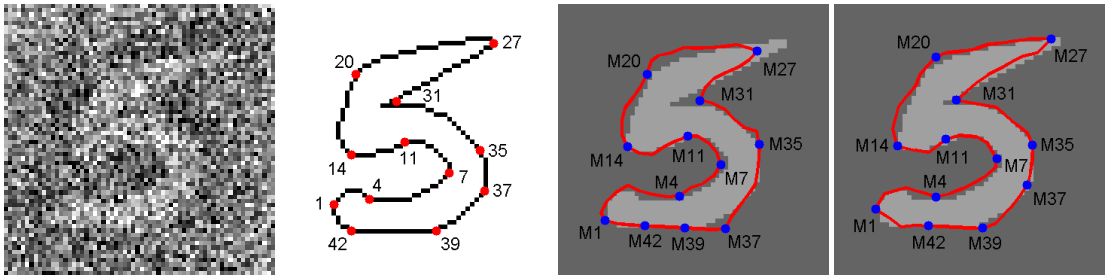


Figure 4.15: Shape matching using embedded PHMMs (2)

experiment, we first use the method presented in Section 3.2 to get an initial segmentation. Then the segmented contour is aligned to a shape prior model. We then replace the original internal energy term with the following:

$$E_{\text{int}}(C) = \sum_i \omega_i |\theta_i - \hat{\theta}_i|^2 \tag{4.19}$$

where θ_i is the bending angle at contour point P_i, while $\hat{\theta}_i$ is the bending angle given by the shape prior model. Once the "standard" internal energy terms is replaced with the one computed using the shape prior, we again run the segmentation algorithm of Section 3.2. This way, the high curvature parts of the contour can be more precisely captured with less contour points. The test image is again the synthetic image Figure 1.1(a) from Section 1.2.1. Note that the shape model is built on a standard shape (Figure 4.16(b)) that is different from the groundtruth of the testing image by a shear transform. Note that the method with shape prior (Figure 4.16(d)) segmented the high curvature contour better than the one without shape prior



(a) Input (b) Prior (c) W/O shape (d) With shape

Figure 4.16: Object segmentation with shape prior

(Figure 4.16(c)) (results are superimposed on ground truth image for clarity).

4.5 Summaries

In this chapter we proposed a new shape modeling framework based on curvature sequence descriptors and profile hidden Markov models. The curvature sequence descriptor is invariant to the object translation, rotation (if the starting point is given). The PHMM representation can address the starting point detection and the scaling problem, as well as the nonrigid deformations and missing contour parts. The structure and sparseness of PHMMs allows for a set of computationally efficient algorithms to be developed for multiple shape analysis tasks. A embedded PHMM model can further capture the global shape information. We applied this framework to various shape analysis problems and showed its robustness to rigid and non-rigid deformations, occlusions and missing contour parts.

An important application of our model is that of serving as the shape prior for image segmentation. In the following chapter, we employ this model to improve the video-based object segmentation. We are currently exploring different higher dimensional shape features/descriptors and the representations based on random fields, so that in the future we can extend the current framework to 3D shape modeling.

Chapter 5

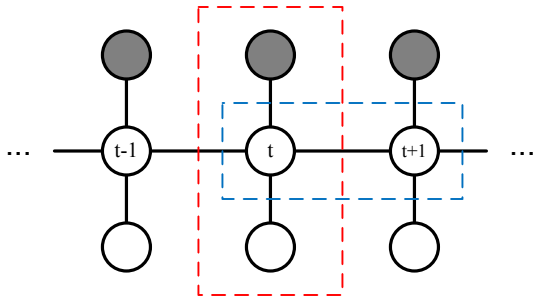
Video-based Object Segmentation Using Graphical Models

5.1 Introduction

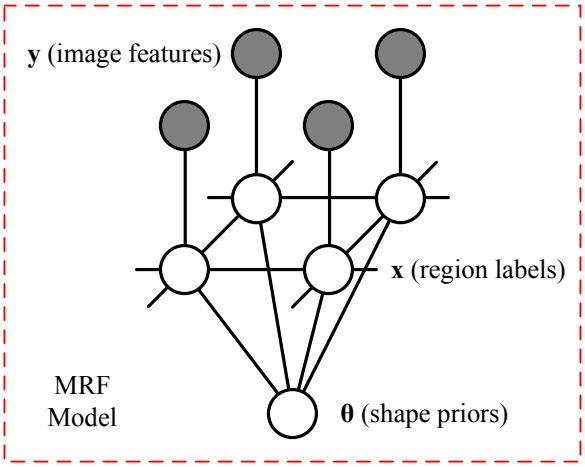
In this chapter we address the video-based object segmentation problem. As stated in Section 1.3.2, simply treating the video data as a stack of independent 2D frames or even an isotropic 3D volume and then directly applying the static image segmentation methods on it is potentially problematic. One obviously needs to utilize the temporal dependencies carried in the video sequence to improve the performance. On the other hand, video-based object segmentation is different from the object tracking or motion detection problem, since more accurate labeling of the related pixels is required. In this chapter we propose a new spatio-temporal MRF framework for video-based object segmentation.

5.2 A new spatio-temporal Markov random field model

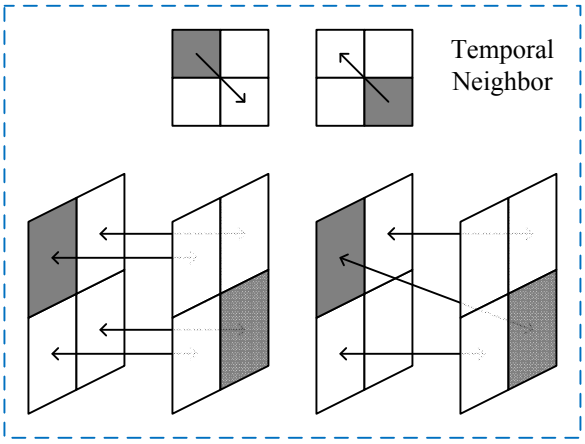
A spatio-temporal MRF model is constructed by stacking the regular MRFs that are used to model the data at different times to form a one dimensional higher MRF model (Figure 5.1(a)). In the video-based object segmentation setting, the spatio-temporal MRF model is three dimensional, though it is possible to have even higher dimensional models (e.g., a series of 3D volumes forming a 4D model). More specifically, given a video sequence of resolution H by W and length T, the 3D spatio-temporal MRF model can be depicted by a graph that consists of $N = H \times W \times T$ nodes representing a set of random variables $\mathbf{x} = \{x_1, ..., x_N\}$. Each random variable x_i represents the label of the corresponding pixel i in the image sequence, i.e., $x_i \in L$, where L is a set of region labels, e.g., $L = \{foreground, background\}$. Each node is connected to a number of other nodes according to a neighborhood (clique) system. The connections among the nodes depict the probabilistic dependencies among the corresponding random variables, defined by compatibility functions (clique potentials). The observation at each node represents the features (e.g., intensity, color, texture, optical flow, etc.) observed at



(a) Spatio-temporal MRF



(b) 2D model for a single frame



(c) Temporal neighbors (top: the optical flow between two consecutive frames; bottom left: temporal neighbors defined by the regular grid; bottom right: temporal neighbors defined by the optical flow)

Figure 5.1: Spatio-temporal MRF model for video-based object segmentation

the corresponding pixel, denoted by $\mathbf{y} = \{y_1, ..., y_N\}$. Note that feature y_i can be computed not only from the single pixel i, but often from a neighborhood centered at that pixel.

The segmentation problem can be viewed as a problem of inferring the MAP solution of the MRF model:

$$\mathbf{x}_{\text{MAP}} = \arg\max_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}) \tag{5.1}$$

which is equivalent to an optimization problem of minimizing the following energy function:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{(i,j) \in \mathcal{N}} \psi_{ij}(x_i, x_j) \tag{5.2}$$

where $\mathcal{V} = \{1, 2, ..., N\}$ is the pixel/node index set, $\mathcal{N}$ is the edge set among the nodes, and $\phi_i(x_i)$ and $\psi_{ij}(x_i, x_j)$ are the unary (association) and pairwise (interaction) potential functions, respectively. Note that we discarded $\mathbf{y}$ since it is known and can be encoded into the potential functions.

The association potential function $\phi_i(x_i)$ is usually used to model the local information that can be exploited to infer the label x_i. We define this function as the sum of two different terms, corresponding to the bottom-up low-level and top-down high-level information, respectively.

$$\phi_i(x_i) = \phi_i^{low}(x_i) + \phi_i^{high}(x_i) \tag{5.3}$$

where the first term

$$\phi_i^{low}(x_i, y_i, \theta_{low}) = -logP(y_i|x_i, \theta_{low}) \tag{5.4}$$

is the traditional observation model which constrains the label to be consistent with the local low-level features, and the second term

$$\phi_i^{high}(x_i, \theta_{high}) = -logP(x_i|\theta_{high}) \tag{5.5}$$

is the high-level prior term which constrains the label to be consistent with our, usually object-specific, knowledge about this image location. θ_{low} and θ_{high} are the bottom-up and top-down model parameters, respectively (Figure 5.1(b)).

The interaction potential function $\psi_{ij}(x_i, x_j)$ is a prior term used to impose region smoothness in the traditional MRFs. In the spatio-temporal MRFs, though, we argue that the temporal dimension has a different physical meaning from the spatial dimensions, hence the temporal smoothness constraints should also be different. In the image plane, when one has no knowledge

about the locations of the region boundaries, the Ising/Potts model imposes spatially generic region smoothness on the whole image. Along the time dimension, however, the additional dynamic information in the video sequences implies where the discontinuities occur. For example, if one knows a point is moving from location (x, y) to $(x + \delta x, y + \delta y)$ from time t to time $t + 1$, then there is a discontinuity between nodes (x, y, t) and $(x, y, t + 1)$, even though they are a pair of neighbors in the traditional 3D MRF structure. Instead, one should impose region smoothness between nodes (x, y, t) and $(x + \delta x, y + \delta y, t + 1)$ (Figure 5.1(c)). In our framework, we use such a more flexible neighborhood defined by the optical flow, and impose different smoothness constraints for spatial and temporal neighbors.

More precisely, to achieve both spatial and temporal smoothness, we divide the above edge set $\mathcal{N}$ into two subsets of spatial connections $\mathcal{N}_s$ and temporal connections $\mathcal{N}_t$. The sum of the interaction potential functions hence becomes the sum of two different terms, corresponding to the spatial smoothness and temporal smoothness, respectively:

$$\sum_{(i,j)\in\mathcal{N}} \psi_{ij}(x_i, x_j) = \sum_{(i,j)\in\mathcal{N}_s} \psi_{ij}^{spa}(x_i, x_j) + \sum_{(i,j)\in\mathcal{N}_t} \psi_{ij}^{temp}(x_i, x_j) \tag{5.6}$$

$\mathcal{N}_s$ is defined in each frame similarly to the traditional 2D MRFs, i.e., each node is connected to its closest neighbors on the regular 2D grid. The size of the neighborhood is usually determined by different applications.

$$\psi_{ij}^{spa}(x_i, x_j, \theta_{spa}) = \begin{cases} 0 & x_i = x_j \\ \theta_{spa} & x_i \neq x_j \end{cases} \tag{5.7}$$

where θ_{spa} controls the strength of the smoothing effect. $\mathcal{N}_t$, on the other hand, should be defined by proper pixel correspondences in the neighboring frames. Once $\mathcal{N}_t$ is defined, one can use an Ising/Potts model with different parameters for the temporal smoothness.

$$\psi_{ij}^{temp}(x_i, x_j, \theta_{temp}) = \begin{cases} 0 & x_i = x_j \\ \theta_{temp} & x_i \neq x_j \end{cases} \tag{5.8}$$

To summarize, in our framework, Equation (5.2) becomes:

$$\begin{aligned} E(\mathbf{x}, \Theta) &= \sum_{i\in\mathcal{V}} \phi_i^{low}(x_i, y_i, \theta_{low}) + \phi_i^{high}(x_i, \theta_{high}) \\ &+ \sum_{(i,j)\in\mathcal{N}_s} \psi_{ij}^{spa}(x_i, x_j, \theta_{spa}) + \sum_{(i,j)\in\mathcal{N}_t} \psi_{ij}^{temp}(x_i, x_j, \theta_{temp}) \end{aligned} \tag{5.9}$$

where Θ is a collection of the parameters in each individual module.

The exact MAP inference in MRFs is computationally infeasible, and various techniques have been used for approximating the MAP estimation. In our method, we use the LBP algorithm for a few preferable advantages (more discussion later). The MRF model parameters (i.e., the parameters in the potential functions) can be learned using the EM algorithm.

5.3 A practical implementation

In this section, we show an example of how to fully exploit various cues and combine them together using our framework to perform video-based object segmentation. It is extremely difficult, if not impossible, to consider a general object segmentation problem. Even for a small number of objects, it still requires successful object categorization, which is also an open problem. Therefore, we consider a specific task in this study, that is, the segmentation of the pedestrian in video sequences. This is still a very hard problem considering all the different human physiques, gaits, and poses, and we do not assume the training data from a specific subject is available.

5.3.1 Incorporating bottom-up information

The bottom-up information is incorporated by the local observation model Equation (5.4). In this case, we choose to use the normalized RGB color features (i.e., the chromaticity coordinates) suggested by [30], hence the observation $\mathbf{y}$ can be computed:

$$y_i = \begin{pmatrix} R_i/(R_i+G_i+B_i) \\ G_i/(R_i+G_i+B_i) \\ (R_i+G_i+B_i)/3 \end{pmatrix} \tag{5.10}$$

where $R_i, G_i, B_i \in [0,1]$ are the RGB color of pixel i. The foreground $P(y_i|x_i = foreground, \theta_{fgd})$ is modeled with a mixture or Gaussian with 10 components in the normalized RGB space. The background model $P(y_i|x_i = background, \theta_{bgd})$ is a single Gaussian at each background pixel. While there are more sophisticated background models such as the adaptive mixture model [85], the Non-parametric model [30], etc., part of our goal is to show how the simple modules can be combined and improved by our framework, so we opted for the simpler models. The foreground mixture model can be estimated from the initialization and re-estimated along with the segmentation procedure as in [46]. In other words, the pixels that are segmented as the foreground will be used to update the foreground model. A particular advantage of the LBP

algorithm for the MRF-MAP inference is that a belief between 0 and 1, instead of a binary label, is assigned to each variable x_i, which allows one to update the model parameters with soft weights. The background model can be learned from the training data or, if the clean background data is not available, estimated in the same way as the foreground model.

5.3.2 Incorporating top-down information

The top-down information is incorporated by Equation (5.5). In our case, this is a shape prior term. More specifically, this shape prior term is again a softmax function induced by the signed distance of a pixel to a specific a priori shape contour:

$$P(x_i = foreground|\theta_{shape}) = \frac{1}{1 + \exp(-\theta_{mag} \times \theta_{dist}(i))} \tag{5.11}$$

where θ_{mag} controls the magnitude of the shape prior, and θ_{dist} is the signed distance map computed from the contour of the chosen shape prior. The probability maps of the background given the shape prior is simply:

$$P(x_i = background|\theta_{shape}) = 1 - P(x_i = foreground|\theta_{shape}) \tag{5.12}$$

Since we do not assume the training data from a specific subject is available, we generated two sets of silhouette images using the standard male and female figure models in *Poser*® as the training data. Like other model parameters, in most cases, one needs to choose the appropriate shape prior for a particular frame, in other words, estimate the pose at the same time of performing segmentation. A simultaneous segmentation and 3D pose estimation method could be quite complicated [11]. Since we have relatively sparse samples (60 frames for one full walking cycle), we simply use the closest training instance to the current segmentation result as the shape prior for the next iteration of segmentation. However, because the usual distance metrics (e.g., Euclidean distance) in the original image space are not necessarily accurate descriptions of the distance between shapes and are very sensitive to the segmentations, one might need to employ some shape manifold embedding approaches to the image space as in [31] or use an explicit shape matching method, which is what we used in our implementation.

Contour-based shape models are usually both effective and efficient in this scenario. We chose the profile hidden Markov model based shape model [48] due to its efficiency and robustness to missing parts and local distortions. More specifically, from a segmented silhouette image, one can easily obtain the shape contour, which is then compared to the profile models of the prior shape contours. The prior shape contour that is closest to the segmented shape contour is used

to generate the signed distance map θ_{dist}, from which the final probability maps are computed according to Equation (5.11). Since the shape prior images are generated from the standard figure models, and there are always differences in physiques, gaits, and poses between the priors and the actual data, an annealing schedule is applied to the parameter θ_{mag} to weaken the top-down model and allow the bottom-up model to better refine the segmentation results. Note that the bottom-up model is re-estimated and improved over time, so its impact should be strengthen over the top-down model in the later stage of the procedure.

5.3.3 Incorporating spatial constraints

We used the traditional region smoothness term as defined in Equation (5.7). More complicated data-dependent terms have been used in [8, 56], which assigned different θ_{spa} values depend on the image locations and the local features, yet in our case we found the simple term works well enough.

5.3.4 Incorporating temporal constraints

The temporal constraints can be simply defined in the same way as the spatial constrains, with different value of θ_{temp}. One more important and difficult task is the construction of $\mathcal{N}_t$. It has been shown in [58] that the temporal neighbors can be defined by using optical flow algorithms to detect the pixel correspondence in neighboring frames. The problem, however, is that multiple optical flows from different nodes in one frame could point to the same node in the next frame. This causes the unstable structure of the MRF model, i.e., each node in the network could have different numbers of neighbors, which forbids a simple and efficient LBP algorithm. In our implementation, given any two consecutive frames, we compute the optical flows in both directions, i.e., one from frame t to frame $t+1$, and the other from frame $t+1$ to frame t [69]. Only those pairs of nodes that have matched flows are connected as temporal neighbors. Two nodes from neighboring frames at the same image coordinates are also considered temporal neighbors if they both do not have any optical flow, which is useful for imposing smoothness in the static background. Figure 5.1(c) shows a simple example of this strategy. Note that some of the nodes could have no temporal neighbors at all.

5.3.5 Inference using sequential loopy belief propagation

We have argued that a video sequence should be treated as a 3D spatio-temporal data instead of a batch of independent images. However, to process a whole video sequence as a single 3D

volume can also be problematic, especially in computational requirements, considering the time dimension can be virtually infinite. In practice, most of the spatio-temporal frameworks apply a small window on the time dimension, i.e., work on $k(k \geq 2)$ consecutive frames at one time. Most of the time a small window can be used in the image plane as well. So only a small (but still 3D) part of the entire model is being processed at one time. The LBP algorithm is used for the inference in the small chunk of 3D data. In each step, the first frame is considered correctly segmented in the last step and the following frames are segmented, which will then be used as initialization in the next step. With this strategy, one only needs to initialize the very first frame of the video sequence. We usually start with the forward sweep with $k = 2$, i.e., two frames at one time, and after the forward propagation, backward sweep is sometimes performed to get smoother results. The whole procedure is then repeated for a few times till convergence. This procedure is essentially a limited sequential loopy belief propagation on the whole 3D data. One can increase the number k for a more aggressive message passing scheme, which may converge faster but at the cost of more memory and possible suboptimal results.

The whole algorithm is outlined below in Table 5.1

Table 5.1: Video-based object segmentation algorithm

Input: image sequence, shape prior images;
Output: binary segmentation of the object in the sequence;
Algorithm:
Initialization
    Segment the first frame, and initialize the foreground and background models;
    Compute the optical flow, and determine the spatio-temporal MRF structure;
Repeat *#Sweeps* or until convergence
{
    Repeat *#Subsets* (forward sweep)
    {
        estimate shape prior
        Repeat until convergence
        {
            Use the LBP algorithm to solve the MRF-MAP problem of segmentation
            Estimate the shape prior;
        }
    }
    Backward sweep if necessary;
    Update the foreground and background model using the segmentation
}

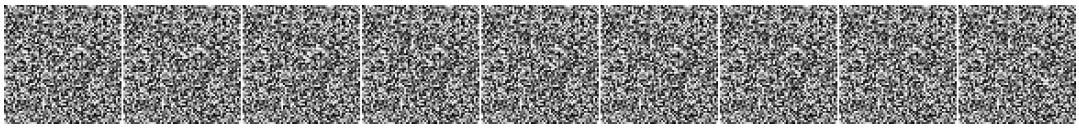
5.4 Experiments

5.4.1 Synthetic data

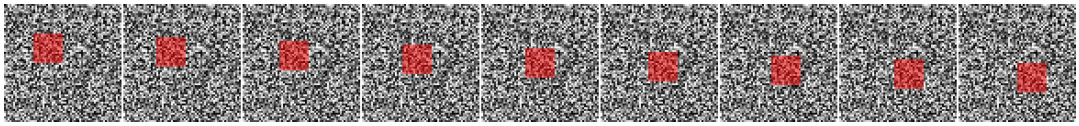
The first experiment is carried out on the synthetic data similar to the one used in [96]. The background is a 64×64 image whose pixels have uniformly distributed intensities between 0 and 1, and the foreground (moving object) is a 16×16 patch generated in the same way as the background. First, this is a perfect example to show the importance of the dynamic information to segmentation in video sequences. It is impossible to detect this type of camouflaged object without going through the time dimension, as shown in Figure 5.2(a), on which we overlaid the groundtruth to obtain Figure 5.2(b). Second, while it might be easy for regular spatio-temporal MRFs based on image differences to recover the moving object in this noisy sequence, it is very hard for most appearance-based methods, as pointed out in [96]. Since the observation model in our framework is based on the foreground/background appearances instead of image differences, it would also be hard for our method if we ignored the temporal constraints. For example, Figure 5.2(c) shows the segmentation results of our model defined on the regular 3D grid structure. Since the background is static, the pixel-based Gaussian background model handles the background fairly well, but the Gaussian mixture foreground model cannot separate the object from the background clearly enough. However, this problem is rectified by the special optical flow induced temporal neighborhood structure of our model, as shown in Figure 5.2(d). We argue that the optical flow induced temporal neighborhood system essentially encoded the image difference information used by traditional spatio-temporal MRF methods such as [62, 96]. On the other and, we even get better results than those in [96] because of the decent appearance model. Note the small holes and rough boundaries in their results, which is acceptable in tracking and motion detection, but not in segmentation.

5.4.2 Real data

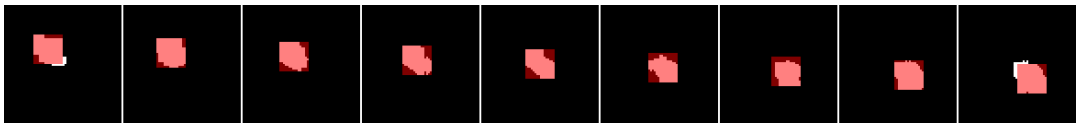
In this experiment, we show the segmentation results on a human walking sequence. Note that the subject is walking in place, so the major part of the upper body is only moving slightly, which makes this sequence very hard for the traditional spatio-temporal MRFs based on the image differences. On the other hand, the moving belt of the treadmill and the highly cluttered background are very hard for simple background modeling methods. To achieve good segmentation performance, one has to combine multiple cues. Figure 5.3(a) shows image frames from the input sequence. Figure 5.3(b) depicts the beliefs from the bottom-up observation



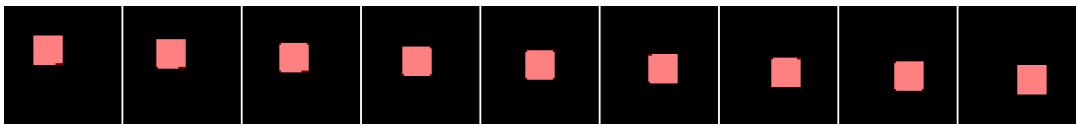
(a) Input sequence



(b) Ground truth



(c) Results from *regular* 3D MRFs with appearance-based observation model (544 pixels misclassified)



(d) Results from our model with reliable temporal constraints induced by optical flow (32 pixels misclassified)

Figure 5.2: Experiment on synthetic videos

model described in Section 5.3.1, i.e., the pixel-based Gaussian background model and the 10 components Gaussian mixture foreground model. This result can be considered as a result from a variant of ordinary background subtraction. After incorporating the spatial and temporal smoothness constraints, we can obtain the results in Figure 5.3(c). This procedure has the similar effect of the false detection suppressing process used in [30], and eliminates most of the random noise. Finally we incorporate the shape prior model and obtain the satisfying results in Figure 5.3(d) using our complete framework.

In Figure 5.4 we show a difficult example of an outdoor human walking sequence. In this sequence, the subject is moving from one side to the other. There are other moving objects in the background, and there is severe interlacing between the foreground and the background. Our model is able to obtain relatively smooth object boundaries.

Finally we show some segmentation results on real world video sequences from the Caviar Database [18]. These sequences are more difficult than the previous ones in that the image quality is lower, the motion of the subject is larger, and the reflection on the floor is severer. Another difficulty is that we do not have clean background data for the background modeling, hence the background is initialized by averaging over the whole sequence. In Figure 5.5(a), we

(a) Input sequence

(b) Results using only low-level information

(c) Results after imposing spatio-temporal constraints

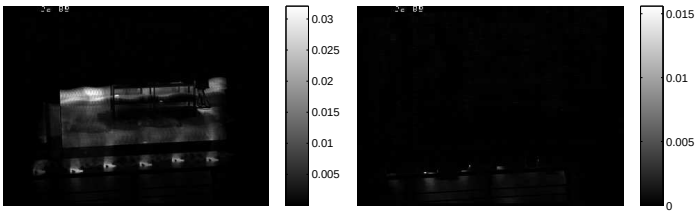
(d) Results after incorporating shape prior

Figure 5.3: Experiment on real world videos (1)

Figure 5.4: Experiment on real world videos (2)

(a) Input sequence with output overlaid (estimated shape prior images inlaid)

(b) Initial background (c) Estimated background (d) Improvement (difference)



(e) Initial covariance (f) Estimated covariance

Figure 5.5: Experiment on real world videos (3)

show the results with all the cues combined, overlaid on the input images. We also show the estimated prior shapes which are embedded in the corresponding images. The the prior shapes are noticeably different from the test images, even though the estimated poses are close enough. Therefore, one really needs to rely on the bottom-up model to capture the fine details, especially in these low quality images. In other words, it is important to update the bottom-up model along with the segmentation. In the second row of Figure 5.5 we show the improvement of the estimation of the background model. Figure 5.5(b) is the averaged background, Figure 5.5(c) is the re-estimated mean of the background model based on the segmentation, and Figure 5.5(d) is the difference between the former two. Note that the re-estimated background is much cleaner than the initial one, which has visible artifacts such as the phantoms of the human foot in multiple locations. The bottom row of Figure 5.5 shows the comparison of the variance maps of the initial averaged background (Figure 5.5(e)) and the final re-estimated background (Figure 5.5(f)). One can clearly see that the variance becomes significantly smaller after model parameter update, which means a stabler background model.

Figure 5.6 shows a sequence with considerably different poses from our training shapes.

Figure 5.6: Experiment on real world videos (4)

Since our shape prior model is based on a softmax function of the signed distance map, and the strength of the shape prior is gradually weakened by an annealing schedule applied to the parameter θ_{mag} in Equation (5.11), our framework is able to tolerate quite large variances of the object shape, and achieve satisfactory performance even when the shape prior is not accurate.

5.5 Summaries

In this chapter we have presented a general framework for video-based object segmentation using spatio-temporal Markov random fields. This framework allows us to incorporate both top-down and bottom-up information, and impose reliable spatial and temporal constraints. The loopy belief propagation algorithm provides an effective and efficient solution for the inference problem. Moreover, one can perform segmentation and estimate the model parameters simultaneously using the EM algorithm.

While we showed results of one practical implementation that combined several specific modules in our framework, one can easily replace one or more of these modules with other methods for various applications. Improvements can be made in some different aspects. For example, currently, the optical flow induced temporal constraints has not been taken special care of. That is, we compute the optical flow as a preprocessing and do not update it like other model parameters. There could be problems caused by the failed optical flow, as shown in [58]. We are further exploring the possibility to refining the optical flow estimation and changing the temporal neighborhood structure dynamically. Another particularly interesting problem is how one can utilize the dynamic information carried in the shape prior model. Currently we use some simple assumptions to reduce the search space when we estimate the shape prior. Theoretically, the training shape sequences should follow the similar dynamics of the objects in the image sequences. Better predictions can be achieved by learning the dynamics. We are also interested in the shape manifold learning [31]. A properly learned manifold and distance metric can help us work with the shape images without resorting to an explicit shape model for

the shape matching problem.

We are working on a Gaussian process latent variable model based shape manifold, which will be able to handle 3D poses and multiple views, instead of the simple silhouette based prior.In object-specific shape prior models, the training shape instances often vary mainly in poses. In such cases, these shapes can be modeled as a smooth finite-dimensional manifold of the infinite-dimensional shape space using the manifold learning techniques [31]. This is especially useful in the video-based object segmentation problem, since the shape manifold often followed the same dynamics carried in the image sequence. For example, theoretically, the shape priors for a human walking sequence will most probably traverse the human pose manifold. One can hence have better predictions of the shape priors. Spectral Regression [14] is an efficient regularized subspace learning technique based on regression and spectral graph analysis. We use spectral regression in our implementation to embed a sequence of silhouette based shape priors into a subspace.

Chapter 6
Conclusions

In this thesis, we addressed the task of object segmentation, a mid-level vision problem, by incorporating the high-level prior information into the low-level image cues, under a unified graphical model framework. In particular, we investigated the object segmentation problem in both image and video settings, and made the following contributions.

First, for static 2D images, we proposed a single generative graphical model framework to couple deformable models with Markov random fields. The model fuses two different sources of information in a stochastic setting: the region appearance cues modeled by the MRF and the shape outline cues embodied in the deformable model. Fusing the two aspects of object appearance leads to discernible improvements in accuracy of object segmentation but it also increases the method's robustness to background clutter and image noise. Despite these benefits, the joint formulation can introduce significant computational burden to the segmentation process. To solve this problem we presented two approximate solutions that exploit the probabilistic graphical structure of the model. One is a variational approach that seemingly decouples the MRF and deformable model, resulting in reduced computational effort, but retains the interaction through an iterative inference process. We contrasted this approach to inference in the fully coupled probabilistic model using contour factorization, solved by loopy belief propagation. Both methods can lead to similar segmentation accuracy for 2D images, yet they have respective advantages. The fully coupled model shows a theoretically tighter coupling of the two sets of cues, while the variational approach provides the flexibility to adopt other modeling methodologies such as physics-based modeling. We demonstrated that the variational approach directly generalizes to the 3D, and potentially higher dimensional, data while retaining the ability to easily incorporate many standard computational models, such as the Finite Element Method. This opens a vast potential to further improve performance of segmentation approaches by combining multiple state-of-the-art methods in a coupled but computationally tractable stochastic modeling manner.

In the second contribution we focused on the task of modeling specific classes of shapes that can be used as shape priors (top-down context), replacing the above, generic deformable

models. To achieve this, we developed a new contour-based shape modeling method based on the family of Profile HMMs. Profile HMMs, a strongly linear subclass of HMMs, have shown great success in modeling biological sequences in a computational efficient manner. We showed that this class of models can bring in substantial benefits to modeling of the shape, across different tasks such as shape matching or shape classification. The profiles are translation and rotation invariant, with the ability to sustain significant changes in scale and local deformations that appear within a class of shapes. At the same time, using the profile structure we developed an $O(\#observations)$ running time algorithm for analysis of shapes, a significant improvement over the quadratic complexity of traditional ergodic HMM shape models. We finally proposed an extension to the model to handle the important long-range dependencies that characterize typical multi-part structure of complex objects. We achieved this by a second-level embedding of the profile model match parameters onto the learned shape manifold. This model showed significant improvements in shape matching on some extremely difficult shape families but still retained efficiency of the profiles it is based upon.

The last part of our work demonstrated that the segmentation and contextual shape modeling can be easily coupled in a graphical model framework to solve the challenging problem of object segmentation in video sequences. We proposed a new spatio-temporal MRF framework that combines top-down high-level prior information, bottom-up low-level image features, spatial region smoothness, and temporal constraints simultaneously. Our experiments show that all the modules are important for accurate segmentation of objects whose shape changes in each frame of the video sequence. The top-down shape prior helps eliminate incorrect topology and obtain a coarse segmentation efficiently, which can then be significantly refined by the bottom-up approaches using the low-level image cues. The temporal smoothness constraints that abandon the traditional fixed-grid structure of prior spatio-temporal MRF approaches in favor of more accurate temporal neighbors are critical for improved segmentation performance. Unfortunately, inclusion of these constraints intimately relies on accurate estimation of the temporal structure which can be easily overwhelmed by noise.

Our work also reveals the opportunity for a number of possible improvements and new research directions to further our understanding and computational modeling of the interplay of bottom-up and top-down influences on middle-vision processes, such as the object segmentation. Some of the most interesting aspects are:

- Extension of coupled segmentation framework to the 4D case. To properly handle the 4D task using our current framework, the 2D shape model needs to be extended to 3D. This

requires a higher dimensional shape description and representation, e.g., a surface-based descriptor and a random field based representation.

- We presented one practical implementation of the new spatio-temporal MRF model for video-based object segmentation. It is currently working on fixed camera settings because of the background modeling method we are using. One can generalize this to the moving camera settings by adopting other background modeling methods.

- Another particularly interesting problem is how to utilize the dynamic information carried in the shape prior model. As we shown in Figure 5.1(a), we currently use independent shape priors for different frames. Theoretically, the prior shape sequences should follow the similar dynamics of the objects in the image sequences. Better predictions can be achieved by learning the dynamics.

- The presented object segmentation framework could potentially be extended to the problem of object recognition. For example, the traditional MRF model can be easily adapted to patch-based object recognition tasks, and a new layer of nodes that carry higher-level information such as object poses can be added into the otherwise only low-level image cues.

References

[1] W. Mio A. Srivastava, S. H. Joshi and X. Liu. Statistical shape analysis: Clustering, learning and testing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(4):590–602, April 2005.

[2] Y. Aloimonos. Visual shape computation. *Proc. IEEE*, 76(8):899–916, August 1988.

[3] Nafiz Arica and Fatos T. Yarman-Vural. A shape descriptor based on circular hidden Markov model. In *ICPR*, volume 1, pages 1924–1927, September 2000.

[4] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, April 2002.

[5] J. E. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society Series B*, 48(3):259–302, 1986.

[6] Manuele Bicego and Vittorio Murino. Investigating hidden Markov models' capabilities in 2D shape classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):281–286, February 2004.

[7] Eran Borenstein, Eitan Sharon, and Shimon Ullman. Combining top-down and bottom-up segmentation. In *Proceedings of IEEE Workshop on Perceptual Organization in Computer Vision*, 2004.

[8] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *ICCV*, volume 2, pages 642–655, 2001.

[9] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, November 2001.

[10] BrainWeb. Simulated brain database, http://www.bic.mni.mcgill.ca/brainweb/.

[11] Matthieu Bray, Pushmeet Kohli, and Philip H. S. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *ECCV*, volume 2, pages 642–655, 2006.

[12] Heinz Breu, Joseph Gil, David Kirkpatrick, and Michael Werman. Linear time euclidean distance transform algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(5):529–533, May 1995.

[13] William Byne, Ruth Bleier, and Lanning Houston. Variations in human corpus callosum do not predict gender: A study using magnetic resonance imaging. *Behavioral Neuroseience*, 102(2):222–227, 1988.

[14] Deng Cai, Xiaofei He, and Jiawei Han. Spectral regression for efficient regularized subspace learning. In *ICCV*, 2007.

[15] Jinhai Cai and Zhi-Qiang Liu. Integration of structural and statistical information for unconstrained handwritten numeral recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(3):263–270, March 1999.

[16] Jinhai Cai and Zhi-Qiang Liu. Hidden Markov models with spectral features for 2D shape recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(12):1454–1458, December 2001.

[17] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[18] CAVIAR. The caviar database. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[19] T. Chen and D.N. Metaxas. Image segmentation based on the integration of markov random fields and deformable models. In *MICCAI*, 2000.

[20] S.M. Choi, J.E. Lee, J. Kim, and M.H. Kim. Volumetric object reconstruction using the 3d-mrf model-based segmentation. *IEEE Trans. Med. Imag.*, 16(6), 1997.

[21] L.D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2), 1991.

[22] L.D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11), 1993.

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

[24] Samuel Dambreville, Yogesh Rathi, and Allen Tannen. Shape-based approach to robust image segmentation using kernel PCA. In *CVPR*, pages 977–984, 2006.

[25] R.H. Davies, C.Twining, T.F. Cootes, and C.J. Taylor. A minimum description length approach to statistical shape modelling. *IEEE Trans. Med. Imag.*, 21(5):525–537, 2002.

[26] R.C. Dubes, A.K. Jain, S.G. Nadabar, and C.C. Chen. Mrf model-based algorithm for image segmentation. In *ICPR*, 1990.

[27] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972.

[28] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[29] Sean R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, October 1998.

[30] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using non-parametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, July 2002.

[31] Patrick Etyngier, Florent Segonne, and Renaud Keriven. Shape priors using manifold learning techniques. In *ICCV*, 2007.

[32] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[33] Ana L. N. Fred, Jorge S. Marques, and Pedro Mendes Jorge. Hidden Markov models vs syntactic modeling in object recognition. In *ICIP*, volume 1, pages 893–896, October 1997.

[34] W. Freeman, E. Pasztor, and O. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, October 2000.

[35] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematics Software*, 3(3):209–226, 1997.

[36] Nir Friedman and Stuart Russell. Image segmentation in video sequences: A probabilistic approach. In *UAI*, pages 175–181, 1997.

[37] Yoram Gdalyahu and Daphna Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(12):1312–1328, 1999.

[38] D. Geiger and D. Heckerman. Knowledge representation and inference in similarity networks and bayesian multinets. *Artificial Intelligence*, 82(6):45–74, November 1996.

[39] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, November 1984.

[40] Z. Ghahramani. On structured variational approximations. Technical Report CRG-TR-97-1, 1997.

[41] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. *Digital Image Processing Using MATLAB*. Prentice Hall, 1st edition, 2004.

[42] Xiao Han, Chenyang Xu, and Jerry L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 25(6):755–768, June 2003.

[43] Yang He and Amlan Kundu. 2-D shape classification using hidden Markov model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(11):1172–1184, November 1991.

[44] S.L. Horowitz and T. Pavlidis. Picture segmentation by a directed split and merge procedure. In *ICPR*, pages 424–433, 1974.

[45] P. V. C. Hough. Machine analysis of bubble chamber pictures. In *International Conference on High Energy Accelerators and Instrumentation*, 1959.

[46] Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas. A graphical model framework for coupling MRFs and deformable models. In *CVPR*, October 2004.

[47] Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas. A hybrid framework for image segmentation using probabilistic integration of heterogeneous constraints. In *CVBIA*, pages 82–92, 2005.

[48] Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas. Embedded profile hidden Markov models for shape analysis. In *ICCV*, October 2007.

[49] T.N. Jones and D.N. Metaxas. Image segmentation based on the integration of pixel affinity and deformable models. In *CVPR*, 1998.

[50] M.I. Jordan, Z. Ghahramani, T. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2), 1999.

[51] Michael I. Jordan, editor. *Learning in Graphical Models*. The MIT Press, Cambridge, MA, 1998.

[52] Michael I. Jordan. Graphical models. *Statistical Science (Special Issue on Bayesian Statistics)*, 19(1):140–155, February 2004.

[53] Shunsuke Kamijo, Katsushi Ikeuchi, and Masao Sakauchi. Segmentations of spatio-temporal images by spatio-temporal markov random field model. In *EMMCVPR*, pages 298–313, October 2001.

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

[55] Frank Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47, 2001.

[56] M. Pawan Kumar, Philip H. S. Torr, and A. Zisserman. Obj cut. In *CVPR*, volume 1, 2005.

[57] Sanjiv Kumar and Martial Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *ICCV*, volume 02, page 1150, 2003.

[58] E. Scott Larsen, Philippos Mordohai, Marc Pollefeys, and Henry Fuchs. Temporally consistent reconstruction from multiple video streams using enhanced belief propagation. In *ICCV*, October 2007.

[59] Anat Levin and Yair Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV*, pages 581–594, May 2006.

[60] Sven Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, August 1998.

[61] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics (Proceedings of SIGGRAPH)*, 21(4), 1987.

[62] F. Luthon, A. Caplier, and M. Liévin. Spatiotemporal mrf approach to video segmentation: application to motion detection and lip segmentation. *Signal Processing*, 76(1):61–80, 1999.

[63] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *Journal of the American Statistical Association*, 82(397):76–89, March 1987.

[64] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In *ICCV*, 1995.

[65] T. McInerney and Demetri Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2).

[66] D.N. Metaxas. *Physics-based Deformable Models: Applications to Computer Vision, Graphics and Medical Imaging*. Kluwer Academic Press, 1997.

[67] J.V. Miller, D.E. Breen, W.E. Lorensen, R.M. O'Bara, and M.J. Wozny. Geometrically deformed models: A method for extracting closed geometric models from volume data. *Computer Graphics (Proceedings of SIGGRAPH)*, 25(4), 1991.

[68] Hossam E. Abd El Munim and Aly A. Farag. A shape-based segmentation approach: An improved technique using level sets. In *ICCV*, pages 930–935, 2005.

[69] A. S. Ogale and Y. Aloimonos. A roadmap to the integration of early visual modules. *International Journal of Computer Vision: Special Issue on Early Cognitive Vision*, 72(1):9–25, January 2007.

[70] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computation Physics*, 79(2):12–49, October 1988.

[71] Stanley J. Osher and Ronald P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002.

[72] V. Pavlovic, B.J. Frey, and T.S. Huang. Variational learning in mixed-state dynamic graphical models. In *UAI*, 1999.

[73] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.

[74] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[75] Fabrice Robichon. Abnormal callosal morphology in male adult dyslexics: Relationships to handedness and phonological abilities. *Brain and Language*, 62:127–146, 1998.

[76] R. Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*, 13(2), 1994.

[77] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 2004.

[78] Mikael Rousson and Nikos Paragios. Shape priors for level set representations. In *ECCV*, pages 78–92, 2002.

[79] Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia. Recognition of shapes by editing shock graphs. In *ICCV*, volume 1, pages 755–762, July 2001.

[80] James A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 2nd edition, 1999.

[81] Linda G. Shapiro and George C. Stockman. *Computer Vision*. Prentice Hall, 2001.

[82] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8):731–737, August 2000.

[83] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis and Machine Vision*. Thomson Learning, 2nd edition, 1998.

[84] Stanford. Volume data archive, http://graphics.stanford.edu/data/voldata/.

[85] Chris Stauffer and W.E.L Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, June 1999.

[86] M. B. Stegmann, R. H. Davies, and C. Ryberg. Corpus callosum analysis using MDL-based sequential models of shape and appearance. In *SPIE*, pages 612–619, February 2004.

[87] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, June 2008.

[88] M.F. Tappen and W.T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *ICCV*, volume 2, pages 900–907, October 2003.

[89] Ninad Thakoor and Jean Gao. Shape classifer based on generalized probabilistic descent method with hidden Markov descriptor. In *ICCV*, volume 1, pages 495–502, October 2005.

[90] G. Tsechpenakis and D.N. Metaxas. CRF-driven implicit deformable model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[91] Remco C. Veltkamp and Michiel Hagedoorn. State of the art in shape matching. pages 87–119, 2001.

[92] Yang Wang, Kia-Fock Loe, Tele Tan, and Jian-Kang Wu. Spatiotemporal video segmentation based on graphical models. *IEEE Trans. Image Process.*, 14(7):937–947, 2005.

[93] Y. Weiss. Belief propagation and revision in networks with loops. Technical Report MIT A.I. Memo 1616, 1998.

[94] C. Xu and J.L. Prince. Gradient vector flow: A new external force for snakes. In *CVPR*, 1997.

[95] J. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *International Joint Conference on Artificial Intelligence, Distinguished Presentations Track*, August 2001.

[96] Zhaozheng Yin and Robert Collins. Belief propagation in a 3d spatio-temporal mrf for moving object detection. In *CVPR*, 2007.

[97] Dengsheng Zhang and Guojun Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, January 2004.

[98] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Trans. Med. Imag.*, 20(1), 2001.

Vita

Rui Huang

EDUCATION

October 2008 Ph.D. in Computer Science, Rutgers University, U.S.A.

August 2002 M.E. in Pattern Recognition and Intelligent Systems, Chinese Academy of Sciences, P.R.C.

July 1999 B.S. in Computer Softwares, Peking University, P.R.C.

EXPERIENCE

Jun.2005-Aug.2005 Summer Intern, Siemens Corporate Research, Princeton, NJ, U.S.A.

Sep.2002-Jun.2008 Graduate Assistant/Teaching Assistant, Department of Computer Science, Rutgers University, New Brunswick, NJ, U.S.A.

Aug.2001-Sep.2001 Summer Intern, Microsoft Research China (now Microsoft Research Asia), Beijing, P.R.C.

Feb.1999-Jun.1999/Sep.2000-Jun.2002 Research Assistant, Institute of Automation, Chinese Academy of Sciences, Beijing, P.R.C.

PUBLICATION

A New Spatio-Temporal MRF Framework for Video-based Object Segmentation, Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas, *submitted for review*, 2008.

A Variational Level Set Approach to Segmentation and Bias Correction of Images with Intensity Inhomogeneity, Chunming Li, Rui Huang, Zhaohua Ding, Chris Gatenby, Dimitris Metaxas, and John Gore, in Proceedings of *the 11th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'08)*, 2008

Embedded Profile Hidden Markov Models for Shape Analysis, Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas, in Proceedings of *the 11th International Conferences on Computer Vision (ICCV'07)*, 2007.

Shape Analysis Using Curvature-Based Descriptors and Profile Hidden Markov Models, Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas, in Proceedings of *the 4th IEEE International Symposium on Biomedical Imaging (ISBI'07)*, 2007.

A Novel Tag Removal Technique for Tagged Cardiac MRI and its Applications, Zhen Qian, Rui Huang, Dimitris N. Metaxas and Leon Axel, in Proceedings of *the 4th IEEE International Symposium on Biomedical Imaging (ISBI'07)*, 2007.

A Graphical Model Framework for Image Segmentation, Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas, *Applied Graph Theory in Computer Vision and Pattern Recognition*, Springer, 2007.

A Profile Hidden Markov Model Framework for Modeling and Analysis of Shape, Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas, in Proceedings of *the 13th International Conference on Image Processing (ICIP'06)*, 2006.

A Tightly Coupled Region-Shape Framework for 3D Medical Image Segmentation, Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas, in Proceedings of *the 3rd IEEE International Symposium on Biomedical Imaging (ISBI'06)*, 2006.

Hybrid Framework for Image Segmentation Using Probabilistic Integration of Heterogeneous Constraints, Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas, in Proceedings of *the 1st International Workshop on Computer Vision for Biomedical Image Applications (CVBIA'05) (Lecture Notes in Computer Science 3765)*, 2005.

Deformable-Model Based Textured Object Segmentation, Xiaolei Huang, Zhen Qian, Rui Huang, and Dimitris N. Metaxas, in Proceedings of *the 5th International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR'05), (Lecture Notes in Computer Science 3757)*, 2005.

A Hybrid Face Recognition Method using Markov Random Fields, Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas, in Proceedings of *the 17th International Conference on Pattern Recognition (ICPR'04)*, 2004.

A Graphical Model Framework for Coupling MRFs and Deformable Models, Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas, in Proceedings of *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, 2004.

Kernel-Based Nonlinear Discriminant Analysis for Face Recognition, Qingshan Liu, Rui Huang, Hanqing Lu, and Songde Ma, in *Journal of Computer Science and Technology*, Vol. 18, No. 6, 788-795, Nov, 2003.

Solving the Small Sample Size Problem of LDA, Rui Huang, Qingshan Liu, Hanqing Lu, and Songde Ma, in Proceedings of *the 16th International Conference on Pattern Recognition (ICPR'02)*, 2002.

Kernel-Based Optimized Feature Vectors Selection and Discriminant Analysis for Face Recognition, Qingshan Liu, Rui Huang, Hanqing Lu, and Songde Ma, in Proceedings of *the 16th International Conference on Pattern Recognition (ICPR'02)*, 2002.

Face Recognition Using Kernel-Based Fisher Discriminant Analysis, Qingshan Liu, Rui Huang, Hanqing Lu, and Songde Ma, in Proceedings of *the 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR'02)*, 2002.