ECOLE CENTRALE PARIS

PHD THESIS

to obtain the title of

Doctor of Ecole Centrale Paris
Specialty : APPLIED MATHEMATICS

Defended by

Bo XIANG

Knowledge-Based Image Segmentation

Using Sparse Shape Priors and High-Order MRFs

prepared at Ecole Centrale de Paris, CVN laboratory

defended on November 28, 2013

Committee :

Reviewers :	Laurent NAJMAN	-	Université Paris-Est
	Jan KYBIC	-	Czech Technical University
Examiners :	Henri MAITRE	-	Télécom ParisTech
	Nikos KOMODAKIS	-	Ecole des Ponts ParisTech
	Gregoire MALANDAIN	-	INRIA Sophia-Antipolis
	Benjamin GLOCKER	-	Microsoft Research Cambridge
	Jean-Francois DEUX	-	Henri Mondor Hospital
Advisor :	Nikos PARAGIOS	-	Ecole Centrale Paris

Abstract

In this thesis, we propose a novel framework for knowledge-based segmentation using high-order Markov Random Fields (MRFs). We represent the shape model as a point distribution graphical model which encodes pose invariant shape priors through L1 sparse higher order cliques. Each triplet clique encodes the local shape variation statistics on the angle measurements which inherit invariance to global transformations (*i.e.* translation, rotation and scale). A sparse higher-order graph structure is learned through MRF training using dual decomposition, producing boosting efficiency while preserving its ability to represent the shape variation.

We incorporate the prior knowledge in a novel framework for model-based segmentation. We address the segmentation problem as a maximum a posteriori (MAP) estimation in a probabilistic framework. A global MRF energy function is defined to jointly combine regional statistics, boundary support as well as shape prior knowledge for estimating the optimal model parameters (*i.e.* the positions of the control points). The pose-invariant priors are encoded in second-order MRF potentials, while regional statistics acting on a derived image feature space can be exactly factorized using Divergence theorem.

Furthermore, we propose a novel framework for joint model-pixel segmentation towards a more refined segmentation when exact boundary delineation is of interest. A unified model-based and pixel-driven integrated graphical model is developed to combine both top-down and bottom-up modules simultaneously. The consistency between the model and the image space is introduced by a model decomposition which associates the model parts with pixels labeling.

Both of the considered higher-order MRFs are optimized efficiently using state-of-the-art MRF optimization algorithms. Promising results on computer vision and medical image applications demonstrate the potential of the proposed segmentation methods.

Résumé

Nous présentons dans cette thèse une approche nouvelle de la segmentation d'images, avec des descripteurs a priori utilisant des champs de Markov d'ordre supérieur. Nous représentons le modèle de forme par un graphe de distribution de points qui décrit les informations a priori des invariants de pose grâce à des cliques L1 discrètes d'ordre supérieur. Chaque clique de triplet décrit les variations statistiques locales de forme par des mesures d'angle, ce qui assure l'invariance aux transformations globales (translation, rotation et échelle). L'apprentissage d'une structure de graphe discret d'ordre supérieur est réalisé grâce à l'apprentissage d'un champ de Markov aléatoire utilisant une décomposition duale, ce qui renforce son efficacité tout en préservant sa capacité à rendre compte des variations.

Nous introduisons la connaissance a priori d'une manière innovante pour la segmentation basée sur un modèle. Le problème de la segmentation est ici traité par estimation statistique d'un maximum a posteriori (MAP). L'optimisation des paramètres de la modélisation - c'est à dire de la position des points de contrôle - est réalisée par le calcul d'une fonction d'énergie globale de champs de Markov (MRF). On combine ainsi les calculs statistiques régionaux et le suivi des frontières avec la connaissance a priori de la forme. Les descripteurs invariants sont estimés par des potentiels de Markov d'ordre 2, tandis que les caractéristiques régionales sont transposées dans un espace de caractéristiques et calculées grâce au théorème de la Divergence.

De plus, nous proposons une nouvelle approche pour la segmentation conjointe de l'image et de sa modélisation ; cette méthode permet d'obtenir une segmentation plus fine lorsque la délimitation précise d'un objet est recherchée. Un modèle graphique combinant l'information a priori et les informations de pixel est développé pour réaliser l'unité des modules "top-down" et "bottom-up". La cohérence entre l'image et sa modélisation est assurée par une décomposition qui associe les parties du modèle avec la labellisation de chaque pixel.

Les deux champs de Markov d'ordre supérieur considérés sont optimisés par les algorithmes de l'état de l'art. Les résultats prometteurs dans les domaines de la vision par ordinateur et de l'imagerie médicale montrent le potentiel de cette méthode appliquée à la segmentation.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Due to the significant advances in imaging devices and technologies, digital images play a more and more important role in our life. An image records a scene of the real world in a numeric representation that can be stored, transmitted and studied afterwards. The well known proverb "a picture is worth a thousand words" indicates that an image has a powerful ability of describing the rich information it carries. In computer vision, images are used to perform perception tasks such as object detection, tracking and recognition among others. In the medical field, images are acquired through various modalities such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT), ultrasound *etc.* , providing an invaluable access to see the interior of human body and allowing physicians to make more accurate diagnosis. In both domains, extraction of useful information from images has become the essential task.

Although humans can solve this task naturally and easily (at least in the case of 2D natural interpretation of scenes), it still remains difficult for a computer to interpret an image automatically. In order to interpret an image or to understand the scene, a fundamental low to mid-level vision task consists on partitioning the image into a number of meaningful parts, which can provide the clues to answer the questions such as: what and where are the components of the scene? This leads us to one of the most essential tasks in computer vision: *image segmentation*.

Given an image, image segmentation is the process of partitioning the image into multiple components, so that each component is meaningful (*i.e.* corresponding to different objects or natural parts of objects). Since an image is composed of a number of pixels, the segmentation problem can be treated as a labeling problem which aims to assign each pixel a label indicating a particular component in the scene. Alternatively, the segmentation task

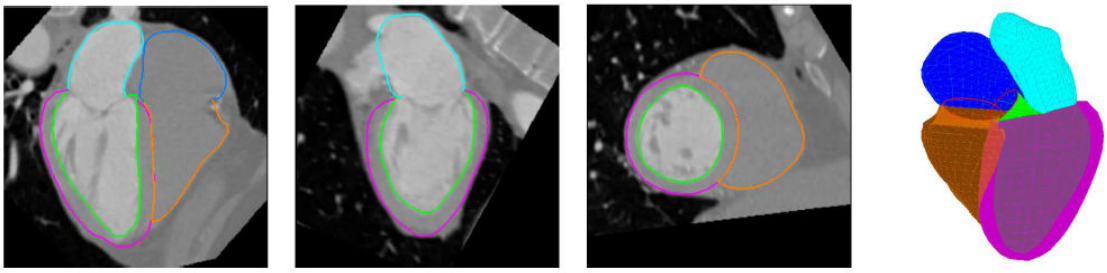


Figure 1.1: Four-chamber heart segmentation [Zheng 2008].

can be considered as extracting the boundaries between different objects, so that the image is partitioned into meaningful regions according to the boundaries. Fig.1.1 shows an example of heart segmentation, where (a,b,c) represent three orthogonal cuts from a 3D cardiac CT volume while (d) shows a reconstructed triangulated model. Four-chamber heart segmentation shows the extracted boundaries: the left ventricle (LV) endocardium in green, the LV epicardium in magenta, the left atrium (LA) in cyan, the right ventricle (RV) in brown and the right atrium (RA) in blue. As soon as the segmentation of an given image is available, it not only enables the computer to know about the image composition, but also enables the computer to analyze the qualitative and quantitative properties of the object of interest based on its segmented region in the image.

Segmentation has been widely applied to various computer vision tasks such as object detection (*i.e.* pedestrian detection, localization of objects in satellite images), recognition (*i.e.* face recognition), occlusion boundary estimation within motion or stereo systems, image compression, image editing, or image retrieval. Among these applications, some tasks (*i.e.* detection) require the localization of the object of interest which could be represented by its centroid, while other tasks (*i.e.* boundary estimation) make use of the image segmentation. In medical image analysis, accurate segmentation plays a crucial role in computer-aided diagnosis and therapy. In this context, the aim of segmentation is to delineate anatomical structures. Based on the segmented images, computers are able to visualize structures of interests in 3D, measure the geometric properties of the structures, while the obtained information can furthermore contribute as the factors for diagnosis and therapy. For example, segmenting the left ventricle is necessary to assess the heart functions using quantitative indicators such as ejection fraction (EF), myocardium mass (MM), and stroke volume (SV). Another example is tumor segmentation which not only detects the tumor inside of human organ, but also allows measuring the size of the tumor and modeling its growth through periodic acquisitions and even making predictions of its evolution.

In brief, image segmentation is now integrated routinely in a multitude of clinical settings such as study of anatomical structure, localization of pathology, quantification of tissue volumes, diagnosis/treatment planning and computer-integrated surgery.

As we can see, the accuracy of the segmentation is an absolute necessity in these applications. The more accurate the segmentation is, the more reliable and the better performance the vision or medical analytical task can achieve. In most cases, manual segmentation by experts can provide the best and the most reliable result, but it is time-consuming and tedious. Moreover, manual segmentation is subjective to operator variability. These facts motivate researchers to develop automatic segmentation methods in order to deal with large datasets, while achieving the accuracy of manual segmentation.

Challenges

Image segmentation remains a difficult task for a computer. The challenge is due to the huge variability of object shape and the variation in image quality. Regarding shape variability, we can take the example of segmenting a human body under various poses (*i.e.* walking, jumping, sitting) for an individual or for different populations (*i.e.* tall ones or short ones). Regarding variations in image quality, we can take the example of medical images. Medical images are often corrupted by noise and sampling artifacts which are introduced during the acquisition process. Moreover, the low contrast between different anatomical structures or the even worse cases when different structures have similar appearances in the image, can cause considerable difficulties. In particular, classical segmentation techniques which rely on edge detection fail to produce the desired segmentation results.

In order to deal with these challenges, shape models have been incorporated as priors for image segmentation where the optimal segmentation map is constrained by the manifold of valid shapes of the object of interest. In this manner, segmentation is more robust to noise, while using the global shape of the object can also alleviate the ambiguity of non-visible boundaries between different objects due to similar tissue properties. Shape models are learned in order to have the ability to describe the shape variations of the same class of the objects.

To this end, statistical models have been proposed to learn the prior knowledge from a training set of the shape instances. These models represent the shape variation by linear or non-linear representations as well as in global or local manner. For example, active shape model (ASM) / point distribution model (PDM) is one of the most popular statistical models used in computer vision community, and it has been widely applied to image segmentation tasks. However, as a linear model, it cannot capture non-linear shape variations such as an articulated object. On top of that, this global model lacks of the flexibility

of control local variations and it requires a large training set to represent the full range of shape variations in high dimensional space. Thus, non-linear and local shape priors have the potential to overcome these limitations. However, local or global models require handling the global transformations (*i.e.* translation, rotation and scale changes) in an explicit manner in both training and learning. Statistical model training requires aligning all the training samples into a common coordinate system before capturing their statistical variations. This alignment process introduces strong bias and requires to explicitly estimate pose parameters (*i.e.* position, rotation and scale) of the model in the test image. Obviously, pose-invariance is one of the desirable properties of statistical shape model.

Given the statistical shape model, segmentation can be formulated as estimating the model parameters in an observed image, where visual support and shape priors are combined in a cost function. The inference process aiming to determine the optimal solution is another challenge, since the quality of the solution depends on the optimization algorithm. Among the existing optimization methods, one can cite variational methods and their derivative-driven minimization for the optimization of continuous objective functions. However, these methods generally converge to local minima, while one has to compute the derivative which restrains their use on differentiable objective functions. On the other hand, significant advances have been made in discrete Markov random field (MRF) optimization methods which can achieve global minimum under certain constraints. However, these methods are based on local interactions, therefore the integration of global shape priors is not straightforward.

1.2 Thesis Overview

In this thesis, we aim to introduce image segmentation methods which are able to address the aforementioned challenges. To this end, a prerequisite is to build a statistical model to represent the shape properties of the object of interest. Such a model should be endowed with the following properties: (1) global pose-invariance *i.e.* invariance under translation, rotation and scale changes, (2) expression of linear and non-linear shape variations in a local manner, (3) compactness and easiness of being encoded into segmentation framework. Then, we want to integrate the shape prior into the automatic segmentation framework, where accurate result can be obtained by using efficient inference algorithms.

In order to achieve the above objective, we introduce a Markov Random Field (MRF) shape constrained model for image segmentation. The pose-invariant shape prior is expressed through a graphical model, where the nodes of the graph correspond to control points on the object boundary, and the cliques of the graph encode local constraints on the relative position of points. Towards compactness and computational efficiency, the

set of optimal cliques representing the observed shape variation is learned from the data. Based on this graph representation, a probabilistic framework expresses the segmentation task as a maximum posteriori estimation process towards recovering the optimal model parameters (*i.e.* the positions of the control points). The energy function encodes regional statistics, boundary support as well as shape prior knowledge. Using efficient MRF inference algorithms, optimal model solution can be obtained and lead to accurate segmentation. Furthermore, we extent this probabilistic framework for image segmentation through a unified model-based and pixel-driven integrated graphical model, where both model parameters and pixel labeling can be solved in a single shot optimization. Integrating both top-down and bottom-up approaches in a unified framework can achieve a more precise segmentation result than the individual approaches.

The remainder of this thesis is organized as follows.

- In chapter 2, we review the state of the art of segmentation techniques which are related to our work, including deformable models, active shape /appearance models, and graph-based methods. In addition, we describe Markov Random Fields and the advanced discrete MRF optimization methods.

- In chapter 3, we propose a novel statistical shape model. The shape model is represented as a point distribution graphical model which encodes pose invariant shape priors through L_1 sparse higher order cliques. In particular, each triplet clique encodes the local shape variation statistics on two inner angles. A subset of cliques from all possible triplet cliques is learned based on MRF learning via dual decomposition from a training set. The selected cliques construct a sparse graph which can provide the best possible compromise between the ability to encode the observed shape variation and compactness.

- In chapter 4, we propose a novel framework for model-based segmentation using shape priors. We formulate the segmentation problem as a maximum a posteriori (MAP) estimation in a probabilistic framework. The MRF energy encodes both data and prior constraints. Prior energy is defined by higher-order potential encoding the local shape priors, while regional statistics can be exactly factorized into pairwise terms (in 2D cases) or second-order terms (in 3D cases) through Divergence theorem. The considered higher-order MRF is optimized using dual decomposition.

- In chapter 5, we propose a novel framework for joint model-pixel segmentation through a unified model-based and pixel-driven coupled approach. A shape decomposition allows the introduction of region-driven image statistics as well as pose-invariant constraints. Regional triangles are associated with pixel labeling, aiming

to create consistency between the model and the image space. Furthermore, it produces the state of the art results of exact boundary delineation through the combined model-pixel graph.

- In chapter 6, we conclude the thesis by summarizing the contributions as well as discussing some improvements and directions for future research.

Chapter 2

State of the Art

In this chapter, we review image segmentation techniques, Markov Random Fields (MRFs) and the associated optimization techniques.

2.1 Segmentation Techniques

Segmentation is the most widely studied topic in computer vision. There are two types of approaches to deal with this problem: model-free methods (bottom-up fashion) and knowledge-based methods (top-down fashion). Model-free methods are often based on clustering, aiming at grouping together pixels with consistent visual properties according to a certain similarity criterion. Knowledge-driven methods, on the other hand, assume that the space of admissible solutions is constrained, and they seek a solution that is a compromise between the one produced from the observations and the one expressed in the model space.

Popular examples in the context of model-free segmentation refer to the mean-shift algorithm [Comaniciu 2002], variational formulations such as the Mumford-Shah framework [Mumford 1989] and its level set variant [Chan 2001, Paragios 2002], or graph-based methods including normalized cuts [Shi 2000], graph-cuts [Boykov 2006] *etc*. Due to the lack of assumptions on the geometric form of the object of interest, these methods are free in terms of admissible solutions, which is a desired property in certain cases but also an undesirable one since it can lead to erroneous results due to intensity variability, occlusions, noise presence, *etc*.

Knowledge-based methods are either manifold constrained or manifold enhanced. The former class of methods models geometric variation of the object of interest and then seeks an instance of this model in the image. Active shape [Cootes 1995] and active appearance models [Cootes 2001] are popular examples. Manifold enhanced methods aim at mini-

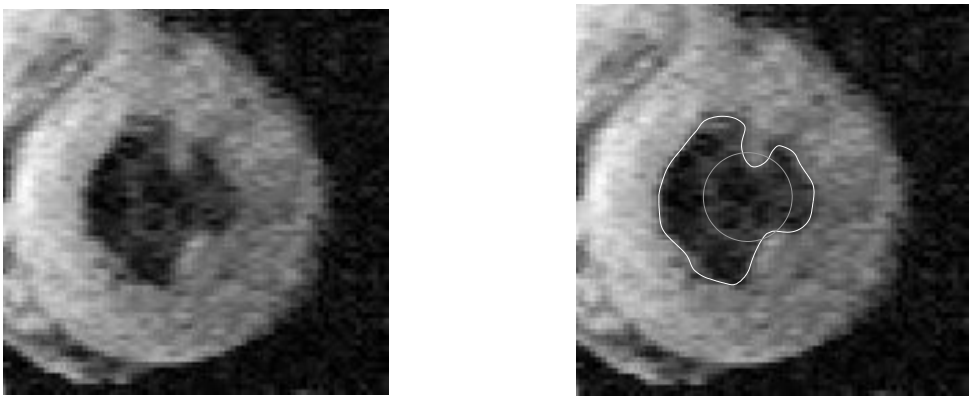
mizing the distance of the solution from the learned manifold. Active contours/surface solutions [Staib 1996] are some examples. Both groups of knowledge-based methods inherit a severe limitation with respect to pose, since the sought solution should be brought to the same referential as the ones used in learning. The manifold constrained methods are rather robust to noise, but it cannot cope with examples not seen during training, while the manifold enhanced methods is often a compromise (to a certain extent) between model-free and manifold constrained methods.

There is another type of knowledge-based approaches widely used in medical imaging, namely *atlas-based* methods. The underlying idea of this approach is to achieve segmentation by registering an atlas to the target image. The atlas is a reference image associated with a ground truth segmentation. The segmentation of the target image is obtained by warping the atlas segmentation based on the deformation field from registration. In this context, choosing the atlas used as prior for the segmentation is very important. Thus, multi-atlas segmentation approaches with more sophisticated atlas construction techniques have been developed to capture shape variations of the object of interest. Since atlas-based segmentation is highly dependent on registration which is another active topic in computer vision, we will not discuss this type of approaches in details and we refer the reader to [Rohlfing 2005].

In the remainder of this section, we will review some popular segmentation methods including deformable models (*i.e.* active contours/surfaces and level sets), active shape models and active appearance models and graph-based methods. These are well studied topics in the community, and therefore we refer the reader to [McInerney 1996, Jain 1998, Montagnat 2001] for deformable contours/surfaces methods, [Cremers 2007] for level sets and variational methods, and [Heimann 2009] for active shape and appearance models. [Boykov 2006] presents detailed technical description of the basic combinatorial optimization for image segmentation via graph cuts.

2.1.1 Deformable Models

Deformable models became a landmark in computer vision and have been widely used in medical image segmentation since the pioneering publication of [Kass 1988]. Snakes, active contours, active surfaces are the various names which have been used in the literature for deformable models. To delineate object boundaries in an image, snakes use curves or surfaces that deform under the influence of *internal forces* and *external forces*. The internal forces are designed to enforce the smoothness of the curve or surface during deformation. The external forces are usually derived from the image to drive the curve or surface towards the desired images features such as strong edges within an image. Deformable models can be generally divided into two classes, depending on the definition



(a) (b)

Figure 2.1: Extraction of the inner wall of the left ventricle using active contour models [Xu 1998].

of the curve and surface: (1) parametric deformable models, also called active contours and active surfaces, and (2) non-parametric deformable models, also called level set and geometric deformable models.

Active Contours/Surfaces

Parametric deformable models represent curves and surfaces explicitly during deformation. Mathematically, a parametric curve is represented by a function $f(s) = (x(s), y(s))$, where x, y denote the coordinates and $s \in [0, 1]$ is the arc-length parameter. Given an initialization, the parameterized curve deforms through the spatial domain of an image to minimize the following energy function:

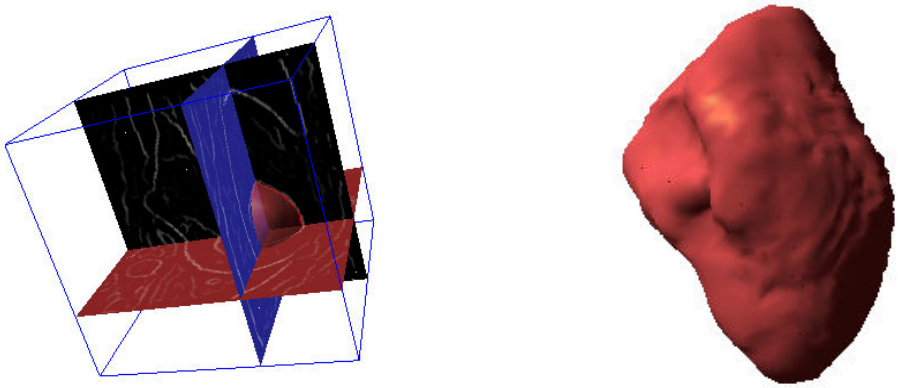
$$E(f) = E_{int}(f) + E_{ext}(f) \tag{2.1}$$

The internal energy $E_{int}(f)$ characterizes the tension or the smoothness of the contour. It consists of a first-order and a second-order continuity terms.

$$E_{int}(f) = \frac{1}{2}\int_0^1 w_1(s)\left|\frac{\partial f}{\partial s}\right|^2 + w_2(s)\left|\frac{\partial^2 f}{\partial^2 s}\right|^2 ds \tag{2.2}$$

where function $w_1(s)$ controls the tension of the curve and function $w_2(s)$ controls its rigidity. In practice, $w_1(s)$ and $w_2(s)$ are often set to be constant. The external energy $E_{ext}(f)$ imposes the image component on the curve.

$$E_{ext}(f) = \int_0^1 E_{ext}(f(s))ds \tag{2.3}$$



(a) (b)

Figure 2.2: Left ventricle segmentation using active surface models [McInerney 1995]. (a) Balloon model. (b) Reconstruction of the left ventricle.

where function $E_{ext}(x, y)$ is derived from the image data and its minima should coincide with intensity extrema, edges as well as other image features of interest. For example, the most typical function designed to attract the contour to intensity edges is:

$$E_{ext}(x, y) = -\left|\nabla \left(G_\sigma(x, y) * I(x, y)\right)\right|^2 \tag{2.4}$$

where ∇ is the gradient operator, $G_\sigma * I$ denotes the image convolved with a Gaussian filter whose deviation σ controls the attraction range. We show an example of active contour models extracting the inner wall of the left ventricle in Figure 2.1, where (a) is an original 2D MR image while (b) represents the initial contour model in gray and the final converged result in white. Another example of left ventricle segmentation using active surface surfaces is shown in Figure 2.2, where (a) embeds a balloon model in the volume image and (b) shows the reconstruction of the left ventricle.

Parametric deformable models have the following advantages: (1) their ability to cope with open or closed parametric curves or surfaces, (2) their low computational complexity, (3) the natural incorporation of a smoothness constraint that provides robustness to noise and spurious edges, and (4) the ability to integrate prior knowledge. A disadvantage is that an initial model is needed. In order to reduce sensitivity to initialization, [Cohen 1991] proposes *balloons* that use a pressure force to increase the attraction range. Another approach [Cohen 1993] of extending attraction range is to define the external energy using a distance map. However, the distance based force can cause difficulties when deforming a contour or surface into boundary concavities. To address this problem, [Xu 1998] proposes a *gradient vector flow* (GVF) field which is based on the diffusion of the edge map

that improved convergence of deforming contours into boundary concavities. However, the most important limitation of parametric models is the difficulty to deal with topological changes such as splitting or merging parts during the deformation, which is a useful property for recovering either multiple objects or an object with unknown topology. [McInemey 1999, Montagnat 2001] propose topology adaptive deformable surfaces for volume segmentation using an efficient reparameterization mechanism.

Level Sets

Geometric deformable models [Caselles 1993, Malladi 1995, Kichenassamy 1995] can handle topological changes of the unknown object to be segmented. These models are based on curve evolution theory and level set methods [Osher 1988, Sethian 1999, Osher 2003]. The curves and surfaces are represented implicitly as a zero level set function. The curve evolution is independent of parameterization, thus topological changes can be handled automatically.

Level sets evolve to fit and track the object boundaries by modifying the underlying level set function instead of the curve. Given a level set function $\phi(x, y, t)$ with the moving curve $f(s, t)$ as its zero level set, we have

$$\phi(f(s,t),t) = 0 \tag{2.5}$$

Using the curve evolution theory, if the curve f moves along its normal direction with a speed V, then the level set function ϕ satisfies the level set equation:

$$\frac{\partial \phi}{\partial t} = V\,|\nabla \phi| \tag{2.6}$$

where $\nabla\phi$ denotes the gradient of ϕ, and V is called *speed function*. There are two most common deformations in curve evolution theory: image-driven and curvature deformation. The curvature deformation is defined as $V = \alpha k$, where α is a positive constant and the curvature k at the zero level set is given $k = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}$. The main idea of the geometric deformable models is to couple the speed of deformation with the image data, so that the evolution of the curve stops at object boundaries.

Geometric deformable models allow the curve evolution equation to be modified in two ways: changing the speed function and adding additional constraints. [Caselles 1993] and [Malladi 1995] independently propose a curve evolution coupled with the image data through a multiplicative stopping term. A problem with this model is that if the object boundary has gaps, the curve passes the boundary and cannot be pulled back to the correct boundaries. [Caselles 1997] and [Yezzi Jr 1997] add an additional term in the equation that allows the curve to be pulled back, and this term behaves like the external force of

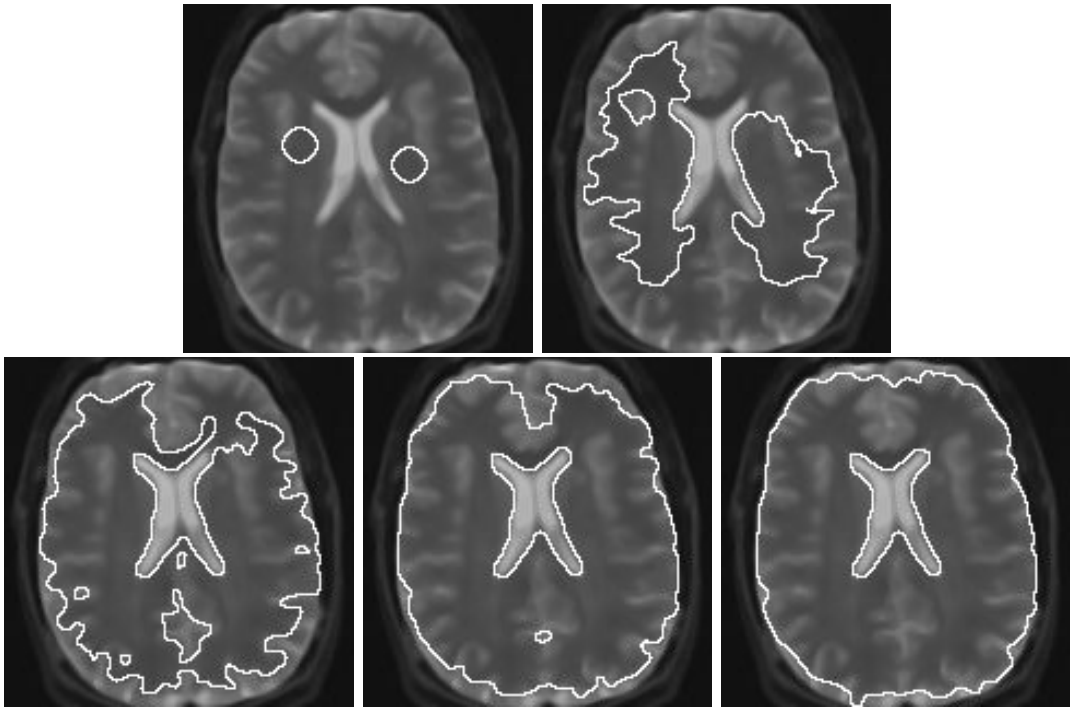


Figure 2.3: Brain segmentation using geometric deformable contours [Siddiqi 1998]. Left to right and top to bottom: iterations 1, 400, 800, 1200 and 1600.

parametric models. These methods are still sensitive to local minima, since it is based on the edges (image gradients). An alternative approach is to use image region characteristics. Earlier energy-based segmentation frameworks [Leclerc 1989, Mumford 1989] define a functional which measures the consistency of the segmented regions. Based on these framework, many level sets approaches have been developed to combine image region statistics with boundary measurements, such as [Chan 2001, Paragios 2002, Jehan-Besson 2003, Tsai 2001b].

We show an example of geometric deformable models for brain segmentation in Figure 2.3. The curve evolution is shown progressively, clearly demonstrating its ability to change the topology of the curves. While geometric deformable models can handle topological changes, they may generate shapes that have inconsistent topology with respect to the object of interest. This is the case when applied to noisy images with significant boundary gaps. Moreover, they are computationally expensive due to the iterative optimization methods of solving a partial differential equation into the entire image domain, even if several variants have been introduced to reduce significantly the computational burden.

2.1.2 Active Shape and Active Appearance Models

Active Shape Models

Active shape model (ASM) [Cootes 1995], is one of the most popular model-based approaches for medical image segmentation. It can be considered as an extension of deformable models when incorporating prior shape information. The shape prior is constructed by Point Distribution Model (PDM) which models the shape variations from a training set. In the PDM, the shape is represented by a set of points distributed on the boundary. Mathematically, it can be defined by a $n \times d$ dimensional vector concatenating each point's coordinates, where n is the number of the points and d is the dimension of the point coordinates. For example, a 2D shape of n points is defined as:

$$\mathbf{x} = (x_1, y_1, \cdots, x_n, y_n)^T \tag{2.7}$$

Given a training set, each shape is represented by n points referring to the same coordinate system (order) throughout the entire training set. Then, these shapes have to be aligned into the same coordinates system to filter out the shape variations caused by translation, rotation and scaling. This procedure is commonly accomplished using the Generalized Procrustes Analysis [Gower 1975], which minimizes the least squared error between the points. Once correspondences have been established, a Principal Component Analysis (PCA) is used to build the statistical shape model. The mean shape of the training set of N samples is calculated using:

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \tag{2.8}$$

A covariance matrix S is computed by:

$$\mathbf{S} = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{2.9}$$

An eigendecomposition of S yields the eigenvectors $\{\mathbf{P}_m\}_{m=1}^{nd}$ (representing the principle modes of variation) and the corresponding eigenvalues $\{\lambda_m\}_{m=1}^{nd}$ ((indicating their importance in the construction of model). Sorting all modes from largest to smallest variance, the first k modes are employed to model the observed variability of the training set. Then, shape instances of this population can be expressed by a linear combination of the k significant modes of variation.

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \tag{2.10}$$

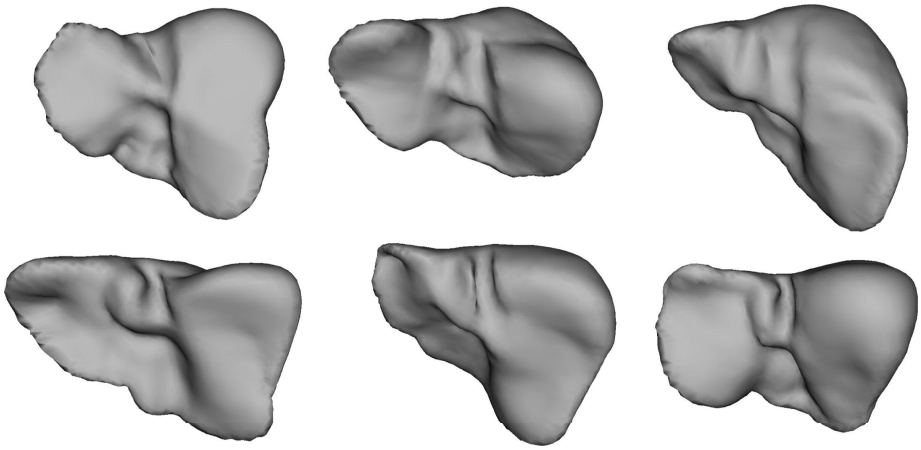


Figure 2.4: First three modes of shape model of 3D liver [Heimann 2009].

where $\mathbf{P} = (\mathbf{P}_1, \cdots, \mathbf{P}_k)$ is the matrix of the first k eigenvectors, and $\mathbf{b} = (b_1, \cdots, b_k)^T$ is a vector of weights, referred to as the *shape parameters*. We note that the number k is significantly smaller than the number of the dimension nd. Varying the parameters $\mathbf{b}$ can generate new examples of the shape. The interval values of $\mathbf{b}$ are imposed to constrain the resulting new shape to be valid. We show an example of point distribution model of 3D liver in Figure 2.4, where the columns from left to right represent the first, the second and the third largest eigenmodes, and the rows from top to bottom represent the resulting shapes taking the variation parameter as $3\sqrt{\lambda_k}$ and $-3\sqrt{\lambda_k}$ respectively.

Now given an image, the instance $\mathbf{y}$ of the model in the image is defined by a similarity transform T and the shape parameter vector $\mathbf{b}$.

$$\mathbf{y} = T(\bar{\mathbf{x}} + \mathbf{P}\mathbf{b}) \tag{2.11}$$

In order to find both the transform T (also called as *pose parameters*) and the shape parameters $\mathbf{b}$, an iterative method is used given an initial model state. At each iteration, a current model state $\mathbf{y}$ is known in the image space. First, an optimal displacement of each model point is calculated according to image observations. This leads to a vector of a suggested movement of the model $\mathbf{dy}$ in the image space. Second, the pose T is adjusted by a Procrustes match of the model to $\mathbf{y} + \mathbf{dy}$, leading to a new transformation T and a new residual displacements $\mathbf{dy_s}$. Next, $\mathbf{dy_s}$ is transformed into model space and then projected into the parameter space to give the optimal parameter updates:

$$\mathbf{db} = \mathbf{P}^T \hat{T}^{-1}(\mathbf{dy_s}) \tag{2.12}$$

where $\hat{T}$ is equal to T but without the translation part. After updating $\mathbf{b}$, a new model example is generated and used to update the state of the model in the image. In this way, only deformations that are similar to the shapes in the training set are allowed. This procedure is repeated until the changes of pose and shape parameters become insignificant.

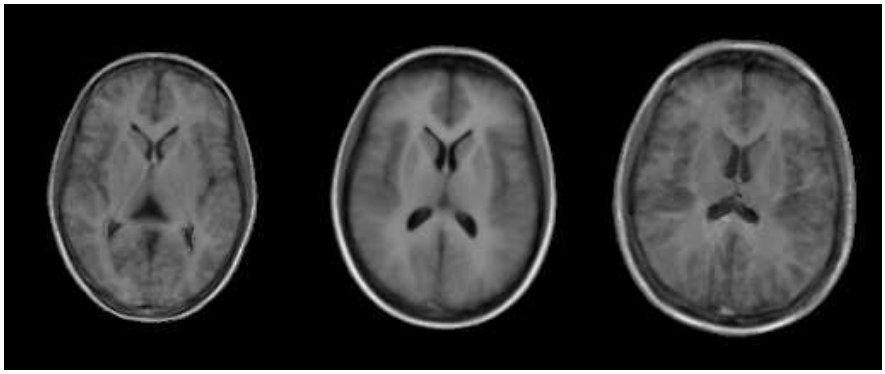
In order to improve the image appearance, variants of the ASM use different features going beyond simple reasoning on intensity. [Jiao 2003] uses Gabor wavelets and models the feature distribution by Gaussian mixture models. [Langs 2006] employs the steerable features to represent the object appearance. Beside Gaussian mixture models, other non-linear models are also used for modeling the appearance distribution. [De Bruijne 2003] proposes a non-parametric appearance model which is trained on both true and false examples of boundary profiles and the probability of a given image profile being part of the boundary is obtained using k nearest neighbor (kNN) probability density estimation. Similarly, [Van Ginneken 2002] uses the non-linear kNN classifier to estimate if the point is inside or outside of the object. [Li 2004, Li 2005] use Adaboost algorithm to build appearance models.

Parallel to the feature space, efforts have been made on the ASM search schemes. [De Bruijne 2004] combines the PDM with a maximum likelihood shape inference, where the optimal solution can be found using particle filtering in an iterated likelihood weighing scheme. The use of a large number of hypotheses makes segmentation by shape particle filtering robust to local maxima and independent of initialization at the expense of increasing computational cost. [De Bruijne 2005] is the extension work of segmenting multi-objects using particle filters. Another direction of improving the search scheme is to incorporate MRF regularization. [Behiels 1999] incorporates a regularization constraint penalizing outlier configurations that is minimized using a dynamic programming algorithm. [Tresadern 2009] proposes a method that combines an MRF-based local shape model for guided candidate selection with a PCA-based global shape model for regularization. Given a new image, the shape estimation involves an alternating scheme: first an MRF inference technique selects the best candidates for each point, then they are used to update the parameters of the global pose and shape model.

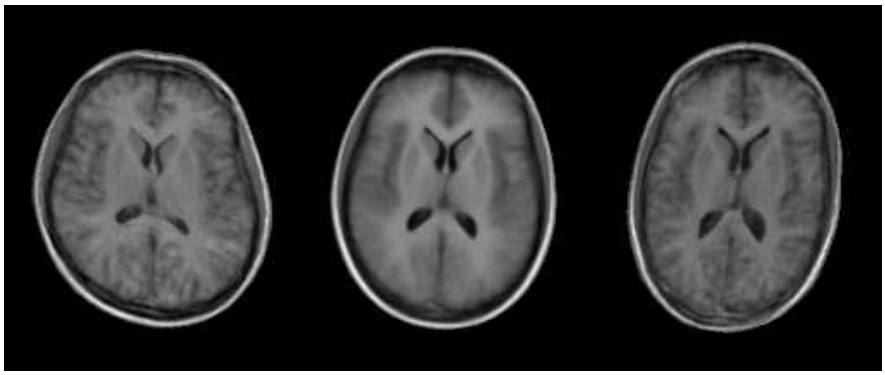
Active Appearance Models

Active appearance model (AAM) [Cootes 1998, Cootes 2001] is an extension to ASM, where the prior model is constructed using shape and appearance information. Similar to PDM that captures the mean shape and the shape variations, AAM encodes an appearance model consisting the mean appearance of the object and its variations, thus it can generate realistic images of the modeled data.

To build a statistical appearance model, each image in the training set is at first warped



c_1 varies by ± 2 s.d.s



c_2 varies by ± 2 s.d.s

Figure 2.5: First two modes of appearance model of full brain cross-section from an MR image [Cootes 1999a].

so that its control points match the mean shape obtained through the PDM procedure of the ASM. After intensity normalization on the shape-normalized images, a PCA is applied to analyze the gray-level variations with a linear model:

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{b}_g \tag{2.13}$$

where $\bar{\mathbf{g}}$ is the mean normalized gray-level vector, $\mathbf{P}_g$ is a matrix consisting of significant modes and $\mathbf{b}_g$ is a vector of gray-level parameters. We show an appearance model of brain images in Figure 2.5, where each row represents a variation mode. As described previously in the ASM, an instance shape is given as

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{b}_s \tag{2.14}$$

where $\mathbf{P}_s$ denotes the principal shape variations and $\mathbf{b}_s$ denotes the shape parameters.

Then, the shape parameters and the gray-level parameters are combined into a single vector $\mathbf{b} = (\mathbf{W}_s\mathbf{b}_s, \mathbf{b}_g)^T$, where $\mathbf{W}_s$ is a diagonal matrix of weights taking account of the units differences between shape and gray-level parameters. Because the shape and gray-level parameters may have correlations, a further PCA is applied on the vector $\mathbf{b}$ obtained from the training set, yielding a further linear model $\mathbf{b} = \mathbf{Q}\mathbf{c}$, where $\mathbf{Q}$ is a set of eigenvectors and $\mathbf{c}$ is a vector of *appearance parameters* that control both shape and gray-level pattern of the model. As a result, the shape model and gray-level model can be represented by a common parameter vector $\mathbf{c}$:

$$\begin{aligned} \mathbf{x} &= \bar{\mathbf{x}} + \mathbf{P}_s\mathbf{W}_s^{-1}\mathbf{Q}_s\mathbf{c} \\ \mathbf{g} &= \bar{\mathbf{g}} + \mathbf{P}_g\mathbf{Q}_g\mathbf{c} \end{aligned} \tag{2.15}$$

where $\mathbf{Q} = (\mathbf{Q}_s, \mathbf{Q}_g)^T$ has two submatrixs $\mathbf{Q}_s$ and $\mathbf{Q}_g$ corresponding to the shape and gray-level parameters respectively.

Given an image, an instance of the model in the image is defined by a similarity transform T and the appearance parameter vector $\mathbf{c}$, combined in a unique parameter vector $\mathbf{p}$. The key idea of AAM search is to adjust the parameters in $\mathbf{p}$ so that the difference between the given image and a synthetic example generated by the appearance model is as small as possible. The image difference has to be calculated within the same reference frame, thus the given image with a model presence $T(\mathbf{x})$ is warped to the mean shape and normalized, resulting a texture vector $\mathbf{g}_s$. The residual is given by $\mathbf{r}(\mathbf{p}) = \mathbf{g}_s - \mathbf{g}_m$, where $\mathbf{g}_m$ is generated from the appearance model. Since it is difficult to adjust the parameter vector $\mathbf{p}$ to minimize the residual error $|\mathbf{r}(\mathbf{p})|^2$ due to its dimensionality, the process to adjust the model parameters during the image search is learned in advance. A linear model is chosen for the relationship between the texture residual $\mathbf{r}(\mathbf{p})$ and the parameter updates $\mathbf{dp}$:

$$\mathbf{dp} = -\mathbf{R}\mathbf{r}(\mathbf{p}) \tag{2.16}$$

where $\mathbf{R}$ is the derivative matrix learned from the training set. It can be computed using multiple multivariate linear regression or using numeric differentiation. After learning the correction of the model parameters, an iterative algorithm is used to solve the optimization.

To speed up the active appearance model fitting, [Matthews 2004] uses the inverse compositional image alignment algorithm so that the effects of appearance variation during fitting can be precomputed. [Andreopoulos 2008] extends the method to 3D cases and demonstrates the framework for cardiac MR image analysis. [Donner 2006] introduces a fast AAM search algorithm based on canonical correlation analysis (CCA-AAM) which efficiently models the dependency between texture residuals and model parameters during search. Active appearance models remains an active topic where for example, variants to deal with illumination and viewpoint variation [Gross 2005] as well as occlusions [Gross 2006] have been introduced.

2.1.3 Graph-based Methods

Graph-based approaches have been developed in the last decade. It considers the image segmentation as a graph partition problem, where an image is represented as a weighted undirected graph $\mathbf{G} = (\mathbf{V}, E)$. The nodes of the graph $\mathbf{V}$ represent the image pixels and the edges E consist of pairs of nodes, while the weight of each edge $w(i, j)$ is a similarity measurement between nodes i and j. To group the pixels, a graph partition is sought to separate the node set $\mathbf{V}$ into disjoint sets $\mathbf{V}_1, \ldots, \mathbf{V}_m$, so that the similarity among the nodes in the set $\mathbf{V}_i$ is high while the similarity across different sets $\mathbf{V}_i, \mathbf{V}_j$ is low.

Normalized Cuts

[Shi 2000] proposes a graph-theoretic criterion for measuring the goodness of an image partition, named *normalized cut*. Assuming a graph partitions into two disjoint sets A, B, the dissimilarity between the two groups can be computed as total weight of the edges that have been removed, called the *cut* between A and B:

$$cut(A, B) = \sum_{i \in A, j \in B} w(i, j) \tag{2.17}$$

The optimal partition can be considered as the one that minimizes the cut value. However, using the *minimum cut* criteria favors cutting small sets of isolated nodes in the graph. Instead of using the cut value, the *normalized cut (Ncut)* computes the cut cost as a fraction of the total edge connections to all the nodes in the graph as disassociation measure:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, \mathbf{V})} + \frac{cut(A, B)}{assoc(B, \mathbf{V})} \tag{2.18}$$

where $assoc(A, \mathbf{V}) = \sum_{i \in A, j \in \mathbf{V}} w(i, j)$ is the total connection from the nodes in A to all nodes in the graph. Using this definition of the disassociation between the groups, the smallest cuts which isolate a single pixel do not have small *Ncut* value. Unfortunately, minimizing the normalized cut is NP-complete. [Shi 2000] embeds the normalized cut problem in the real-valued domain so that an approximate discrete solution can be found efficiently. Assuming a graph partitions into two sets A and B, let $\mathbf{x}$ be an N-dimensional indicator vector, where $x_i = 1$ if node i belongs to A, and $x_i = -1$ otherwise. Let $\mathbf{W} = [w(i, j)]$ be an $N \times N$ symmetrical matrix, and $\mathbf{d} = \mathbf{W1}$ is the row sums of the matrix $\mathbf{W}$, and $\mathbf{D}$ is a diagonal matrix with $\mathbf{d}$ on its diagonal. Minimizing the normalized cut over all possible indicator vectors $\mathbf{x}$ is equivalent to:

$$\min_{\mathbf{y}} \frac{\mathbf{y}^T(\mathbf{D} - \mathbf{W})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}} \tag{2.19}$$

where $\mathbf{y} = (\mathbf{1} + \mathbf{x}) - b(\mathbf{1} - \mathbf{x})$ and $\mathbf{y}^T\mathbf{D}\mathbf{1} = 0$. The above Rayleigh quotient can be minimized by solving the generalized eigenvalue system:

$$(\mathbf{D} - \mathbf{W})\mathbf{y} = \lambda\mathbf{D}\mathbf{y} \tag{2.20}$$

The second smallest eigenvector $\mathbf{y}$ of the generalized eigenvalue system is the real valued solution (an approximation in real-valued domain) to the normalized cut problem. Thus the proposed normalized cut criterion for graph partition can be computed efficiently by solving a generalized eigenvalue problem.

In order to accelerate the computation of the normalize cuts, [Sharon 2006] proposes a segmentation method of weighted aggregation which is derived from algebraic multigrid solvers and it consists of fine-to-coarse pixel aggregation. [Alpert 2007] presents a segmentation approach of probabilistic bottom-up aggregation and cue integration. The probabilistic approach is integrated into a graph coarsening scheme, providing a complete hierarchical segmentation of the image.

Graph Cuts

The segmentation problem can be formulated as a classic pixel-based energy function which encodes boundary measurements and regional statistics such as the continuous approaches [Mumford 1989], [Chan 2001]. However, the optimization of the continuous approaches using iterative gradient descent techniques is slow and prone to get trapped in local minima. [Boykov 2001b] is the first to apply Markov random fields (MRFs) optimization to binary segmentation. The main advantage of their segmentation method is that it provides the global optimal of segmenting an N-dimensional image using graph cuts.

Their interactive segmentation includes a user indication of certain pixels (seeds) being part of the object and certain pixels (seeds) being part of the background respectively, as hard constraints of segmentation. Moreover, the cost function of segmenting the rest of the image is defined in terms of boundary and regional properties of the segments, which can be considered as soft constraints of segmentation. The segmentation problem is equivalent to compute the global minimum of the cost function among all segmentations that satisfy the hard constraints imposed by a user. The globally optimal segmentation can be efficiently achieved by powerful graph cut algorithms. To segment a given image, an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is created with two terminals:

$$\mathcal{V} = \mathcal{P} \cup \{S, T\} \tag{2.21}$$

We illustrate the graph in Figure 2.6, where the gray nodes are the nodes $\mathcal{P}$ corresponding to image pixels, while the red node is the source node S representing an "object" terminal and the blue node is the sink node T representing a "background" terminal. The edge set

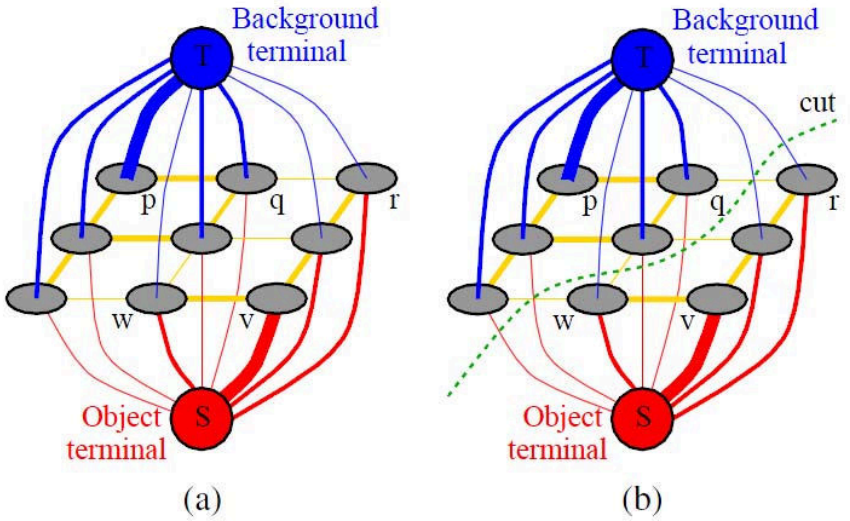


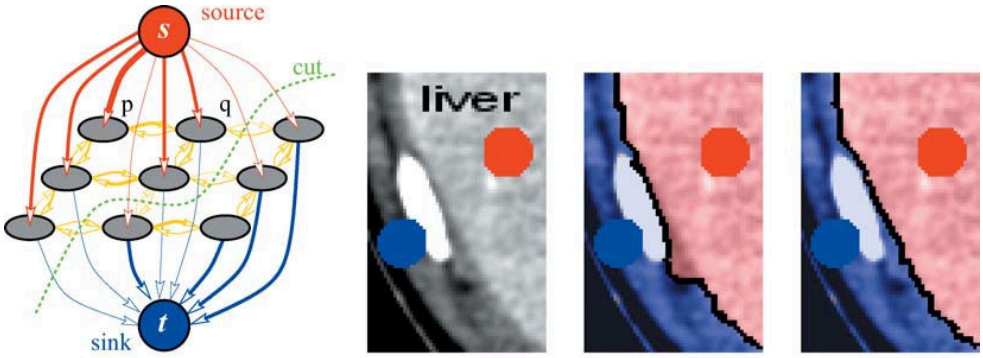
Figure 2.6: Graph cuts for N-D image segmentation [Boykov 2001b]. (a) Graph. (b) The minimum cut.

$\mathcal{E}$ consists of two types of undirected edges: *n-links* and *t-links*. Each pixel node has two t-links $\{p, S\}$ (red edges) and $\{p, T\}$ (blue edges) connecting to each terminal. Each pair of connected pixel nodes $\{p, q\}$ is an n-link (yellow edges) in a neighborhood system $\mathcal{N}$.

$$\mathcal{E} = \mathcal{N} \bigcup_{p \in \mathcal{P}} \{\{p, S\}, \{p, T\}\} \tag{2.22}$$

The edge weights of n-links encode the boundary terms of the cost function, while the weights of t-links encode the regional terms of the cost function for non-seed pixels or they are defined as hard constraints for seed pixels. In Figure 2.6, the weight of each edge is reflected by its thickness. Given the graph $\mathcal{G}$, the image segmentation can be solved by finding the minimum cut on the graph. The globally optimal minimum cut separating two terminals can be computed in polynomial time using the new version of "max-flow" algorithm [Boykov 2004]. Figure 2.6 (b) shows the minimum cost cut (green dashed line) of the graph in (a).

The binary segmentation framework of [Boykov 2001b] has been extended in different directions. For instance, [Rother 2004] proposes a *GrabCut* algorithm which has made three improvements: (1) Gaussian Mixture Models (GMM) are used to model the object and the background in RGB space. (2) An iterative energy minimization scheme is employed to alternate between estimation and GMM parameter learning until convergence. (3) The iterative minimization allows a considerably reduced degree of user interaction.



(a) directed graph (b) image (c) undir. result (d) dir. result

Figure 2.7: Segmentation via cuts on a directed graph [Kolmogorov 2005].

Specifically, the user interaction involves simply dragging a rectangle loosely around the object of interest. More recently, [Cui 2008] integrates a local color pattern model and edge model in the graph-cut framework in order to improve robustness and enhance the discriminability of the method.

[Kolmogorov 2005] presents an s/t cut on a directed graph which depends on the cut's orientation. One of their major contributions is that flux of vector fields is represented as the directed edge weights and it is integrated into the global optimization of graph cuts, and thus discrete cut metrics on directed regular grids have a geometric interpretation via standard continuous concepts of length/area and flux. Figure 2.7 shows an example of a directed graph, where each pair of the neighboring nodes is associated with two directed edges (p, q), (q, p) and different weights $w(p, q)$, $w(q, p)$. If a cut separates two neighboring nodes p and q so that p is connected to the source S and q is connected to the sink T, then $w(p, q)$ is considered for the cost of the cut. A comparison of the segmentation results using indirect graph and directed graph is shown in Figure 2.7 (c,d). [Boykov 2006] reviews a large number of known extensions of s/t graph cut algorithms for object segmentation.

To overcome the "shrinking bias" problem of graph cut, [Vicente 2008] imposes an additional connectivity prior on the graph cut segmentation. Several versions of the connectivity constraints are considered, but the corresponding optimizations are all NP-hard. They propose two optimization methods: (1) a heuristic algorithm, named *DijkstraGC* which merges the Dijkstra algorithm and graph cut, and (2) a slow method based on dual decomposition which provides a lower bound on the problem. Some practical examples show that DijkstraGC is able to find the global minimum.

Random Walker

[Grady 2006] introduces the *random walker* algorithm for multi-label interactive image segmentation. Given a small number of pixels (seeds) with user-defined labels, the probability that a random walker starting at each unlabeled pixel first reaches one of the seeds is calculated. In particular, each unlabeled pixel has a K-dimensional vector while the $i \in \{1, \cdots, K\}$-th component specifies the probability that a random walker starting at this location would reach the i-th seed first. Then, each unseeded pixel is assigned to the label of the most probable seed that the random walker reaches, generating the image segmentation. Given a graph where the nodes represent the image pixels and the edges represent the pairs of pixels, each edge is associated with a real-valued weight corresponding to the likelihood that a random walker crosses that edge. The desired probability of a random walker first reaching a seed point equals to the solution of the combinatorial Dirichlet problem. [Grady 2005] extends random walkers segmentation by incorporating a nonparametric probability density model, so that it can locate disconnected objects and does not require user-defined labels. [Grady 2008] performs a precomputation of eigenvectors of the weighed Laplacian matrix of a graph and uses this information to produce a linear-time approximation of the Random Walker segmentation.

The graph cuts and the random walker algorithms are closely related. [Sinop 2007] presents a general seeded image segmentation algorithm which can take the form of either Graph Cuts or Random Walker algorithms, depending on the choice of the norm by which the gradient of the potential function is minimized. Using l_1 norm, the algorithm is equivalent to Graph cuts, whereas using l_2 norm it leads to the Random Walker algorithm. [Couprie 2009] also extends a common framework that includes the graph cuts, random walker and shortest path optimization algorithms. [Singaraju 2009] continues the same direction and explores image segmentation using continuous-valued Markov random fields (MRFs) with probability distributions following the p-norm of the difference between configurations of neighboring sites. They use integrative reweighted least squares (IRLS) techniques to find the global minimizer of the proposed cost function chosen $1 < p < 2$ so that amenable trade off can be achieved between Graph Cuts and Random Walker.

Incorporating Shape Priors with Graph Cuts

The graph cuts suffers from metrication artifacts and the shrinking bias. The random walker can overcome these problem, but it is biased towards the location of seeds. More recent development in graph-based techniques is to incorporate shape priors. Since knowledge about the object shape is used, the resulting segmentation is robust to the shape to be segmented as well as to the user interaction.

[Slabaugh 2005] presents a graph cuts based image segmentation that incorporates an

elliptical shape prior. The shape constraint is encoded in the terminal weights as unary terms of the pixels, and it is defined by a binary mask from the ellipse shape. An iterative approach is used to update the shape and the graph cut: given an initial ellipse, the shape mask is generated, and the mean intensities of the pixels inside and outside the mask are computed. Then the graph cut is applied with the regional terms using the mean intensity information as well as shape prior terms using shape mask. The result of the graph cut is used to update the ellipse shape. The process continues until the solution converges.

[Funka-Lea 2006] proposes an automatic segmentation of the entire heart in Computer Tomography (CT) cardiac scans based on graph cut with automatic determination of seed-regions. Given a point within the heart, the segmentation is initialized as an ellipsoid of maximum volume within the heart. An additional "blob" constraint is added to the graph-cut formulation as Potts interaction. It encourages cuts to produce the shape where the edges are oriented perpendicular to the direction toward the center of the seed-region. This simple shape prior information (the heart is a compact blob) prevents segmentation leaking into the aorta or pulmonary vessels.

[Veksler 2008] imposes a generic shape prior called *star* on graph cut segmentation, since a star shape is defined with respect to a center point given by a user. The advantage of the star shape prior is that it can be directly include in the objective function as length-based "ballooning" term (a pairwise term of neighboring pixels) that encourages a large object segment. This alleviates the bias of a graph cut towards short segmentation boundaries. Compared to the standard graph cut which requires a number of seeds, this method only requires a single pixel which is often automatically obtained. Similarly, [Das 2009] incorporates the *compact* shape prior in the graph cut segmentation, assuming that the object to be segmented can be approximately by several connected roughly collinear compact pieces. Due to the shape prior, a bias parameter is introduced, allowing them to counteract the shrinking bias of the graph cut segmentation.

[Freedman 2005] uses a fixed template to represent the shape prior. The template is specified as a distance function ϕ whose zero level set corresponds to the template, and the shape energy can be defined by n links:

$$E_{shape} = \sum_{(p,q)\in\mathcal{N}:x_p\neq x_q} \phi(\frac{p+q}{2}) \tag{2.23}$$

where each pair of neighboring pixels p, q in the neighborhood system $\mathcal{N}$ is penalized by the distance if they have different labels. This energy constraints the boundary of the segmented object to lie near the shape template. In order to deal with rigid transformations of the template, the user input is required to match the template to the data via Procrustes Method. Once the optimal rigid transformation is computed, a gaussian pyramid of the image is used. The best segmentation among all the scales is obtained by comparing the

optimal energies of each pyramid level using the template of the fixed scale. The minimum cut problem is solved by the maximum flow algorithm.

[Slabaugh 2005] shows how highly variable nonlinear shape priors can be added to existing iterative graph cut methods. A statistical shape model is learned from the training set using kernel Principle Component Analysis (KPCA). Given a user-initialized segmentation, the algorithm is operated iteratively. At each iteration, the intensity histograms of object and background are computed given the segmented regions, while the current segmentation is used to obtain a projection in the learned shape space. Then these priors including the shape and histograms are used in a Bayesian formulation to perform segmentation via the graph cut technique.

[Ali 2007] proposes a graph cut based segmentation approach which combines image appearance and shape priors. A template image is generated from a set of aligned images, consisting of three segments: object, background, and shape variability region. They estimate the shape variations using a distance probabilistic model which approximates the distance marginal densities of the object and the background inside the variability region. To segment a new image, they align it with the training images in order to use the probabilistic template. The shape prior is encoded as terminal weights for each pixel, measuring how much the pixel labeling disagrees with the shape information.

[Vu 2008] defines the shape prior using a discrete version of the shape distance proposed by [Chan 2005] of level sets framework, and incorporates it as terminal weights of the graph. A multiphase graph cut framework is proposed to segment multiple objects, where a pixel is allowed to have multiple labels. Then the shape prior energy is extended to encompass multiple shape priors. The segmentation is performed in an integrative manner. In each iteration dealing with one object, the shape energy is computed based on the aligned template, while the data energy is computed to account for the overlap between objects. A new labeling for this object is obtained by the min-cut solution. This process is repeated for all the objects until convergence is reached. However, the iterative algorithm is not guaranteed to converge to the global optimum.

[Schoenemann 2007] presents a globally optimal image segmentation with a translation invariant elastic shape prior. They compute cycles of minimal ratio in a large graph representing the product space spanned by the input image and all points of the shape template. The specific structure of the graph allows for run-time and memory efficient implementations. Recently, [Ayed 2009] proposes a discrete kernel density matching energy for left ventricle segmentation. Given a manual segmentation of the first frame, the algorithm propagates the segmentation to the other frames using two priors, geometric (distance-based) one and photometric one, each measuring a distribution similarity between the segmented region and a model of the first frame. An original first-order approximation of the Bhattacharyya measure yields a global graph cut optimum in nearly real-time.

Model-based Methods Using Graph Representation

The above mentioned graph-based segmentation methods use Markov Random Fields to represent the image labeling, where each node variable of the graph represents a pixel label assignment. In order to incorporate shape priors, most of the approaches impose the shape priors through terminal weights. Typically, the shape prior is represented as a template such as a distance map [Freedman 2005], a probabilistic shape image [Ali 2007] (statistical shape prior). However, the segmentation approaches which integrate shape priors in graph cut algorithm are usually implemented in an iterative optimization scheme in order to couple the shape parameters and the MRF variables. This framework does not guarantee to achieve global optimum. In addition, the new image has to be aligned to the same coordinate system of the template. It is an ill-posed problem since the global pose of the object to be segmented is unknown. Alternatively, there exists another class of segmentation approaches based on graphical model which combine shape priors in a top-down fashion.

[Zhang 2004] presents a graph-based method of localizing the articulated human body. The body contour is represented by a Bayesian graphical model, where the nodes correspond to point positions along the shape contour. The shape priors include both local non-rigid deformation and rotation motion of the joint, while the image likelihood includes edge gradient map, foreground/background mask, skin color mask, and appearance consistency constraints. The constructed Bayes model is sparse and chain-like, thus Bayesian formula can be optimized by efficient spatial inference through Sequential Monte Carlo sampling methods. However, the specific shape model is not general to other applications.

[Felzenszwalb 2005] describes how to represent and detect generic deformable shapes. They represent the shape by triangulated polygons which decompose the complex shape into simple parts. The involved triangles that decompose a polygon without holes are connected together in a tree structure (called a chordal graph) which yields a discrete representation closely related to medial axis transform. The polygon model is used to detect non-rigid objects in new image using boundary information, while the shape prior is defined by the independent triplets. The detection algorithm can efficiently provide a global optimal solution to the deformable template matching problem due to the elimination scheme property of the graph. However, this method does not generalize to higher dimensions (*e.g.* 3D cases).

[Seghers 2008] presents a model-based segmentation using graph representations. The object is represented as a graph where the nodes correspond to the landmarks and the edges define the landmark dependencies. The shape prior is described as a concatenation of the local constraints of connected landmarks. The segmentation problem is formulated by a maximum a posteriori (MAP) criterion, thus the objective function includes an intensity

energy of each landmark and a local shape energy of each edge. The discretization of the objective function transforms the segmentation as a labeling problem where one candidate per landmark needs to be selected. It enables robust optimization techniques such as mean field annealing and dynamic programming techniques.

[Donner 2010] proposes a framework of localizing an object model as the solution to the optimal labeling task of a Markov Random Field (MRF). The prior information about the geometric configuration of landmarks and local appearance features is built by *Sparse Appearance Models*. The proposed rotation invariant local descriptors based on Gradient Vector Flow (GVF) capture local appearance details as well as global structure that allows stable detection and identification of individual points. The MRF combines costs of non-rigid deformation and local descriptor feature difference between the target and the model, and it is solved efficiently in a single optimization step using the max-sum algorithm.

2.2 Markov Random Fields and Optimization

As we have discussed above, a growing number of graph-based approaches have been developed for image segmentation. One of the main reasons behind their popularity is the availability of efficient algorithms for Markov Random Field (MRF) inference problem, which in turn allows for the computation of globally optimal solutions of the MRF energy functions. In this section, we review the MRFs and the related optimization techniques.

A *Markov Random Field* is a set of random variables having a Markov property described by an undirected graph. Given an undirected graph $G = (\mathcal{V}, \mathcal{E})$ which consists of a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$, a set of random variables $\mathbf{X} = \{X_i\}_{i\in\mathcal{V}}$ form a Markov random field with respect to G if they satisfy the local Markov property:

$$X_i \perp X_{\mathcal{V}\setminus\{i\}} | X_{\mathcal{N}_i} \tag{2.24}$$

which states that a variable $X_{i\in\mathcal{V}}$ is conditionally independent of all other variables given all it neighbors $\mathcal{N}_i = \{j|\{i,j\} \in \mathcal{E}\}$ in the graph G.

A graphical concept called *clique* is defined as a subset of the nodes $c \subseteq \mathcal{V}$ if every two nodes in this subset c are connected by an edge. A *maximal clique* is a clique of the largest possible size in a given graph. Let $p(\mathbf{X} = \mathbf{x})$ denote the probability of the random variables $\mathbf{X}$ taken a particular field configuration $\mathbf{x}$. When $\mathbf{X}$ forms a Markov random field with respect to G, the joint density $p(\mathbf{x})$ can be factorized over the maximal cliques of the graph.

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{c\in C} \phi_c(\mathbf{x}_c) \tag{2.25}$$

where Z is the normalization constant, C is the set of cliques of G. Each clique $c \in C$ consists of the set of variables $\mathbf{x}_c$, and it is associated a potential function $\phi_c(\mathbf{x}_c) \geq 0$. MRFs were introduced to computer vision domain by [Geman 1984] providing a probabilistic framework where prior knowledge can be integrated as local neighborhood interactions. In addition, MRF models can be defined over discrete variables such as pixel labels. Thus the computer vision task can be formulated as a maximum a posteriori (MAP) estimation:

$$\mathbf{x}^{\text{opt}} = \arg\max_{\mathbf{x}} p(\mathbf{x}) \tag{2.26}$$

According to the Hammersley-Clifford theorem, for a Markov random field, the probability $p(\mathbf{x})$ follows a Gibbs distribution:

$$p(\mathbf{x}) = \frac{1}{Z} \exp\{-E(\mathbf{x})\} \tag{2.27}$$

where the MRF energy $E(\mathbf{x})$ can be factorized into the clique potentials:

$$E(\mathbf{x}) = \sum_{c \in C} \theta_c(\mathbf{x}_c) \tag{2.28}$$

where the MRF potential functions $\theta_c(\mathbf{x}_c) = -\log \phi_c(\mathbf{x}_c)$ are the negative logarithm of the ones defined for the probability distribution $p(\mathbf{x})$. Thus the MAP inference is equivalent to the minimization of the MRF energy $E(\mathbf{x})$.

Pairwise MRFs

The *pairwise* MRFs are the most widely used form in computer vision, where each clique includes no more than two variables. The pairwise MRF energy consists of unary potentials on single variable and pairwise potentials on pairs of the variables.

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(x_i, x_j) \tag{2.29}$$

Considering the labeling problem in discrete case, the unary potentials $\theta_i(x_i)$ measure the cost of assigning the particular label to the node given the observation related to the node, while the pairwise potentials $\theta_{ij}(x_i, x_j)$ encode the dependencies of the two labels corresponding the neighboring nodes.

In computer vision, the pairwise MRFs have been widely represented as *grid-like* structures and *pictorial* structures. In the first case, the nodes of a graph correspond to the lattices of pixels while the edges correspond to the neighborhood systems (4-connected or 8-connected) of the pixels. The variable x_i of each node represents a physical quantity

of a specific problem, *e.g.* a label indicates the class of the pixel in image segmentation problem. The unary and pairwise potential functions are designed for the specific problem such as image denoising/restoration [Geman 1984, Greig 1989], and image segmentation [Boykov 2003, Rother 2004, Boykov 2006]. On the other hand, MRFs of pictorial structures [Felzenszwalb 2005, Sigal 2006] provide a part-based representation for deformable objects, where the nodes represent different parts of the object, the edges represent the interactions of the pair of the parts, and the random variables of the nodes represent a physical state of the parts.

Higher-order MRFs

More recently, *higher-order* MRFs haven been studied to model more complex interactions between the random variables. The cliques defined by the higher-order MRFs can contain more than two nodes and they can not be factorized into lower orders. The higher-order MRF energy can be written as:

$$E(\mathbf{x}) = \sum_{k=1}^{K} \sum_{c \in \mathcal{C}_k} \theta_c(\mathbf{x}_c) \tag{2.30}$$

where K determines the order of the MRF, and C_k denotes the set of cliques where each clique consists of k nodes.

Higher-order models have been applied in image denoising [Roth 2005, Roth 2009] and image segmentation [Kohli 2009a, Kohli 2009b]. The main advantage of introducing higher-oder MRFs is that better prior knowledge can be incorporated, since higher order interactions can capture the intrinsic spatial properties which two variables can not obtain. For example, [Kwon 2008] uses a third-order spatial prior for image registration, while [Glocker 2010] uses higher-order potentials in optical flow formulation. Global models which include interaction of all the nodes, have been proposed with related inference algorithm. [Vicente 2008], [Nowozin 2009] and [Delong 2012] are some examples.

Conditional Random Fields

Conditional Random Field (CRF) was introduced by [Lafferty 2001] for text modeling, and then introduced to computer vision domain by [Kumar 2003]. The difference between MRF and CRF is that the latter models a posterior distribution $p(\mathbf{X}|\mathbf{D})$ where $\mathbf{X}$ denotes a set of latent variables and $\mathbf{D}$ denotes a set of observed variables. It can be interpreted as an MRF globally conditioned on the observed data $\mathbf{D}$. The conditional distribution $p(\mathbf{X}|\mathbf{D})$ is also a Gibbs distribution and it can be written as:

$$p(\mathbf{x}|\mathbf{D}) = \frac{1}{Z(\mathbf{D})} \exp\{-E(\mathbf{x}, \mathbf{D})\} \tag{2.31}$$

where the CRF energy takes the following form:

$$E(\mathbf{x}, \mathbf{D}) = \sum_{c \in \mathcal{C}} \theta_c(\mathbf{x}_c, \mathbf{D}) \tag{2.32}$$

One of the main advantages of using CRF is that it provides the ability of modifying the prior model based on the observed data. For example, [Boykov 2001b] uses this idea for interactive segmentation. They modulate the smoothness weights by the intensity gradients, leading to a conditional random filed (CRF). [Blake 2004, Rother 2004] are some other examples for image segmentation. In addition to relating smoothness terms with the observed data, [Kumar 2003] proposes *Discriminative Random Fields* (DRF), where the unary MRF data term $\theta_i(x_i, d_i)$ is modified by a neighborhood function over the data as $\theta_i(x_i, \mathbf{D})$. [Kumar 2006] also mentions that CRFs and DRFs are easier to learn the model parameters than MRFs.

Despite the differences in the probabilistic explanation and for the purpose of convenience, we use the term MRF to refer to the generative MRF including CRF and DRF in the following context. One of the main challenges of using MRF models is to develop efficient inference algorithms. Now We review the most widely used MRF inference techniques.

2.2.1 Graph Cuts

Graph cuts is a large family of MRF inference algorithms based on solving one or more *min-cut* or *max-flow* problems [Boykov 2001a, Boykov 2006]. [Greig 1989] was the first to use a s/t graph cut to perform the exact minimization of a binary MRF in polynomial time. Basically, in the min-cut graph, the nodes in an MRF are connected to two additional terminal nodes (*i.e.* source node s and sink node t) which represent two classes of the binary labeling. Each edge in the graph is associated with non-negative capacity or a weight. To optimize the MRF energy is equivalent to find the minimum cut between the terminal nodes.

[Ford 1962] proves that the solution of the min-cut problem is equivalent to the solution of finding the maximum flow from the source s to the sink t via the capacitated edges. The min-cut/max-flow problem can be solved in polynomial time with augmenting paths methods [Ford 1962] and push-relabel methods [Goldberg 1988]. [Boykov 2004] presents a modified augmenting paths algorithm which can obtain the best results. [Kolmogorov 2004] demonstrates that the global optimum of the energy can be guaranteed by using graph cuts if the MRF energy function is *submodular*. Considering a binary case, the energy of a pairwise discrete MRF is submodular if each pairwise potential θ_{ij} satisfies:

$$\theta_{ij}(0,0) + \theta_{ij}(1,1) \leq \theta_{ij}(0,1) + \theta_{ij}(1,0), \quad (i,j) \in \mathcal{E} \tag{2.33}$$

However in many vision problems, the energy functions can not satisfy the submodular condition, and the minimization problem remains NP-hard. [Kolmogorov 2007] reviews the *quadratic pseudo-boolean optimization* (QPBO) algorithm [Hammer 1984] which can achieve a partial optimal labeling for Arbitrary functions (with both submodular and non-submoddular terms). To construct the graph, besides the two terminal nodes, each pixel $p \in \mathcal{V}$ corresponds to two nodes p and $\bar{p}$. Ideally, the variable $x_{\bar{p}}$ should be the negation of x_p, *i.e.* $x_{\bar{p}} = 1 - x_p$. The important observation is that the new energy of variables $\{x_p, x_{\bar{p}}\}$ is submodular, and thus can be minimized in polynomial time using min-cut/max-flow. The partial labeling $\mathbf{x}$ is determined as follows:

$$x_p = \begin{cases} 0 & \text{if } p \in S, \bar{p} \in T \\ 1 & \text{if } p \in T, \bar{p} \in S \\ \emptyset & \text{otherwise} \end{cases} \tag{2.34}$$

where $\emptyset$ means that the node is unlabeled. If the constraints that nodes p and $\bar{p}$ belong to different sets of the cut are enforced, then a global minimum of the energy can be obtained, otherwise a part of an optimal solution is found. Later, [Boros 2006] and [Rother 2007] present two different techniques in order to extend QPBO to achieve a complete solution.

Graph cut techniques have been extended in order to deal with multi-label MRFs problems. [Boykov 2001a] presents two algorithms α-*expansion* and α-β-*swap* for multi-dimensional energy minimization that uses graph cut iteratively. In contrast to standard moves which allow only one node variable to change its label at a time, α-expansion and α-β-swap moves allow a large number of pixels to change their labels simultaneously. In particular, given a label α, an α-expansion move allows any set of nodes to change their labels to α. Given two different labels α and β, an α-β-swap move defines that some pixels which were labeled α are now labeled β, and some pixels which were labeled β are now labeled α. The expansion algorithm finds a labeling within a known factor of the global minimum, while the swap algorithm can handle more general energy functions. Approximate solution to the NP-hard minimization problem can be achieved with guaranteed optimality bounds. More recently, [Komodakis 2008b, Rother 2007] provide approximate solution for more general energy functions, while [Kohli 2005, Juan 2006, Alahari 2010] propose efficient algorithms for dynamic MRFs.

2.2.2 Linear Programming Relaxation

Over the last years, significant progress in MRF optimization has been made by making use of *linear programming* (LP) relaxation of the MRF energies which was first introduced by [Shlezinger 1976]. In this context, the MRF optimization problem can be equivalently

formulated as a linear integer program as follows:

$$\min \sum_{p\in\mathcal{V}}\sum_{a\in\mathcal{L}} \theta_p(a)x_p(a) + \sum_{(p,q)\in\mathcal{E}}\sum_{a,b\in\mathcal{L}} \theta_{pq}(a,b)x_{pq}(a,b) \tag{2.35}$$

$$s.t. \quad \sum_a x_p(a) = 1, \forall p \in \mathcal{V} \tag{2.36}$$

$$\sum_a x_{pq}(a,b) = x_q(b), \forall b \in \mathcal{L}, (p,q) \in \mathcal{E} \tag{2.37}$$

$$\sum_b x_{pq}(a,b) = x_p(a), \forall a \in \mathcal{L}, (p,q) \in \mathcal{E} \tag{2.38}$$

$$x_p(\cdot), x_{pq}(\cdot,\cdot) \in \{0,1\} \tag{2.39}$$

where two types of binary indicator variables $x_p(\cdot)$ and $x_{pq}(\cdot)$ are introduced to linearize the MRF energy function. The unary variable x_p indicates which label is assigned to the MRF node p, *i.e.* $x_p(a) = 1$ indicating label a is assigned to p. Similarly, the pairwise variable x_{pq} indicates which pair of labels is assigned to the pair of nodes (p,q), *i.e.* $x_{pq}(a,b) = 1$ indicating label a is assigned to p and b is assigned to q. In order to be equivalent to the MRF formulation, the constraints on the binary variables have been added. The constraints in Eq.(2.36) make sure that only one label can be assigned to each node, while the constraints in Eq.(2.37, 2.38) enforce consistency between the unary and the pairwise variables so that if $x_p(a) = x_q(b) = 1$, then we have $x_{pq}(a,b) = 1$ as well. However, the above integer LP problem is NP-hard in general. The LP relaxation is formed by replacing the integer constraints in Eq.(2.39) with linear constraints:

$$x_p(\cdot), x_{pq}(\cdot,\cdot) \geq 0 \tag{2.40}$$

This relaxation can provide a good approximation to the integer LP problem, thus an approximately optimal solution to the MRF estimation problem can be obtained by solving the LP relaxation. However, general-purpose LP solvers can not handle large problems in computer vision. To this end, a number of approaches have been proposed to solve the dual problem of the LP relaxation which can provide a lower bound on the optimal solution of the primal problem. These approaches solve the LP problem by maximizing the lower bound provided by the dual. For example, one can cite the *max-sum diffusion* [Werner 2007], *tree-reweighted message passing* (TRW) [Wainwright 2005, Kolmogorov 2006], *dual decomposition* [Komodakis 2007a, Komodakis 2011b]. Although the algorithms based on LP relaxation generate excellent results for some vision problems, this is not guaranteed in all cases. More recently, researches have studied tightening of the LP relaxation [Sontag 2007, Komodakis 2008a, Schraudolph 2010] which produce tighter bounds of the original MRF inference at the expense of additional computational cost.

Primal-Dual Schema

[Komodakis 2007b] introduces the primal-dual schema (a powerful tool for deriving approximation algorithms to problems of integer LP) for MRF optimization. The primal-dual schema aims to find a pair $(\mathbf{x}, \mathbf{y})$ of primal and dual solutions such that the corresponding primal-dual gap is small enough (*e.g.* the ratio between the two costs is smaller than a factor f), then it is guaranteed that the primal solution $\mathbf{x}$ is an f-approximation to the unknown optimal solution of the original integer LP problem. The primal variables $\mathbf{x} = \{x_p\}_{p \in \mathcal{V}}$ denote the labels assigned to the MRF nodes. The dual to the LP relaxation of integer program can be written as follows:

$$\max \sum_{p \in \mathcal{V}} y_p \tag{2.41}$$

$$s.t. \;\; y_p \leq \min_{a \in \mathcal{L}} h_p(a) \tag{2.42}$$

$$y_{pq}(a) + y_{qp}(b) \leq \theta_{pq}(a, b), \;\; \forall (p, q) \in \mathcal{E}, (a, b) \in \mathcal{L} \times \mathcal{L} \tag{2.43}$$

where the dual variables are called balance variables $y_{pq}(\cdot)$ and height variables $h_p(\cdot)$. Variables $y_{pq}(a), y_{qp}(a)$ are conjugate, *i.e.* $y_{qp}(\cdot) \equiv -y_{pq}(\cdot)$. The height variables $h_p(\cdot)$ are given by $h_p(\cdot) \equiv \theta_p(\cdot) + \sum_{q:(p,q) \in \mathcal{E}} y_{pq}(\cdot)$, thus only the vector $\mathbf{y} = \{y_{pq}\}$ is needed for specifying a dual solution.

Basically, the primal-dual algorithm for MRF optimization is an iterative procedure, where the pairs of integral-primal, dual solution $(\mathbf{x}^k, \mathbf{y}^k)$ are iteratively updated until the elements of the last pair $(\mathbf{x}, \mathbf{y})$ satisfy the chosen *relaxed complementary slackness* conditions as follows:

$$h_p(x_p) = \min_{a \in \mathcal{L}} h_p(a), \;\; \forall p \in \mathcal{V} \tag{2.44}$$

$$y_{pq}(x_p) + y_{qp}(x_q) = \theta_{pq}(x_p, x_q), \;\; \forall (p, q) \in \mathcal{E} \tag{2.45}$$

$$y_{pq}(a) + y_{qp}(b) \leq 2\theta_{max}, \;\; \forall (p, q) \in \mathcal{E}, (a, b) \in \mathcal{L} \times \mathcal{L} \tag{2.46}$$

If the above conditions hold true, then the solution $\mathbf{x}$ defines an f-approximation to the optimal MRF energy, where $f = 2\frac{\theta_{max}}{\theta_{min}}$ [1]. Given the initialization of $(\mathbf{x}, \mathbf{y})$, the primal-dual pair of solutions is updated in each iteration of a label $c \in \mathcal{L}$. This update reduces to solving a max-flow problem for a certain capacitated graph $\mathcal{G}^c$ and its construction depends on the current primal-dual pair of solutions $(\mathbf{x}^k, \mathbf{y}^k)$. In particular, the directed graph $\mathcal{G}^c$ consists of the internal nodes and two external nodes (*i.e.* the source s and the sink t). The interior edges are responsible for updating the balance variables, while the exterior edges are in charge of updating the height variables.

[1] $\theta_{max} \equiv \max_{a \neq b} \theta_{pq}(a, b), \theta_{min} \equiv \min_{a \neq b} \theta_{pq}(a, b)$

In fact, the complexity of the primal-dual method largely depends on the complexity of the max-flow problem (determined by the exterior edges and their capacities), which in turn depends on the number of augmenting paths per max-flow. [Komodakis 2007b] proposes an efficient primal-dual method called *Fast-PD*. It makes use of the pair of primal and dual solutions from the previous iteration in order to reduce the number of augmenting paths for the next iteration. In particular, the primal-dual gap is an approximate upper bound on the number of such paths and they manage to reduce this gap throughout the iterations, thus resulting a significant speed up on the MRF optimization. In addition, Fast-PD can provide good optimality properties: if $\theta_{pq}(\cdot,\cdot)$ is a metric, Fast-PD is as powerful as α-expansion with the same solution but at least 3-9 times faster. If the pairwise function $\theta_{pq}(\cdot,\cdot)$ is a non-metric, it still guarantees an almost optimal solution.

Tree-reweighted Message Passing

[Wainwright 2005] introduces the *tree-reweighted max-product message passing* (TRW) algorithms which connect the message passing algorithm to the LP relaxation of the integer program. It is motivated by tying to optimize the LP relaxation problem by maximizing the dual of the LP. The basic idea is to use a convex combination of tree-structured distributions to obtain an upper bound on the optimal value of the original problem in terms of the combined optimal values of the tree problems. They prove that any such bound is tight if and only if the trees share a common configuration which must be the optimal for the original problem. Under a certain condition called *tree agreement*, fixed points of the tree-reweighted max-product updates correspond to dual-optimal solution of the tree-relaxed linear program, thus they guarantee to give a MAP solution. However, the TRW algorithms do not guarantee the convergence as well as the increase of the lower bound.

[Kolmogorov 2006] proposes a modification of the TRW algorithms called *sequential tree-reweighted message passing* (TRW-S). Similar to the TRW algorithms, it is formulated by maximizing a lower bound on the energy:

$$\max_{\boldsymbol{\theta}\in\mathcal{A},\sum_{T\in\mathcal{T}}\rho^T\theta^T\equiv\bar{\theta}} \Phi_\rho(\boldsymbol{\theta}) \tag{2.47}$$

where $\mathcal{T}$ is a collection of trees in the graph $\mathcal{G}$. For a tree $T\in\mathcal{T}$, ρ^T is some distribution of non-zero probability, while θ^T is the energy parameter. By concatenating all the tree vectors, a vector $\boldsymbol{\theta}$ is formed and it must belong to the constraint set $\mathcal{A}$. The known parameter $\bar{\theta}=\{\{\theta_p(\cdot)\},\{\theta_{pq}(\cdot,\cdot)\}\}$ represents the potentials of the original MRF. The concave function $\Phi_\rho(\boldsymbol{\theta})$ is a lower bound on the optimal value of the MRF if $\sum_{T\in\mathcal{T}}\rho^T\theta^T=\bar{\theta}$. The TRW-S algorithms consist of reparameterization and averaging operations as same as the TRW algorithms. The difference is that TRW-S algorithms update the messages θ^T

in a sequential order rather than a parallel order as in the TRW algorithms. Their main contribution is that the value of the bound is guaranteed not to decrease. They provide the *weak tree agreement* (WTA) condition which characterizes the local maximum of the bound with respect to TRW algorithms. They prove that the algorithm has a subsequence converging to a vector satisfying the WTA condition. If a vector satisfies the WTA condition, then it maximizes the lower bound (the lower bound will not increase). They also show that their algorithm requires half as much memory compared to traditional message passing approaches. However, WTA is not guaranteed to give the global maximum of the lower bound, and the convergence is still not guaranteed. Nevertheless, in some specific cases when the original MRF is binary and submodular, the fixed points of TRW-S correspond to the global maximum of the dual problem.

Dual Decomposition

[Komodakis 2007a, Komodakis 2011b] propose a new message-passing scheme for MRF optimization based on *Dual Decomposition* (DD) technique. Unlike the existing message-passing methods, it can provably solve the dual LP (*i.e.* maximize the lower bound) and it behaves better theoretical properties than the TRW methods.

The basic idea of this method is to decompose the original MRF optimization problem (NP-hard) which is defined on the graph $\mathcal{G}$ into a set of easier MRF subproblems where each of them is defined on a tree $T \subset \mathcal{G}$. Given a set of subtrees $\mathcal{T}$ covering all the nodes and edges of graph $\mathcal{G}$, each tree $T \in \mathcal{T}$ is associated with a vector of MRF parameters θ^T and a vector of MRF variables $\mathbf{x}^T$. The Lagrangian dual of the original MRF problem is defined as:

$$\max_{\{\boldsymbol{\lambda}^T\} \in \Lambda} g(\{\boldsymbol{\lambda}^T\}) = \sum_{T \in \mathcal{T}} g^T(\boldsymbol{\lambda}^T) \tag{2.48}$$

where the introduced Lagrange multipliers $\boldsymbol{\lambda}^T$ are constrained to the feasible set Λ, and each function $g^T(\cdot)$ is defined as:

$$\begin{aligned} & g^T(\boldsymbol{\lambda}^T) = \min_{\mathbf{x}^T} E(\theta^T + \boldsymbol{\lambda}^T, \mathbf{x}^T) \\ & s.t. \quad \mathbf{x}^T \in \mathcal{X}^T \end{aligned} \tag{2.49}$$

which is equivalent to the task of minimizing the MRF energy $E(\cdot, \cdot)$ over a tree T, *i.e.* a much easier problem. In this manner the dual problem is decomposed into solvable subproblems. The problem in Eq.(2.48) is called a *master* problem, and the subproblems in Eq.(2.49) are called *slave* problems. In order to optimize the master, they use the projected subgradient method where the solutions of the master and the slaves are combined in a principled way. In fact, the optimization can be considered as an iterative "message

passing" between the master problem and the slave problems. In each iteration, the slave problems are solved according to the current MRF potentials, then the solutions of the slaves (messages) are sent to the master problem in order to update the MRF potentials.

The resulting elegant MRF optimization framework carries great flexibility and generality. They show that by appropriately choosing the subproblems, it allows to design powerful MAP estimation algorithms. As a result, they can derive algorithms which (1) generalize and extend state-of-the-art message-passing methods while providing better theoretical properties, (2) optimize much tighter LP relaxation associated to an MRF problem, and (3) take advantage of the structure of any particular class of MRF, thus allowing to use fast inference techniques *i.e.* graph-cut based method.

Chapter 3

Statistical Shape Model

In this chapter, we introduce a pose invariant shape model which is represented as a high-order graph. The triplet cliques encode the local shape variation statistics while inheriting invariance to global transformations. The shape manifold is constructed through L_1 sparse higher-order graph structure which is learned through dual decomposition from a training set, while preserving the ability to describe shape variability as well as being compact.

3.1 Introduction

The shape of an object is a geometrical description of the object boundary and it plays an important role in computer vision tasks such as recognition, segmentation and tracking. In the context of image interpretation, *a priori* knowledge of the object of interest (color, texture, shape) is very useful in distinguishing the object from the background. In particular, incorporating the shape information of the object in the segmentation task shows significant advantages of improving accuracy and robustness of the algorithms, when the appearance features of the object in the image alone are not sufficient due to noise, background clutter or partial object occlusions. Therefore, it is very necessary and critical to build a shape model which represents the prior shape knowledge of the object class, and it should be easily incorporated in solving the vision tasks.

Shape prior modeling is a very challenging task due to shape variability of the object of interest. A simple way to represent shape prior is to use a shape template which is a typical shape example of the object of interest. However, it is not specific to describe the object class with considerable natural variability. For example, the shape of human organ shows both inter-individual variability (differences within populations) and intra-individual variability (differences of the same subject in tests taken at different time or in different conditions). In order to construct a reliable shape model with information about

the common variation of the object class, a straight direction is to gather these information by statistical means from a number of observations of the object (as many as possible), which leads to statistical shape models (SSMs).

3.1.1 Previous Work

Statistical shape modeling is a well-studied problem. Generally, it consists of two critical components:

- A mathematical definition of the shape representation for statistic analysis.
- Constructing a statistical model that describes the observed shape variations from a training set.

With respect to the shape representation, point-based or landmark-based representation [Cootes 1995, Cootes 2001], implicit representations [Cremers 2007], superquadric model [Metaxas 1993], medial model [Pizer 1999], Fourier surface [Staib 1996] are some examples. Based on the shape representation, the statistical model can be represented either by a mean shape and the principal modes of variations or by a probability density function. The most popular method used in the first case is principal component analysis (PCA) [Jolliffe 2002] which approximates the shape by a linear combination of largest modes of variations. In the second cases, Gaussian probability density function [Rousson 2002], Gaussian mixture models [Cootes 1999b], kernel density [Cremers 2006a] as well as manifold learning [Etyngier 2007] have been employed.

One of the most well-known methods in the area of statistical shape modeling is the Active Shape Model (ASM) [Cootes 1995]. This approach models the shape variations in a Point Distribution Model (PDM) which is learned in the following steps: (1) Given a training set, each shape sample is represented by a number of points with correspondences across the training set; (2) An alignment of the training shapes in a common reference frame using Procrustes Analysis [Dryden 1998] has to be performed in order to eliminate pose variations. (3) The mean shape of all aligned training samples can be computed and the variation modes with respect to the mean shape are computed by principal component analysis (PCA). (4) At last, the shape model is represented by a linear combination of the largest modes of variation. Using bounded coefficient parameters, new shape instances can be generated to remain into the allowable shape domain (ASD) in order to look like the ones in the training set. Successfully applied to various types of shapes (faces, hands, organs), PDM has became a standard in statistical shape modeling in particular in the context of medical image segmentation.

The PCA based shape modeling method can be applied to not only point-based representation, but also other shape descriptions. [Székely 1996] used Fourier parametrization

of the object surface [Staib 1996] and applied principal component analysis (PCA) in the Fourier coefficient space. Closely, [Kelemen 1999] expanded surface representation into series of spherical harmonics (SPHARMs) and calculated shape eigenmodes in the shape parameter space as well. [Davatzikos 2003] used wavelet transform of the object contour and built a hierarchical shape model via PCA on the coefficients in each band, and [Nain 2007, Yu 2007] used spherical wavelets to extend this approach to 3D cases.

Implicit shape representation given by the level set framework is another popular choice in shape representation, where the contour of the object is embedded as the zero level set of a higher dimensional surface. The embedding function is often chosen as signed distance maps. [Leventon 2000] represented each training shape as a signed distance map sampled at regular intervals and performed PCA on the signed distance maps to build the shape model. [Tsai 2001a] also applied PCA to a collection of signed distance representations of the training set and optimized the shape parameters directly in segmentation process. Based on this global statistical representation, [Rousson 2002] considered a more challenging shape model that accounts for local variation as well. A common criticism of performing PCA on signed distance maps is that it can lead to invalid shapes since distance maps do not form a linear space. [Pohl 2006] addressed this problem by embedding the signed distance map manifold into the linear LogOdds vector space and applying PCA on the latter space.

Given the richness of the literature in statistical shape modeling, we strongly recommend the recent review [Heimann 2009] of statistical shape models for 3D medical image segmentation. We also refer the reader to [Cremers 2007] for a review of statistical approaches to level set segmentation. In this thesis, we represent shape using point-based representation as same as Point Distribution Model (PDM), since it is straight to do the statistics as well as to infer the shape instance in an observed image.

Drawbacks of PDM

One drawback of the Point Distribution Model (PDM) is that it models the shape variations in a global way since each mode of variation influences all the variables of shape at the same time. This global effect limits shape model to have the flexibility of controlling local variations which is a desired property in shape analysis or diagnostic purposes. In order to obtain variation modes which only affect a limited number of local landmarks, the Orthomax method employed by [Stegmann 2006] rotates the PCA modes to increase sparsity while maintaining the orthogonality of components. Another solution is Sparse PCA [Sjostrand 2007] which obtains the sparse modes and produces near-orthogonal components. Independent component analysis (ICA) [Hyvärinen 2000] does not assume a Gaussian distribution and delivers statistically independent projections without orthogonality

criteria. These non-PCA methods however provide no natural ordering for the variation modes thus different techniques have to be employed.

Another practical issue in PDM is that it is problematic to represent the full range of shape variations in high dimensional space from a small training set. The size of the training set is always relatively small since the available images and their required manual segmentation are not easy to obtain especially in 3D cases, while the maximum number of eigenmodes can not exceed the number of the training examples minus one. [Cootes 1996] addressed this problem by introducing additional synthetic variance and covariance directly to the covariance matrix and coupling the movements of the neighboring points along the boundary. Another approach to solve this problem is to divide the shape model into smaller parts whose variations can be captured with less training samples. [Davatzikos 2003] represented a hierarchical shape model in terms of its wavelet transform, while the lower bands of the transform correspond to global shape characteristics and the higher bands to the local ones.

Moreover, PCA as a linear model, cannot adequately model non-linear shape variations such as bending and shape variations of an articulated object. [Cootes 1999b] estimates the probability density function of the distribution of shapes as a mixture of Gaussians to deal with this problem. Kernel Principal Component Analysis (KPCA) [Schölkopf 1998] introduces a non-linear mapping of the data to a feature shape and PCA is performed in the feature space, and it was applied by [Twining 2001] to the task of constructing non-linear ASMs. KPCA has become popular for implicit shape representation [Cremers 2003] since it solves the problem that signed distance maps do not form a linear vector space. Manifold learning is also used to model non-linear shape prior. For example, [Etyngier 2007] modeled a category of shapes as a shape prior manifold using Diffusion maps which generate a mapping from the original shape space into a low-dimensional space.

In addition, the shape models such as PDM are not pose-invariant since they are modeled in a common coordinate frame. In this context, an estimation of the global pose parameters (translation, rotation and scale) of the shape is required in both training and inference stages. Such a process introduces a strong statistical bias on the segmentation task, and at the same time it makes the applicability of the model problematic when referring to diseased subjects. Furthermore, since the shape inference which involves the global pose estimation is often operated in a local search, these methods require the initialization to be very close to the ground truth.

Related Work

[Cootes 1992] developed a shape model using Chord Length Distribution (CLD) representation. Given a training set, each shape example is represented by a vector which contains

the distances between all pairs of points, and then a PCA is applied to model the variations of the chord lengths. A new shape can be generated by varying the parameters which change the distances. This shape model is invariant to the object orientation since it relies on internal distances, and it can model bending objects while the linear PDM fails. However the inference of the shape is more complex from distances rather than from points.

Instead of building one single global shape model, [Seghers 2008] built the shape prior from a concatenation of local statistical shape models. This method is based on graph representations where a vertex corresponds to a landmarks and an edge defines the local interaction between a pair of neighboring landmarks using their mahalanobis distance. Such a shape model inherits invariant properties with respect to the translation and rotation changes of the global shape, but both training and testing images still have to be registered and resampled to the reference image to filter out the scaling effect. More importantly, considering only the constraints on neighboring landmarks does not guarantee of learning the underlying shape manifold.

[Besbes 2009] built the shape model as an incomplete graph that consists of intra and inter-cluster connections representing the inter-dependencies of control points, after clustering the control points according to their behavior within the training set. The interaction between a pair of control points is represented by the distribution of the normalized distance which is the Euclidean distance of the two points normalized by the scale of the object. This normalized distance representation is similarity-invariant and is available for the training set. However, it can not to be encoded in a pairwise term in the MRF inference since it requires the scale of the object which is determined by all the control points. Based on an interactive searching scheme, they approximate the global scale as the scale of the obtained shape at the previous iteration of the shape evolution. Thus, the scale-invariant property may not hold when the estimated scale is not close to the actual scale.

[Wang 2010] introduced a pose-invariant shape model where the prior manifold is constructed through the accumulation of local constraints on triplet cliques. Instead of using pair distance normalized by the global scale, each local interaction is represented by two normalized pair distances in the triplet clique, while the normalizing factor is the sum of the three pair distances of the clique. This representation is invariant with respect to translations, rotations and scales, and it can be exactly encoded in a higher-order MRF framework. Unfortunately, a fully connected graph which consists all possible triplet cliques leads to the inherited computational complexity and makes this method impractical.

3.1.2 Our Proposed Method

In this thesis, we propose a novel statistical shape model which can address the limitations of the existing approaches. We adopt a point distribution graphical model which encodes

pose invariant shape priors through L_1 sparse higher order cliques. The second-order potentials encode the local shape variation statistics, while the subset of cliques from all possible second-order cliques is learned through dual decomposition from a training set. This is to provide the best possible reconstruction of the observed shape variation, while being as compact as possible. We summarize the advantages of our model as follows:

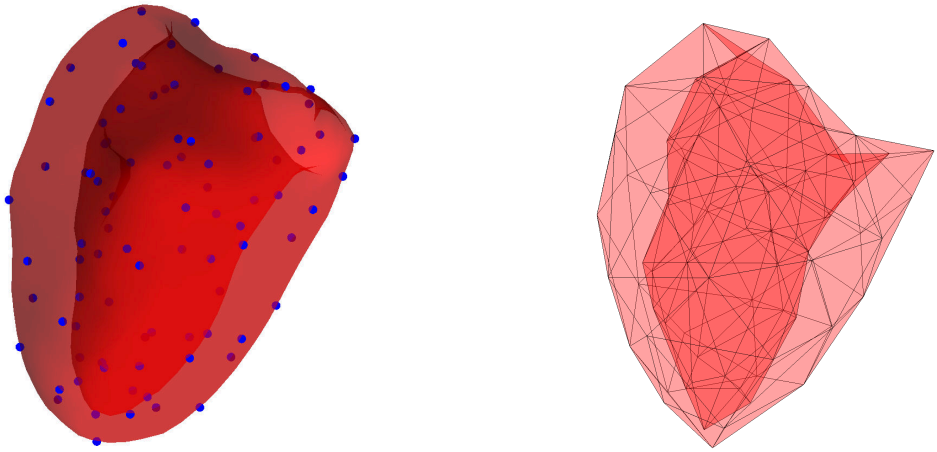
- We propose a model that is pose invariant, efficient to learn and perform inference on, and does not suffer from bad local minima issues at test time.
- Such a model, unlike most active shape ones, does not need to align neither training samples nor testing shape in a common coordinate frame.
- Due to imposing pose invariance through the use of local interactions (non-linear model), this allows the model to have the flexibility and thus generalization with respect to unseen shapes.
- Our statistical modeling on local interactions can capture shape variations from a small training set.
- Last but not least, a sparse graph structure achieved from Markov Random Field (MRF) learning eliminates the redundancy in the model, thus boosting efficiency while preserving its ability to represent the data.

The remaining of the chapter is organized as follows. In Section 3.2, we introduce a pose invariant shape model which is formulated in a Markov Random Field (MRF) where triplets of points encode the local shape variations statistics while inheriting invariance to global transformations. Section 3.3 presents a compact representation of the shape priors where the L_1 sparse higher-order graph is learned through dual decomposition from a training set, while preserving the ability to describe shape variability. Experimental results are presented in Section 3.4.

3.2 Pose Invariant Shape Model

3.2.1 Point-based Representation

The first fundamental decision in statistical shape model is the choice of shape representation. In this thesis, we use a point-based representation since it is straight-forward and easy to do statistics on points in order to learn their behavior. As Point Distribution Models (PDM), the shape of the object of interest is represented by a number of points which are distributed on the shape boundary. For example, Figure 3.1(a) shows a point-based



(a) (b)

Figure 3.1: Point-based model of 3D myocardium. (a) Distribution of the control points. (b) Triangle mesh.

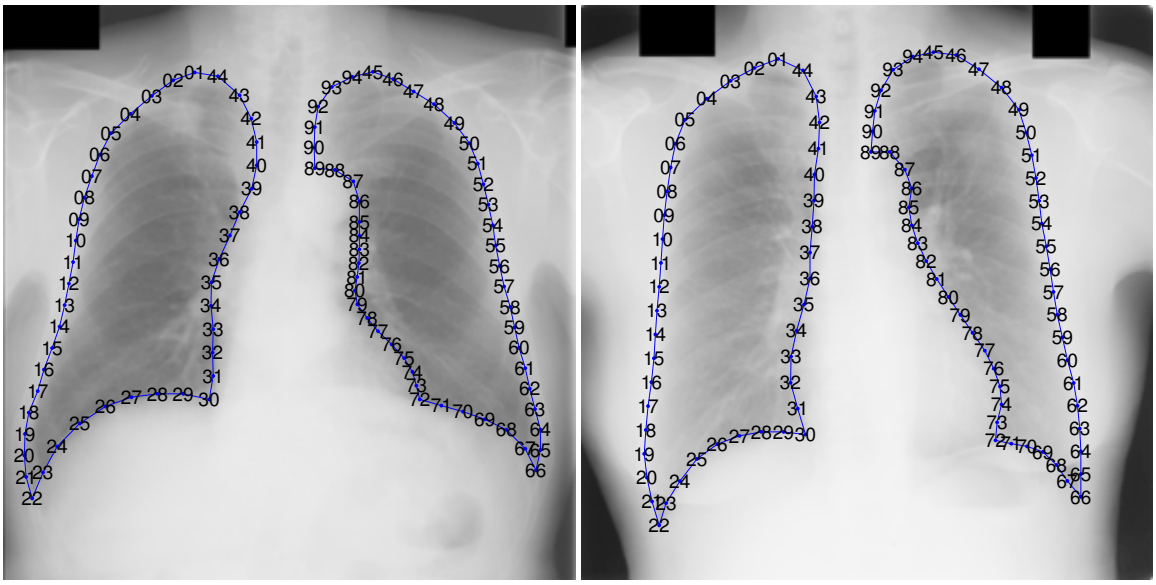
model of 3D myocardium where the involved control points are marked as blue dots. In this manner, a shape instance $\mathbf{X}$ can be described by a set of boundary points:

$$\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \tag{3.1}$$

where n denotes the number of the points, $\mathcal{V} = \{1, \cdots, n\}$ denotes the point set, and $\mathbf{x}_{i \in \mathcal{V}}$ denotes the coordinates of the i-th point. The involved points are often referred as *landmarks* in Statistical Shape Model (SSM) literature, although they are not obliged to be located at salient feature points as the common definition for anatomic landmarks. We also call them *control points* to avoid confusion. As soon as the positions of the control points are known, the shape boundary can be approximately reconstructed using connectivity information between neighboring points, forming a closed curve for 2D shapes (*e.g.* see Figure 3.2) or a triangle mesh for 3D shapes (*e.g.* see Figure 3.1(b), where the control points are modeled as vertices in the mesh).

Point Correspondences

Based on the shape representation, a training set of K shape instances $\{\mathbf{X}^1, \cdots, \mathbf{X}^K\}$ is necessary to build the statistical shape model, where each sample is represented with n control points, *i.e.* $\mathbf{X}^k = \{\mathbf{x}_1^k, \cdots, \mathbf{x}_n^k\}, \forall k \in \{1, \cdots, K\}$. A very important requirement of *point correspondences* has to be guaranteed on the training set, which requires that the n control points in each sample should be located in a consistent manner. In particular,



(a) (b)

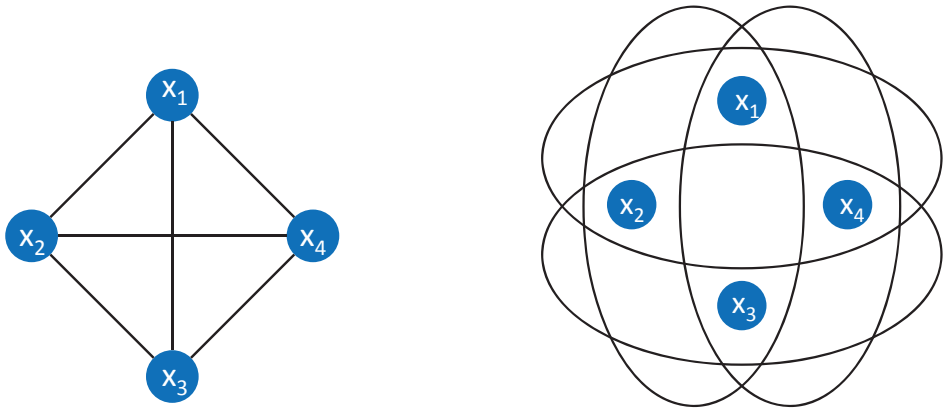
Figure 3.2: Landmark labeling on two samples of 2D lung with point correspondences.

$\forall i \in \mathcal{V}$, the i-th landmark in each sample $\{\mathbf{x}_i^1, \cdots, \mathbf{x}_i^K\}$ should correspond to the same location on the shape. For instance, in Figure 3.2 two shape samples (a) and (b) are used to show the point correspondences, where the control points with the same index in different samples are located at the same position of the shape. In practice, the correspondences can be obtained through manually labeling the control points for each training sample, or labeling a shape instance at first and then deducing the control points of the other instances in the training set through registration between the labeled one and non-labeled ones.

3.2.2 Statistical Shape Prior

Given a training set, we aim to learn the intrinsic geometric prior of a class of shape. Mathematician and statistician David George Kendall [Kendall 1984] defined "shape" informally as all the geometrical information that remains when location, scale and rotational effects are filtered out from an object. In this context, a crucial requirement of a statistical shape model is to be *pose invariant*, *i.e.* invariant to translation, rotation and scale changes of the object. Shape variations induced by these global transformations should be excluded from the modeling.

Most statistical shape models such as Active Shape Models (ASMs) involve an alignment process to filter out translation, rotation and scale change effects. All the training



(a) (b)

Figure 3.3: Local interactions of the shape. (a) Connections by pairs of control points. (b) Connections by triplets of control points.

samples (as well as the testing shapes) have to be aligned in a common coordinate frame. Such a process introduces strong statistical bias, and at the same time it makes the applicability of the model problematic when referring to diseased subjects.

In this section, we propose a statistical shape model which is pose-invariant, thus it does not need shape alignment. The global shape constraint is expressed as a combination of local interactions. In the following, two types of local interactions based on a subset of control points are going to be studied.

$$\begin{aligned} \mathcal{E} =& \{(i,j)|i,j \in \mathcal{V} \text{ and } i \neq j\} \\ \mathcal{C} =& \{(i,j,k)|i,j,k \in \mathcal{V} \text{ and } i \neq j \neq k\} \end{aligned} \tag{3.2}$$

where $\mathcal{E}$ denotes a set of all possible pairs of control points, and $\mathcal{C}$ denotes a set of all possible triplets of control points on the shape. We show the two types of interactions in Figure 3.3 with a simple model which contains 4 control points. The pairwise set $\mathcal{E}$ is represented by the edges linking every two points in (a); the triplet set $\mathcal{C}$ is represented by a group of ellipses, where each ellipse surrounds three control points.

Now we describe the shape probability $p(\mathbf{X})$ in terms of co-occurrence probabilities, according to the pioneering works of [Cremers 2006b] and [Seghers 2008]. Based on product rule, the joint probability of n control points can be written as:

$$\begin{aligned} p(\mathbf{x}_1, \cdots, \mathbf{x}_n) &= p(\mathbf{x}_1)\ p(\mathbf{x}_2, \cdots, \mathbf{x}_n|\mathbf{x}_1) \\ &= p(\mathbf{x}_1)\ p(\mathbf{x}_2|\mathbf{x}_1)\ p(\mathbf{x}_3, \cdots, \mathbf{x}_n|\mathbf{x}_1, \mathbf{x}_2) \\ &= p(\mathbf{x}_1)\ p(\mathbf{x}_2|\mathbf{x}_1)\ p(\mathbf{x}_3|\mathbf{x}_1, \mathbf{x}_2) \cdots\ p(\mathbf{x}_n|\mathbf{x}_1 \cdots, \mathbf{x}_{n-1}) \end{aligned} \tag{3.3}$$

Two assumptions are considered for shape prior. First, we assume $\forall i \in \mathcal{V}$ a constant $p(\mathbf{x}_i)$ since the shape prior should be invariant to translations. This assumption also gives the equivalence of the conditional probabilities for each pair $(i, j) \in \mathcal{E}$.

$$p(\mathbf{x}_i|\mathbf{x}_j) = p(\mathbf{x}_j|\mathbf{x}_i) \propto p(\mathbf{x}_i, \mathbf{x}_j) \tag{3.4}$$

Second, the control points are only dependent within a certain order of interaction, *e.g.* for pairwise interaction, the co-occurrence probability function of two control points does not depend on a third point.

$$p(\mathbf{x}_i, \mathbf{x}_j|\mathbf{x}_k) \approx p(\mathbf{x}_i, \mathbf{x}_j), \forall k \neq i, j \tag{3.5}$$

Under the above assumptions, Eq.(3.3) can be simplified by choosing one particular decomposition of the joint probability as follows.

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_n) \propto \prod_{i=1}^{n-1} p(\mathbf{x}_i, \mathbf{x}_{i+1}) \tag{3.6}$$

On the right side of the equation, a number of $n - 1$ pairs are considered to cover all the control points without generating loops, composing a structure named spanning tree. We can perform the same simplification to all possible spanning trees of n control points. Multiplying all these equations, we have:

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_n)^{\Gamma} \propto \prod_{(i,j)\in\mathcal{E}} p(\mathbf{x}_i, \mathbf{x}_j)^{\nu} \tag{3.7}$$

where Γ is the number of all possible spanning trees on the control point set, and ν denotes the number of times each pair appears in the overall product. Obviously, each pair in a complete graph has the same number of appearances. Since one spanning tree includes $n - 1$ pairwise factors, all the spanning trees have $(n - 1) \cdot \Gamma$ factors. Meanwhile on the right side, the number of all possible pairs is $\frac{n(n-1)}{2}$, the total pairs include $\frac{n(n-1)}{2} \cdot \nu$ factors, then we have the relation $\Gamma = \frac{n}{2}\nu$. Therefore, Eq.(3.7) can be simplified as:

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_n) \propto \prod_{(i,j)\in\mathcal{E}} p(\mathbf{x}_i, \mathbf{x}_j)^{\frac{2}{n}} \tag{3.8}$$

Similarly, assuming that the co-occurrence of any triplet of control points does not depend on a fourth point, the shape probability can be estimated by the product of probabilities of all possible triplets.

$$p(\mathbf{x}_1, \cdots, \mathbf{x}_n) \propto \prod_{(i,j,k)\in\mathcal{C}} p(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)^{\gamma} \tag{3.9}$$

where the constant γ denotes the number $n - 2$ of triplets included in one decomposition divided by the number $C(n, 3)$ of all possible triplets, thus we have $\gamma = \frac{6}{n(n-1)}$.

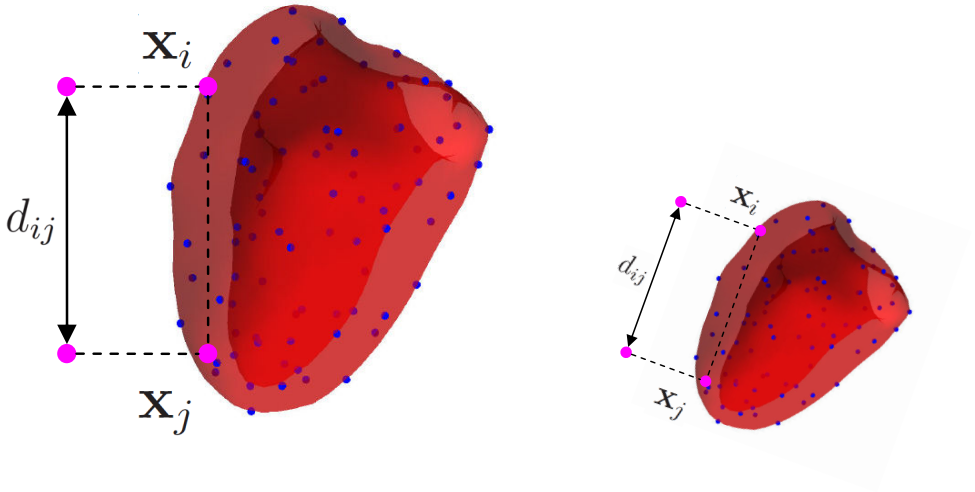


Figure 3.4: Pairwise interaction representation: chord length. Left: a shape instance. Right: a similarity transformation of the shape.

Probabilities of Pairs

Let us consider a pair $(i, j) \in \mathcal{E}$ of control points, the constraint between the two points $(\mathbf{x}_i, \mathbf{x}_j)$ can be described by their chord length, *i.e.* Euclidean distance d_{ij} between the two points (see Figure 3.4).

$$p(\mathbf{x}_i, \mathbf{x}_j) = p_{ij}(d_{ij}) = p_{ij}(\|\mathbf{x}_i - \mathbf{x}_j\|) \tag{3.10}$$

This chord length representation is invariant to shape changes with respect to position and orientation, but it is variant with the scale of the shape. Figure 3.4 shows a shape instance (left) and a similarity transformation (right), where we can see that the chord length changes due to the scaling effect. The global scale of a shape instance can be estimated by the average distance over all pair distances, $\bar{d} = \frac{2}{n \cdot (n-1)} \sum_{(i,j) \in \mathcal{E}} \|\mathbf{x}_i - \mathbf{x}_j\|$. In order to be invariant to the scale changes as well, the Euclidean distance between two points $(\mathbf{x}_i, \mathbf{x}_j)$ is divided by the global scale $\bar{d}$, thus producing a normalized distance $\hat{d}_{ij}$.

$$p(\mathbf{x}_i, \mathbf{x}_j) = p_{ij}(\hat{d}_{ij}) = p_{ij}(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\bar{d}}) \tag{3.11}$$

Given K shape samples, for each pair $(i, j) \in \mathcal{E}$, a set of normalized distance samples $\{\hat{d}_{ij}^1, \cdots, \hat{d}_{ij}^K\}$ can be calculated. Then the possibility density distribution $p_{ij}(\hat{d}_{ij})$ can be learned using a standard probabilistic model. However, although being scale invariant,

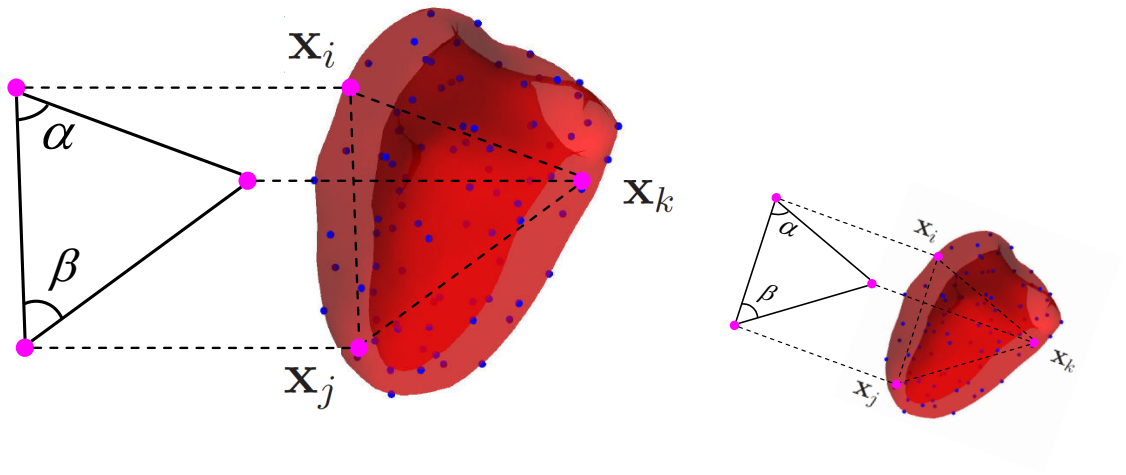


Figure 3.5: Pose invariant representation of triplet interaction: inner angles. Left: a shape instance. Right: a similarity transformation of the shape.

$p_{ij}(\hat{d}_{ij})$ is actually a posteriori probability. It is based on the condition that the global shape scale $\bar{d}$ is known, which is the case for training samples but is not the case for shape inference step unless a global scale estimation is performed in advance. Therefore, co-occurrence probabilities of pairs are neither pose invariant referring to the pair distance d_{ij} nor feasible in shape inference process referring to the normalized distance $\hat{d}_{ij}$.

Probabilities of Triplets

Now we consider higher-order cliques - triplets - to introduce a pose-invariant representation. For a triplet clique $c = (i, j, k) \in \mathcal{C}$, considering the control points $\mathbf{x}_c = (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ as end points, three line segments $\{\mathbf{x}_i\mathbf{x}_j, \mathbf{x}_i\mathbf{x}_k, \mathbf{x}_j\mathbf{x}_k\}$ are composed, producing three angles $\{\alpha_c = \angle\mathbf{x}_j\mathbf{x}_i\mathbf{x}_k, \beta_c = \angle\mathbf{x}_i\mathbf{x}_j\mathbf{x}_k, \theta_c = \angle\mathbf{x}_i\mathbf{x}_k\mathbf{x}_j\}$ between every two segments. The geometric shape of triplet c can be defined by its two inner angles (α_c, β_c) (see Figure 3.5).

$$p_c(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = p_c(\alpha_c, \beta_c) \tag{3.12}$$

where the inner angles (α_c, β_c) are given as follows, and the third angle θ_c is a linear combination of the two, *i.e.* $\theta_c = 180^\circ - \alpha_c - \beta_c$.

$$\alpha_c = \arccos \frac{\overrightarrow{\mathbf{x}_i\mathbf{x}_j} \cdot \overrightarrow{\mathbf{x}_i\mathbf{x}_k}}{\|\mathbf{x}_i\mathbf{x}_j\| \, \|\mathbf{x}_i\mathbf{x}_k\|}, \quad \beta_c = \arccos \frac{\overrightarrow{\mathbf{x}_j\mathbf{x}_k} \cdot \overrightarrow{\mathbf{x}_j\mathbf{x}_i}}{\|\mathbf{x}_j\mathbf{x}_k\| \, \|\mathbf{x}_j\mathbf{x}_i\|} \tag{3.13}$$

This representation of triplet of control points is invariant to global pose (*i.e.* translation, rotation, scale) of the shape of the object. As we can see in Figure 3.5, the angle information defined by a triplet is unchangeable under similarity transformation.

Given a training set of K shape samples, for each triplet $c \in \mathcal{C}$ has K instances $\{(\alpha_c^1, \beta_c^1), \cdots, (\alpha_c^K, \beta_c^K)\}$. Then, the probability density distributions $p_c(\alpha_c, \beta_c)$ of triplet c are learned using a standard probabilistic model, such as a Gaussian distribution $\mathcal{N}$.

$$p_c(\alpha_c, \beta_c) = \mathcal{N}(\alpha_c, \beta_c | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \tag{3.14}$$

where $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ are the mean and the variance matrix learned from the training set. Without loss of generality, both simple and complex distribution models can be used to learn the statistics.

3.2.3 Shape Inference

To this end, our pose invariant shape model $\mathbf{M} = (\mathcal{V}, \mathcal{C}, \mathcal{P})$ is built. It consists of a set $\mathcal{V}$ of control points, a set of triplets $\mathcal{C}$ and the statistical shape prior $\mathcal{P} = \{p_c(\alpha_c, \beta_c) | c \in \mathcal{C}\}$ which is constructed through the accumulation of triplet constraints based on Eq.(3.12). Based on this representation, the shape model can be easily encoded in an MRF model towards efficient optimization.

Now we perform the shape inference problem as a higher-order Markov Random Fields (MRF) formulation. Let $G = (\mathcal{V}, \mathcal{C})$ denote a hypergraph with a node set $\mathcal{V}$ and a clique set $\mathcal{C}$. We associate a control point to a node $i \in \mathcal{V}$ and a triplet of control points to a second-order clique $c \in \mathcal{C}$. Let X_i denote the latent variable (*i.e.* the coordinates of control point) of node i, and $\mathcal{U}_i$ denotes the candidate space for the variable X_i. The shape inference problem is transformed into estimating an optimal configuration $\mathbf{X} = (\mathbf{x}_i)_{i \in \mathcal{V}}$ of all the nodes over shape candidate space $\mathcal{U} = \prod_{i \in \mathcal{V}} \mathcal{U}_i$.

$$\mathbf{X}^{\text{opt}} = \arg\min_{\mathbf{X} \in \mathcal{U}} E(\mathbf{X}) \tag{3.15}$$

where the MRF energy $E(\mathbf{X})$ is formulated as the negative log of the shape probability $p(\mathbf{X})$ defined in Eq.(3.9). Thus the energy $E(\mathbf{X})$ can be calculated as the sum of higher-order terms defined on the triplet clique set $\mathcal{C}$.

$$E(\mathbf{X}) = -\log p(\mathbf{X}) = \sum_{c \in \mathcal{C}} h_c(\mathbf{x}_c) \tag{3.16}$$

where for each $c \in \mathcal{C}$ the second-order potential $h_c(\mathbf{x}_c)$ is defined by the co-occurrence probability of three control points which can be represented by the inner angles according to Eq.(3.12). The probability distribution $p_c(\alpha_c, \beta_c)$ is learned from the training set.

$$h_c(\mathbf{x}_c) = -\log p_c(\mathbf{x}_c) = -\log p_c(\alpha_c, \beta_c) \tag{3.17}$$

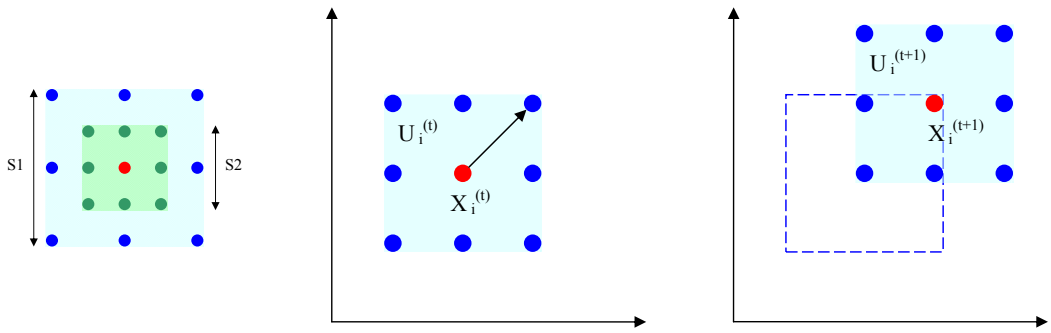


Figure 3.6: Model search space. Left: Two local candidate spaces of a node using two scales. Middle: the candidate space of a node at iteration t. Right: the candidate space of a node at iteration $t+1$.

To overcome the exhausting size of the search space, we adopt a coarse-to-fine iterative search strategy to recover the optimal shape configuration. For each node variable X_i, its candidate space $\mathcal{U}_i$ is iteratively determined by three factors: a sampling system $\mathcal{N}$ (*e.g.* a grid pattern), a neighborhood scale S and the current configuration $\mathbf{x}_i$. For example, on the left of Figure 3.6, a node variable X_i is represented by a red dot, two neighborhoods by scales S_1, S_2 around its current position $\mathbf{x}_i$ are draw in blue and green, the candidate space $\mathcal{U}_i$ is composed by the sample positions (blue dots or green dots) in the neighborhood, including its current position (red dot). In other words, the candidate space $\mathcal{U}_i$ can be considered as a transformation of a sampling system $\mathcal{N}$ by a translation to $\mathbf{x}_i$ and by a scaling S.

Given an initialization of the shape $\mathbf{X}_0$, the candidate space $\mathcal{U}$ is determined according to a sampling pattern $\mathcal{N}$ and an initial scale S_0 of the neighborhood. By searching over the candidate space, an optimal solution $\mathbf{X}^{\text{opt}}$ of the MRF formulation in Eq.(3.15) is obtained. The optimal shape will be taken as the initialization for the next iteration, also resulting a new candidate space. We start from a large search scale S_0 at the beginning so that the model can evolve to an approximate optimal shape by big displacements. When no better solution than the current configuration can be recovered in the search space, then we reduce the scale S of the candidate space by a rate r in order to refine the shape by smaller displacements. The iterative process ends until a threshold scale S_{min}. In particular for each node, an optimal position is recovered among its candidates at t-th iteration (see Figure 3.6 middle, the red dot represents the current position, while the arrow points to the optimal solution which is among the candidates marked by blue dots). Then the control point is moved to the optimal position, along with a local candidate space for the next

Algorithm 3.1 A coarse-to-fine local search strategy.

Input:

An initial shape $\mathbf{X} = \mathbf{X}_0$; A sampling system $\mathcal{N}$; An initial scale $S = S_0$; A scale threshold s_{min}; A scale rate r.

Iteration:

while $S > s_{\text{min}}$ **do**
  $\forall i \in \mathcal{V}, \mathcal{U}_i = \{S * \mathcal{N} + \mathbf{x}_i\}$;
  Optimize MRF to get $\mathbf{X}^{\text{opt}}$;
  if $\mathbf{X}^{\text{opt}} = \mathbf{X}$ **then**
    $S = S * r$;
  end if
  $\mathbf{X} = \mathbf{X}^{\text{opt}}$;
end while

optimization (see Figure 3.6 right). This iterative process allows evolving the shape in a coarse-to-fine manner with respect to the dynamic candidate space, thus producing a valid shape instance at the end. We show the search framework in Algorithm 3.1.

Based on such a MRF representation, we can apply a dual-decomposition optimization framework [Komodakis 2007a, Komodakis 2011b] to perform the inference of the proposed higher-order MRF, which does not suffer from bad local minima issues at test time. So far, we propose a shape model that is pose invariant, efficient to perform inference on. Such a model, unlike most active shape ones, does not need to align either training samples or testing shape in a common coordinate frame, meanwhile no global scale parameter estimation in the inference step is needed. Our shape model encodes global consistency as well as local variations. It can capture shape variations even with a small number of training examples.

However, the main limitation of this shape model is the inherited computational complexity that is proportional to the excessive number of higher order cliques. Thus we need to investigate a compact manner to encode prior knowledge that requires the smallest possible number of triplet cliques without altering the ability to express the shape manifold. Furthermore, assuming independence between all cliques may not always hold since there can be strong correlation between them at least at local scale. Last but not least, the significance of the different triplets towards capturing the observed deformations of the training set is not the same.

3.3 L_1 Sparse Graphic Model

In this section, we work on a compact representation of shape prior model while preserving the ability to describe shape variability. We parameterize[1] the MRF energy (3.16) by introducing an additional vector of parameters $\mathbf{w} = \{w_c | c \in \mathcal{C}\}$ containing one component w_c per clique c.

$$E(\mathbf{X}; \mathbf{w}) = \sum_{c \in \mathcal{C}} w_c h_c(\mathbf{x}_c) \tag{3.18}$$

where $h_c(\mathbf{x}_c)$ is defined in Eq.(3.17). Based on the above formulation, a clique c is associated with a contribution weight w_c such that clique c can be ignored if the corresponding element is zero, *i.e.* , if it holds $w_c = 0$. Therefore, the role of the introduced vector $\mathbf{w}$ is essentially to select which of the cliques are going to be retained in our shape prior model.

3.3.1 Max-Margin Learning

To estimate this vector $\mathbf{w}$, we use an MRF training procedure [Komodakis 2011a] during which we impose a sparsity-enforcing prior on vector $\mathbf{w}$ in order to eliminate as many redundant cliques as possible. Let $\{\mathbf{X}^k\}_{k=1}^K$ be the training set of shape instances. A max-margin learning formulation [Taskar 2004] is employed for computing $\mathbf{w}$, in which case we minimize the following regularized empirical loss:

$$\min_{\mathbf{w}} \lambda ||\mathbf{w}||_1 + \sum_{k=1}^{K} \xi_k \tag{3.19}$$

subject to the constraints:

$$E(\mathbf{X}^k; \mathbf{w}) \leq E(\mathbf{X}; \mathbf{w}) - \Delta(\mathbf{X}, \mathbf{X}^k) + \xi_k, \quad k \in \{1, \cdots, K\} \tag{3.20}$$

In the above expression (3.19), the term $\lambda ||\mathbf{w}||_1$ is a sparsity inducing L_1-norm regularizer, and a slack variable ξ_k denotes the loss with respect to a training example $\mathbf{X}^k$ for the defined MRF. Ideally the slack variable should be equal to zero, however in general, it will hold $\xi_k > 0$ and we must adjust the parameter vector $\mathbf{w}$ in order that the sum of the slack variables in the training set takes a minimal value.

The constraint (3.20) expresses the fact that we seek a parameter vector $\mathbf{w}$ such that the MRF energy of a ground truth shape $E(\mathbf{X}^k; \mathbf{w})$ should be smaller than the MRF energy of any other shape $E(\mathbf{X}; \mathbf{w})$ by at least a margin $\Delta(\mathbf{X}, \mathbf{X}^k)$, where $\Delta(\mathbf{X}, \mathbf{X}')$ represents a

[1] With a slight abuse of notation, symbols $E(\mathbf{X})$ and $E(\mathbf{X}; \mathbf{w})$ will hereafter be used interchangeably for denoting the energy of an MRF.

dissimilarity measure between two solutions $\mathbf{X}$ and $\mathbf{X}'$. This constrained learning problem is equivalent to an unconstrained one, when the loss function ξ_k is given as hinge loss:

$$\xi_k(\mathbf{X}^k; \mathbf{w}) = E(\mathbf{X}^k; \mathbf{w}) - \min_{\mathbf{X}} \Big(E(\mathbf{X}; \mathbf{w}) - \Delta(\mathbf{X}, \mathbf{X}^k) \Big) \tag{3.21}$$

There are two main challenges that we need to deal with in this case: (i) The MRF that we wish to train contains high-order terms, (ii) The learning must take into account the fact that, if $\mathbf{X}^k$ is a ground truth shape, then any transformed shape instance $T(\mathbf{X}^k)$ under a similarity transformation T is an equally good solution and should not be penalized during training, *i.e.* it should hold $\Delta(\mathbf{X}, T(\mathbf{X})) = 0$.

To deal with (ii), we choose the dissimilarity function $\Delta(\mathbf{X}, \mathbf{X}')$ that decomposes into the following higher-order terms:

$$\Delta(\mathbf{X}, \mathbf{X}') = \sum_{c \in \mathcal{C}} \delta_c(\mathbf{x}_c, \mathbf{x}'_c) \tag{3.22}$$

where the term $\delta_c(\mathbf{x}_c, \mathbf{x}'_c)$ equals 0 if two triplets of points $\mathbf{x}_c$ and $\mathbf{x}'_c$ are *similar*, and equals 1 otherwise. The similarity property of triplets can be defined using the angle representation (3.13) which is invariant to similarity transformation, *i.e.* if $\alpha_c = \alpha'_c$ and $\beta_c = \beta'_c$, then $\mathbf{x}_c$ and $\mathbf{x}'_c$ are similar.

We define a new MRF energy E^k associated with k-th sample.

$$E^k(\mathbf{X}; \mathbf{w}) = \sum_{c \in \mathcal{C}} g_c^k(\mathbf{x}_c; w_c), \quad g_c^k(\mathbf{x}_c; w_c) = w_c h_c(\mathbf{x}_c) - \delta_c(\mathbf{x}_c, \mathbf{x}_c^k) \tag{3.23}$$

Then, the slack variable ξ_k can be reformulated as follows:

$$\xi_k(\mathbf{X}^k; \mathbf{w}) = E^k(\mathbf{X}^k; \mathbf{w}) - \min_{\mathbf{X}} E^k(\mathbf{X}; \mathbf{w}) \tag{3.24}$$

which indicates the fact that the slack variable ξ_k equals to zero only if the minimum MRF energy E^k is obtained when the optimal solution is the ground truth shape $\mathbf{X}^k$ or its transformed shape $T(\mathbf{X}^k)$ under a similarity transformation T. Now we have to deal with the minimization problem of the regularized empirical loss (3.19). Using the definition of the loss function ξ_k (3.24), our objective function thus becomes equal to:

$$\min_{\mathbf{w}} \lambda ||\mathbf{w}||_1 + \sum_{k=1}^{K} \Big(E^k(\mathbf{X}^k; \mathbf{w}) - \min_{\mathbf{X}} E^k(\mathbf{X}; \mathbf{w}) \Big) \tag{3.25}$$

3.3.2 MRF Learning via Dual Decomposition

However, to minimize the above function is generally intractable due to the appearance of the minimum of the energy function $E^k(\mathbf{X};\mathbf{w})$ which contains higher-order potentials in our case. To deal with this challenge, we make use of the recently proposed dual decomposition framework for MRF optimization [Komodakis 2007a, Komodakis 2011b] which provides a general and flexible method for deriving and solving tight dual relaxations.

In dual decomposition framework, the original graph $G = \{\mathcal{V}, \mathcal{C}\}$ is decomposed into a set of sub-graphs $G_i = \{\mathcal{V}_i, \mathcal{C}_i\}$ such that their union covers all the nodes and cliques in the original graph, *i.e.* $\mathcal{V} = \cup\mathcal{V}_i$ and $\mathcal{C} = \cup\mathcal{C}_i$. The original minimization problem on graph G (called the master) is then decomposed into a set of easier sub-problems on sub-graph G_i (called the slaves), as shown in Figure 3.7. In general, the original MRF energy on graph G can be defined with unary potentials ϕ and higher-order potentials φ, while each slave MRFs on graph G_i are defined with their own unary potentials θ_i and the inherited higher-order potentials φ from the master MRF.

$$\begin{aligned} \text{Master: } \text{MRF}_G(\phi,\varphi) &= \min_{\mathbf{X}} \sum_{p\in\mathcal{V}} \phi(\mathbf{x}_p) + \sum_{c\in\mathcal{C}} \varphi(\mathbf{x}_c) \\ \text{Slave: } \text{MRF}_{G_i}(\theta_i,\varphi) &= \min_{\{\mathbf{x}_{p\in\mathcal{V}_i}\}} \sum_{p\in\mathcal{V}_i} \theta_i(\mathbf{x}_p) + \sum_{c\in\mathcal{C}_i} \varphi(\mathbf{x}_c) \end{aligned} \tag{3.26}$$

where the slave unary potentials should satisfy:

$$\sum_{i\in I_p} \theta_i(\mathbf{x}_p) = \phi(\mathbf{x}_p), \quad p \in \mathcal{V} \tag{3.27}$$

We denote I_p as the set of indices of those sub-graphs which contain the node p, *i.e.* $I_p = \{i | p \in \mathcal{V}_i\}$. This above equation makes sure that for each node p in the graph G, the sum of the unary potentials of the slaves which include the node p is equal to its unary potential of the master MRF. Because the sum of the minimum energies of the slaves always provides a lower bound to the minimum energy of the master MRF, *i.e.* $\sum_i \text{MRF}_{G_i}(\theta_i,\varphi) \leq \text{MRF}_G(\phi,\varphi)$, the dual relaxation of the master problem can be achieved through maximizing the sum of the minimum energies of the slaves by adjusting the slave unary variables $\boldsymbol{\theta}_i = \{\theta_i(\mathbf{x}_p) | p \in \mathcal{V}_i\}$.

$$\text{MRF}_G(\phi,\varphi) \approx \text{Dual}_{\{G_i\}}(\phi,\varphi) = \max_{\{\boldsymbol{\theta}_i\}} \sum_i \text{MRF}_{G_i}(\theta_i,\varphi) \tag{3.28}$$

where $\{\boldsymbol{\theta}_i\}$ are the dual variables and they should satisfy the constraints (3.27).

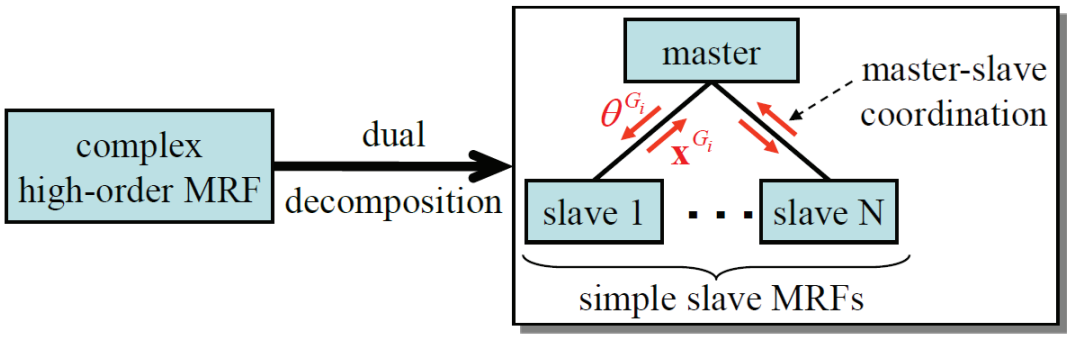


Figure 3.7: MRF optimization via dual decomposition [Komodakis 2007a].

Now we replace the minimum of the MRF energy in Eq.(3.25) with the dual relaxation:

$$\min_{\mathbf{X}} E^k(\mathbf{X};\mathbf{w}) \approx \text{Dual}_{\{G_i\}}(\phi,\varphi) = \max_{\{\boldsymbol{\theta}_i^k\}} \sum_i \text{MRF}_{G_i}^k(\theta_i^k,\varphi^k) \tag{3.29}$$

where we denote the slave problem on sub-graph G_i as $\text{MRF}_{G_i}^k$ with respect to the k-th sample. According to Eq.(3.23), the master energy in our case consists only higher-order potentials (*i.e.* $\varphi^k = g^k$) and no unary potentials (*i.e.* $\phi = 0$), and thus the slave unary potentials should satisfy $\sum_{i\in I_p^k} \theta_i^k(\mathbf{x}_p) = 0$ according to Eq.(3.27). The slave problem on sub-graph G_i can be written as:

$$\text{MRF}_{G_i}^k(\theta_i^k, g^k) = \min_{\mathbf{X}_{G_i}} E_{G_i}^k(\mathbf{X}_{G_i};\mathbf{w}_{G_i}) \tag{3.30}$$

where the slave MRF energy $E_{G_i}^k$ is defined as:

$$E_{G_i}^k(\mathbf{X}_{G_i};\mathbf{w}_{G_i}) = \sum_{p\in\mathcal{V}_i} \theta_i^k(\mathbf{x}_p) + \sum_{c\in\mathcal{C}_i} g_c^k(\mathbf{x}_c) \tag{3.31}$$

The master MRF energy of k-th training sample $E^k(\mathbf{X^k};\mathbf{w})$ is equal to the sum of the slave MRF energies due to the constraints (3.27).

$$E^k(\mathbf{X}^k;\mathbf{w}) = \sum_i E_{G_i}^k(\mathbf{X}_{G_i}^k;\mathbf{w}_{G_i}) \tag{3.32}$$

As a result, the loss function ξ_k (3.24) with respect to the k-th sample can be reformu-

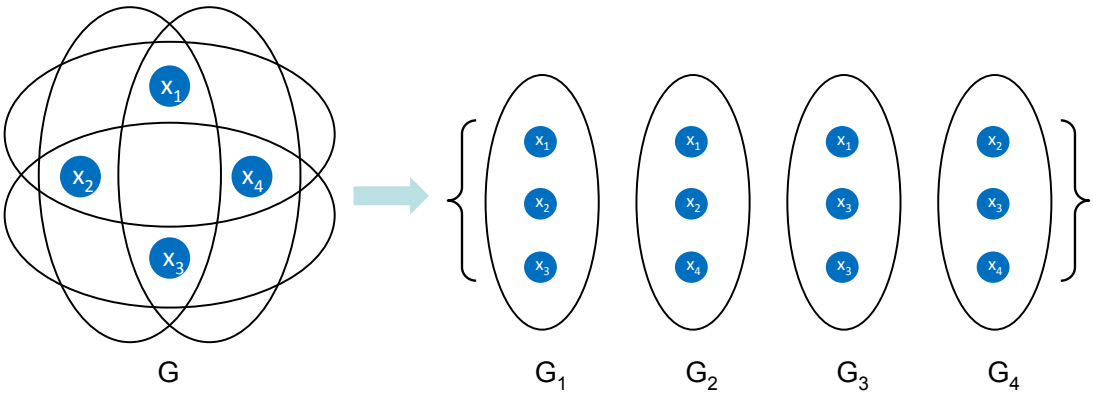


Figure 3.8: Decomposition of the graph.

lated using the slave terms:

$$\begin{aligned}\xi_k &\approx \sum_i E^k_{G_i}(\mathbf{X}^k_{G_i};\mathbf{w}_{G_i}) - \max_{\{\boldsymbol{\theta}^k_i\}} \sum_i \min_{\mathbf{X}_{G_i}} E^k_{G_i}(\mathbf{X}_{G_i};\mathbf{w}_{G_i}) \\ &= \min_{\{\boldsymbol{\theta}^k_i\}} \sum_i \left(E^k_{G_i}(\mathbf{X}^k_{G_i};\mathbf{w}_{G_i}) - \min_{\mathbf{X}_{G_i}} E^k_{G_i}(\mathbf{X}_{G_i};\mathbf{w}_{G_i}) \right) \\ &= \min_{\{\boldsymbol{\theta}^k_i\}} \sum_i L^k_{G_i}(\mathbf{X}^k_{G_i}, \boldsymbol{\theta}^k_i;\mathbf{w}_{G_i})\end{aligned} \tag{3.33}$$

where $L^k_{G_i}$ denotes the loss function of k-th sample with respect to the sub-graph G_i. To this end, the objective function (3.25) which we want to minimize is equal to:

$$\min_{\mathbf{w},\{\boldsymbol{\theta}^k_i\}} \lambda||\mathbf{w}||_1 + \sum_k \sum_i L^k_{G_i}(\mathbf{X}^k_{G_i}, \boldsymbol{\theta}^k_i;\mathbf{w}_{G_i}) \tag{3.34}$$

As can be seen, such a framework essentially manages to reduce the task of training a complex high-order model on the graph G, *i.e.* minimizing the regularized empirical loss (3.25) to the much easier task of training in parallel a series of slave MRFs defined on sub-graphs of G. The only restriction that must be obeyed by these sub-graphs is that (i) their union should cover the original graph G, and (ii) one should be able to minimize the energy of the so-called loss-augmented slave MRFs defined on these subgraphs.

In our case, we choose one sub-graph corresponding to each clique $c = \{o, p, q\} \in \mathcal{C}$ of graph G. Now the slave energy on the sub-graph (*i.e.* clique c) could be given as:

$$E^k_c(\mathbf{x}_c, \boldsymbol{\theta}^k_c; w_c) = \theta^k_c(\mathbf{x}_o) + \theta^k_c(\mathbf{x}_p) + \theta^k_c(\mathbf{x}_q) + w_c h_c(\mathbf{x}_c) - \delta(\mathbf{x}_c, \mathbf{x}^k_c) \tag{3.35}$$

Then the minimum of the slave energies $\min_{\mathbf{x}_c} E_c^k$ is easier to solve since there are only three variables to optimize. We redefine the loss function of k-th sample with respect to a clique c.

$$L_c^k(\mathbf{x}_c^k, \boldsymbol{\theta}_c^k; w_c) = E_c^k(\mathbf{x}_c^k, \boldsymbol{\theta}_c^k; w_c) - \min_{\mathbf{x}_c} E_c^k(\mathbf{x}_c, \boldsymbol{\theta}_c^k; w_c) \tag{3.36}$$

Therefore, the objective function in our case becomes:

$$\begin{aligned} \min_{\mathbf{w}, \{\boldsymbol{\theta}_c^k\}} \lambda||\mathbf{w}||_1 + \sum_k \sum_c L_c^k(\mathbf{x}_c^k, \boldsymbol{\theta}_c^k; w_c) \\ \text{s.t.} \quad \sum_{c \in C_p} \theta_c^k(\mathbf{x}_p) = 0 \end{aligned} \tag{3.37}$$

where $c \in \mathcal{C}$ denotes a triplet clique on the graph G, and the set $\mathcal{C}_p$ denotes the triplet cliques which contain the node $p \in \mathcal{V}$.

3.3.3 Projected Subgradient Algorithm

Now we adopt a projected subgradient algorithm to minimize the above objective function (3.37). The subgradient method is a simple algorithm for minimizing a non-differentiable convex function. Suppose $f : \mathbf{R}^n \to \mathbf{R}$ is convex, to minimize f, the subgradient method use iteration:

$$x^{(t+1)} = x^{(t)} - \alpha_t \cdot g^{(t)} \tag{3.38}$$

where $x^{(t)}$ is the t-th iteration, $g^{(t)}$ is any subgradient of f at $x^{(t)}$, and $\alpha_t > 0$ is the t-th step size. One extension of the subgradient method is the projected subgradient method, which solves the constrained convex optimization problem:

$$\text{minimize} \;\; f(x) \;\; \text{subject to} \;\; x \in \Omega \tag{3.39}$$

where Ω is a convex set. The projected subgradient method is given by

$$x^{(t+1)} = P(x^{(t)} - \alpha_t \cdot g^{(t)}) \tag{3.40}$$

where P is Euclidean projection on Ω, and $g^{(t)}$ is any subgradient of f at $x^{(t)}$.

In our case to minimize the objective function (3.37), the variables $\mathbf{w}$, $\{\boldsymbol{\theta}_c^k\}$ should be updated at each iteration as follows.

$$\begin{aligned} \mathbf{w}^{(t+1)} &= \mathbf{w}^{(t)} - \alpha_t \cdot d\mathbf{w}^{(t)} \\ \boldsymbol{\theta}_c^{k,(t+1)} &= P(\boldsymbol{\theta}_c^{k,(t)} - \alpha_t \cdot d\boldsymbol{\theta}_c^{k,(t)}) \end{aligned} \tag{3.41}$$

where $d\mathbf{w}$,$\{d\boldsymbol{\theta}_c^k\}$ denote the components of a subgradient of the objective function, $P(\cdot)$ denotes the projection onto the set $\Omega = \{\{\boldsymbol{\theta}_c^k\} | \sum_{c \in C_p} \theta_c^k(\mathbf{x}_p) = 0, \forall p \in \mathcal{V}\}$. The step size α_t is taken the non-summable diminishing step lengths.

$$\alpha_t = \frac{\gamma_t}{\|\{d\mathbf{w}, d\boldsymbol{\theta}_c^k\}\|} \tag{3.42}$$

where $\gamma_t > 0$ is a positive multiplier used for t-th iteration and it satisfies $\lim_{t\to\infty} \gamma_t = 0$ and $\sum_{t=0}^{\infty} \gamma_t = \infty$.

Our objective function is non-differentiable due to the term $\min_{\mathbf{x}_c} E_c^k(\mathbf{x}_c, \boldsymbol{\theta}_c^k; w_c)$ included in the definition of loss L_c^k (3.36). In order to compute a subgradient of the objective function, first we have to compute a subgradient for $\min_{\mathbf{x}_c} E_c^k(\mathbf{x}_c, \boldsymbol{\theta}_c^k; w_c)$. A subgradient of this term is given by the vector $\nabla E_c^k(\widehat{\mathbf{x}}_c^k, \boldsymbol{\theta}_c^k; w_c)$, where $\widehat{\mathbf{x}}_c^k$ is a minimizer of the slave MRF on a clique c. The vector has the following components:

$$\begin{aligned} dw_c &= \frac{\partial E_c^k(\widehat{\mathbf{x}}_c^k, \boldsymbol{\theta}_c^k; w_c)}{\partial w_c} = h_c(\widehat{\mathbf{x}}_c^k) \\ d\theta_c^k(\mathbf{x}_p) &= \left[\widehat{\mathbf{x}}_p^k = \mathbf{x}_p\right] \end{aligned} \tag{3.43}$$

where $[\cdot]$ equals 1 if the expression in square brackets is satisfied, and 0 otherwise.Then, the components $d\mathbf{w}$ and $\{d\boldsymbol{\theta}_c^k\}$ of the total subgradient of the objective function are given by:

$$\begin{aligned} d\mathbf{w} &= \lambda \nabla(\|\mathbf{w}\|_1) + \sum_{k,c} \left(h_c(\mathbf{x}_c^k) - h_c(\widehat{\mathbf{x}}_c^k)\right) \\ d\theta_c^k(\mathbf{x}_p) &= \left[\mathbf{x}_p^k = \mathbf{x}_p\right] - \left[\widehat{\mathbf{x}}_p^k = \mathbf{x}_p\right] \end{aligned} \tag{3.44}$$

When updating the slave unary potentials $\{\boldsymbol{\theta}_c^k\}$, we have to make sure that the resulting variables are projected on to the feasible set Ω. This projection is equivalent to subtracting the average vector $\sum_{c \in \mathcal{C}_p} \theta_c^k(\mathbf{x}_p) / |\mathcal{C}_p|$ from each vector $\theta_c^k(\mathbf{x}_p)$ such that the sum remains equal to zero as required by the constraints, where $|\mathcal{C}_p|$ denotes the number of cliques which contains the node p. Based on the above, the variables θ_c^k are updated at each iteration as follows:

$$\theta_c^k(\mathbf{x}_p)^{(t+1)} = \theta_c^k(\mathbf{x}_p)^{(t)} - \alpha_t \cdot \left(\left[\mathbf{x}_p^k = \mathbf{x}_p\right] - \frac{\sum_{c \in \mathcal{C}_p} \left[\widehat{\mathbf{x}}_p^k = \mathbf{x}_p\right]}{|\mathcal{C}_p|} \right) \tag{3.45}$$

In this manner, given a training set $\{X^k\}_{k=1}^K$, we can update the parameter vector $\mathbf{w}$ and slave unary potentials $\theta_c^k(\mathbf{x}_p)$ at each iteration until the objective function converges. The projected subgradient learning algorithm is shown in Algorithm 3.2. In addition,

Algorithm 3.2 Projected subgradient learning algorithm.

Input:

A training set of K shape samples $\{X^k\}_{k=1}^K$; a graph $G = (\mathcal{V}, \mathcal{C})$; high-order feature functions $\{h_c(\cdot)\}$.

$\forall k, c$, initialize $\boldsymbol{\theta}_c^k = \mathbf{0}$.

Iteration:

repeat

// Optimize slave MRFs

$\forall k, c$, compute minimizer $\widehat{\mathbf{x}}_c^k$ of slave MRF on clique c of k-th sample;

// Update $\mathbf{w}$

compute $d\mathbf{w}$ according to Eq.(3.44);

update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \alpha_t \cdot d\mathbf{w}^{(t)}$;

// Update $\boldsymbol{\theta}_c^k$

$\forall k, c, p$, update $\theta_c^k(\mathbf{x}_p)^{(t+1)}$ using Eq.(3.45);

until convergence

all known convergence rate results for subgradient methods carry over to our case. The correctness of the algorithm is guaranteed by the following general theorem according to the subgradient methods:

Theorem 1. *if multipliers* $\alpha_t \geq 0$ *satisfy* $\lim_{t\to\infty} \alpha_t = 0$ *and* $\sum_{t=0}^{\infty} \alpha_t = \infty$, *then the proposed algorithm converges to an optimal solution of problem* (3.37).

To summarize the above MRF learning method, the original problem of training a complex high-order MRF on the graph $G = (\mathcal{V}, \mathcal{C})$ is reduced to the training in parallel a series of easy-to-handle slave MRFs on sub-graph $\{G_i = (\mathcal{V}_i, \mathcal{C}_i)\}$ by applying the dual decomposition method. We choose the decomposition $\{G_i = G_c\}$ so that each sub-graph contains a high-order clique $c \in \mathcal{C}$, *i.e.* $\mathcal{V}_c = \{p | p \in c\}$ and $\mathcal{C}_c = c$. Thus it is not difficult to find the minimizer of the slave MRF, while it is a NP-hard to find the minimizer of the original MRF. In our case, the high-order clique c contains three nodes, so in each slave MRF energy we have three unary potentials $\boldsymbol{\theta}_c$ and one high-order potential.

Then, the projected subgradient method is used to do the optimization. On one hand (in the local view), we adjust the parameter w_c for each slave so that the minimizer of each slave MRF $\widehat{\mathbf{x}}_c$ should be coincide with the ground truth solution $\mathbf{x}_c^k$. On the other hand (in the global view), we modify the unary potentials $\boldsymbol{\theta}_c$ so that the minimizers of different slaves MRF agree on a common solution for each node, while the sum of the slave unary potentials of the same node should be equal to its unary potential of the original MRF.

To this end, we can achieve from a training set a proper vector $\mathbf{w}$ to parameterize the high-order potentials in MRF formulation (3.18), in order to better describe the shape modeling problem. Due to the sparsity regularizer, a large number of the cliques will be endowed with zero-value weight, *i.e.* $w_c = 0$. By eliminating those cliques which do no contribution in the MRF formulation, we obtain a sparse structure with no redundancy. In practice, one can use a certain threshold to select the useful cliques. Thus, the sparse graph $\mathcal{G} = (\mathcal{V}, \mathcal{F})$ is composed by the cliques whose corresponding weights are above the threshold t, *i.e.* $\mathcal{F} = \{c | w_c > t, c \in \mathcal{C}\}$, while the number of the cliques in the graph $\mathcal{G}$ is much smaller than in the complete graph G, *i.e.* $|\mathcal{F}| << |\mathcal{C}|$.

3.4 Experimental Validation

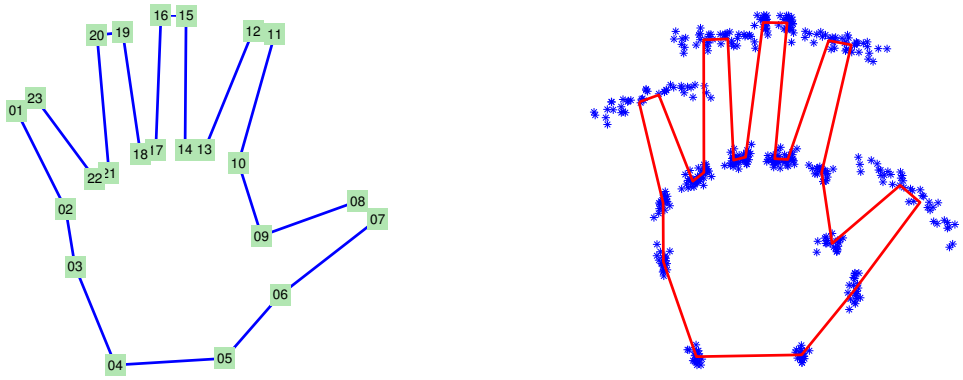
We validated the proposed shape model for both 2D and 3D shapes, in the medical and non-medical settings. The tests are done in three steps:

- The control point set of each shape example is known, and local statistics of all triplets are learned from the training set.
- MRF learning which imposes sparsity of the graph is used to estimate a parameter vector $\mathbf{w}$, and the cliques with the largest components are remained in constructing shape prior.
- Shape prior is applied in the inference to recover the mean shape.

3.4.1 2D Hand Dataset

Our 2D hand training set consists of 20 right hand examples. A number of 23 control points are manually labeled on each training sample with point correspondences across the training set. Figure 3.9 shows the landmark annotation on an example in (a) and the point distributions of the training set in (b). A fully connected higher-order graph consists of a number of 1771 triplet cliques and each clique is represented by its two inner angles (α, β). Figure 3.10 shows the training set of a clique (including point 19, 20 and 21) in angle representation in (a), and a probability density function obtained by fitting Gaussian model to the training set (where red/blue indicates large/small probability) in (b).

In MRF training procedure, we set the regularization weight $\lambda = 1$ and the step size $\alpha_t = \frac{0.01}{\sqrt{1+t}}$ at iteration t. Figure 3.11 (a) shows how the learning objective function varies during training (the converge time is 1 minute), while the estimated components of vector $\mathbf{w}$ are shown in (b), and a number of 100 selected cliques with the largest coefficient parameters above the threshold (red line) are considered, producing a sparse graph for the



(a) (b)

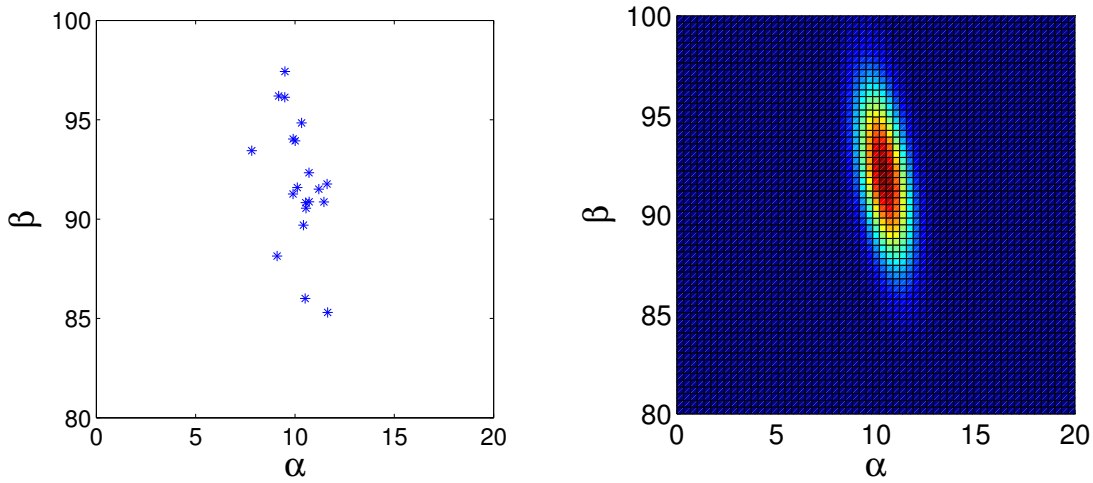
Figure 3.9: Point-based model of 2D hand. (a) Landmark labeling of an example. (b) The training set.

shape prior. Figure 3.12, from left to right, shows the cliques corresponding to the largest 20, 50 and 100 components of parameter vector $\mathbf{w}$. We can see that the obtained sparse graph (with 100 selected cliques) can cover the full range of the shape, while the finger tip points are included by more cliques than the points on the palm, which is reasonable because the finger movements cause more variations.

Then we apply the resulting shape prior in a shape inference where an optimal shape is recovered with the minimal MRF energy. This optimal shape can be considered as the mean shape of the object of interest when there is only shape prior term to define the MRF energy. The results with two initializations (in red) are shown in Figure 3.13, which shows the performance of our shape prior. Moreover, the speed of the shape inference with our sparse graph is much faster than the complete graph. For example, the test in Figure 3.13(b) using 100 cliques takes about 20 seconds to recover the mean shape, while the shape prior with complete graph (1771 cliques) takes more than 4 minutes.

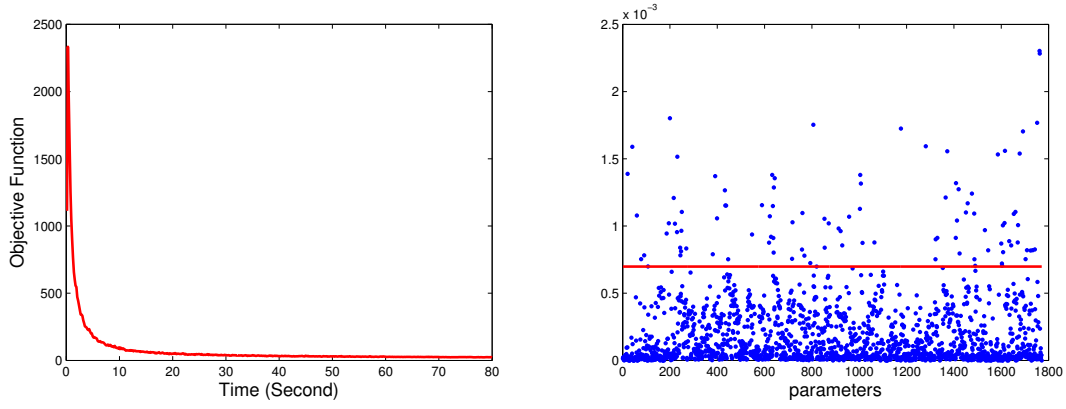
3.4.2 2D Left Ventricle Dataset

The 2D left ventricle (LV) training set consists of 78 samples from 2 subjects. A number of 40 control points are manually labeled, 24 points are on the epicardium boundary (outer contour of the myocardium muscle) and 16 points are on the endocardium boundary (inner contour of the myocardium muscle) respectively. In Figure 3.14, (a) shows the point distribution of an example and (b) shows the training set. The control points on epicardium are shown in blue, while and the points on endocardium are shown in red. This training set



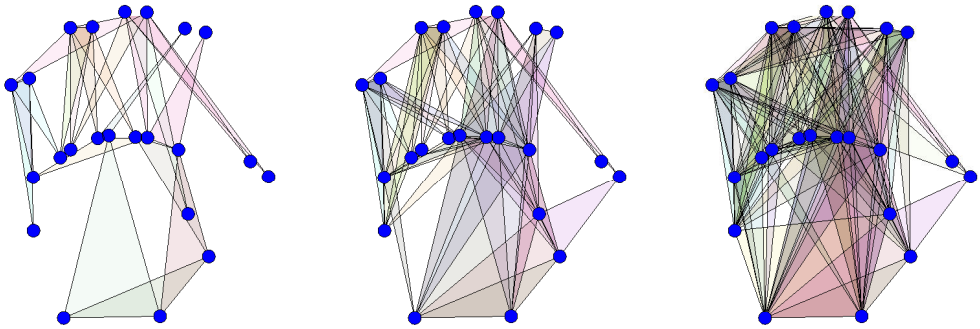
(a) (b)

Figure 3.10: Learning statistics of a clique. (a) The training set represented by angles. (b) The learned Gaussian distribution.



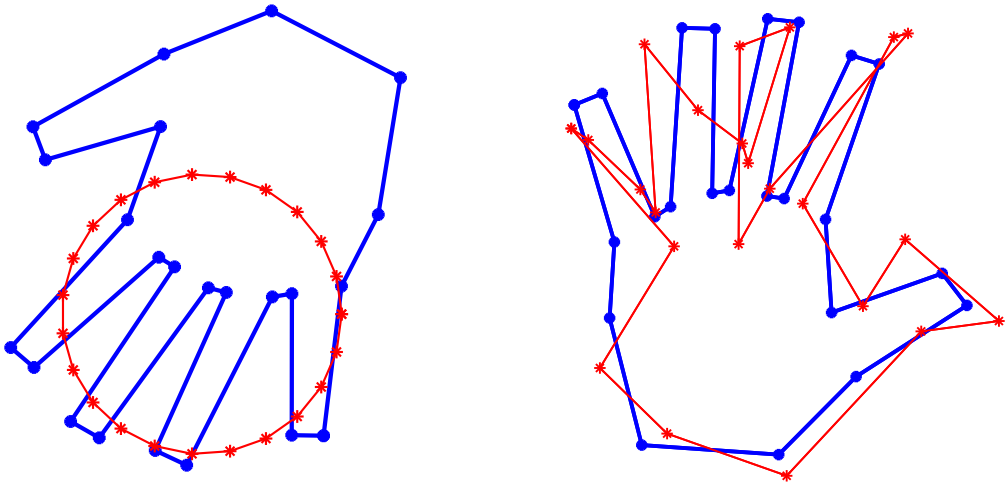
(a) (b)

Figure 3.11: MRF learning with hand dataset. (a) Primal objective function during training. (b) Learned parameters w.



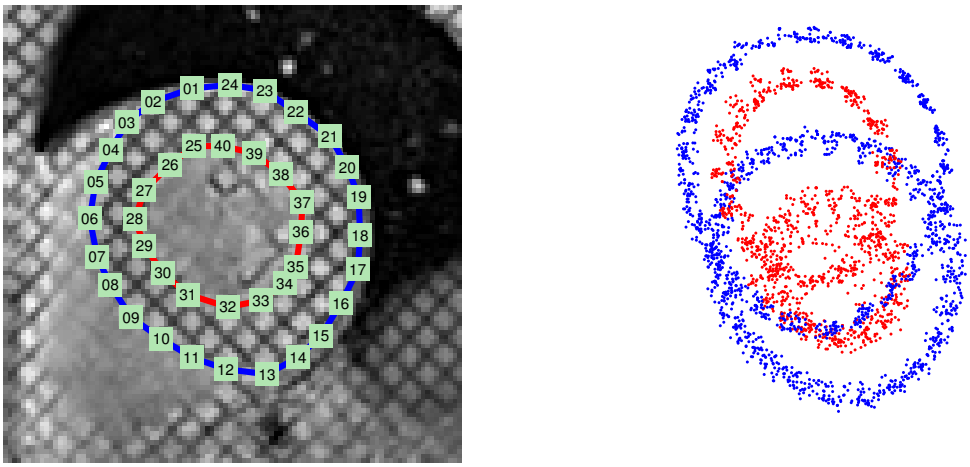
(a) 20 cliques (b) 50 cliques (c) 100 cliques

Figure 3.12: Cliques with the largest component values of the parameter vector $\mathbf{w}$.



(a) (b)

Figure 3.13: Hand shape prior applied to two initializations.



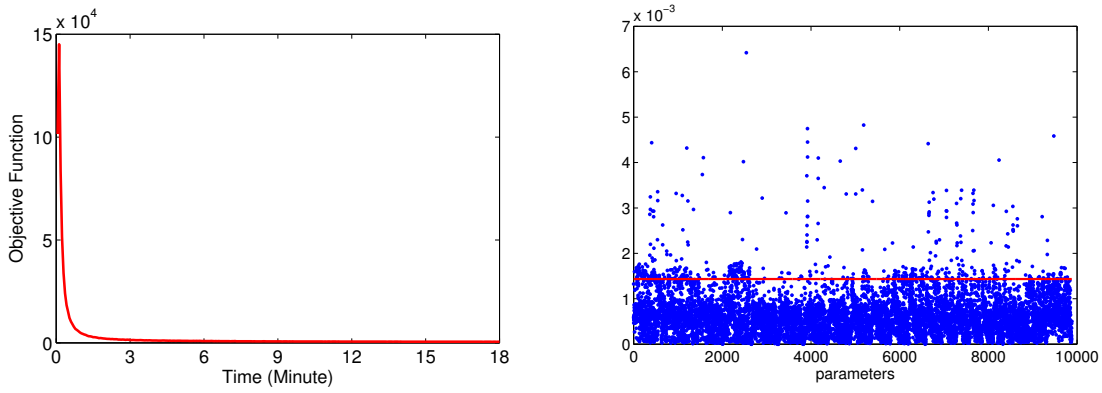
(a) (b)

Figure 3.14: Point-based model of 2D LV. (a) Landmark labeling of an example. (b) The training set.

exhibits the shape changes with respect to the global pose. The fully connected graph with a number of 9880 triplet cliques is fed to the MRF learning. We show the learning result in Figure 3.15 with the regularization weight $\lambda = 1$ and the step size $\alpha_t = \frac{0.1}{\sqrt{1+t}}$ at iteration t, where (a) shows the objective function in 18 minutes and (b) shows the estimated parameters of all the cliques. A number of 500 cliques with the largest parameters are chosen in the sparse graph. We apply the obtained shape prior in an MRF inference where the optimal solution is the mean shape. Figure 3.16 shows the results using two different initializations, the initial point positions are marked in red and the optimized positions are marked in blue, while the blue circle indicates the first point in the model and the red circle indicates the last point. The result in Figure 3.16 (b) takes 30 seconds, while the inference with all the 9880 cliques takes more than 16 minutes.

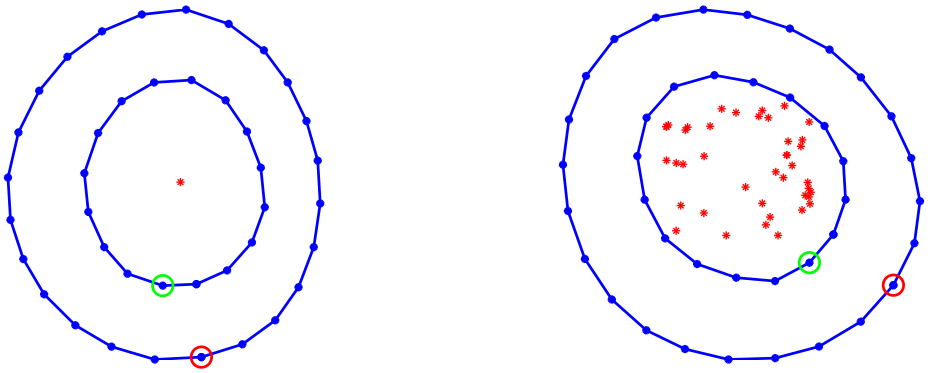
3.4.3 3D Left Ventricle Dataset

The 3D LV dataset consists of 20 3D Computed Tomography (CT) cardiac volumes and their segmentations of the left ventricle are manually done on the dataset. Randomly chosen one example, a number of 88 control points are manually labeled on the myocardium surface from the segmentation. Then a registration step is applied to estimate the correspondences of the control points in the other images, which generates the training set of the shape samples. The point-based model of the left ventricle is shown in Figure 3.1. The complete graph consists of 109736 triplet cliques. We show the learning result in Figure



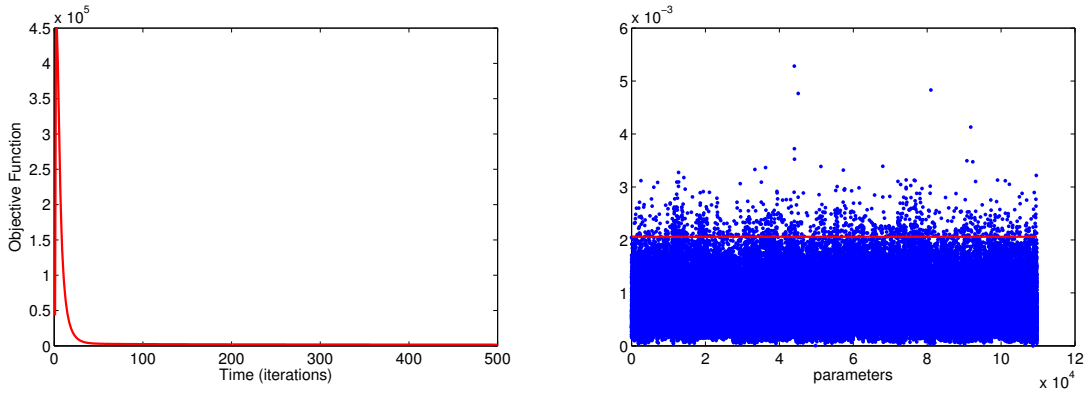
(a) (b)

Figure 3.15: MRF learning with 2D heart dataset. (a) Primal objective function during training. (b) Learned parameters $\mathbf{w}$.



(a) (b)

Figure 3.16: 2D LV shape prior applied to two initializations.



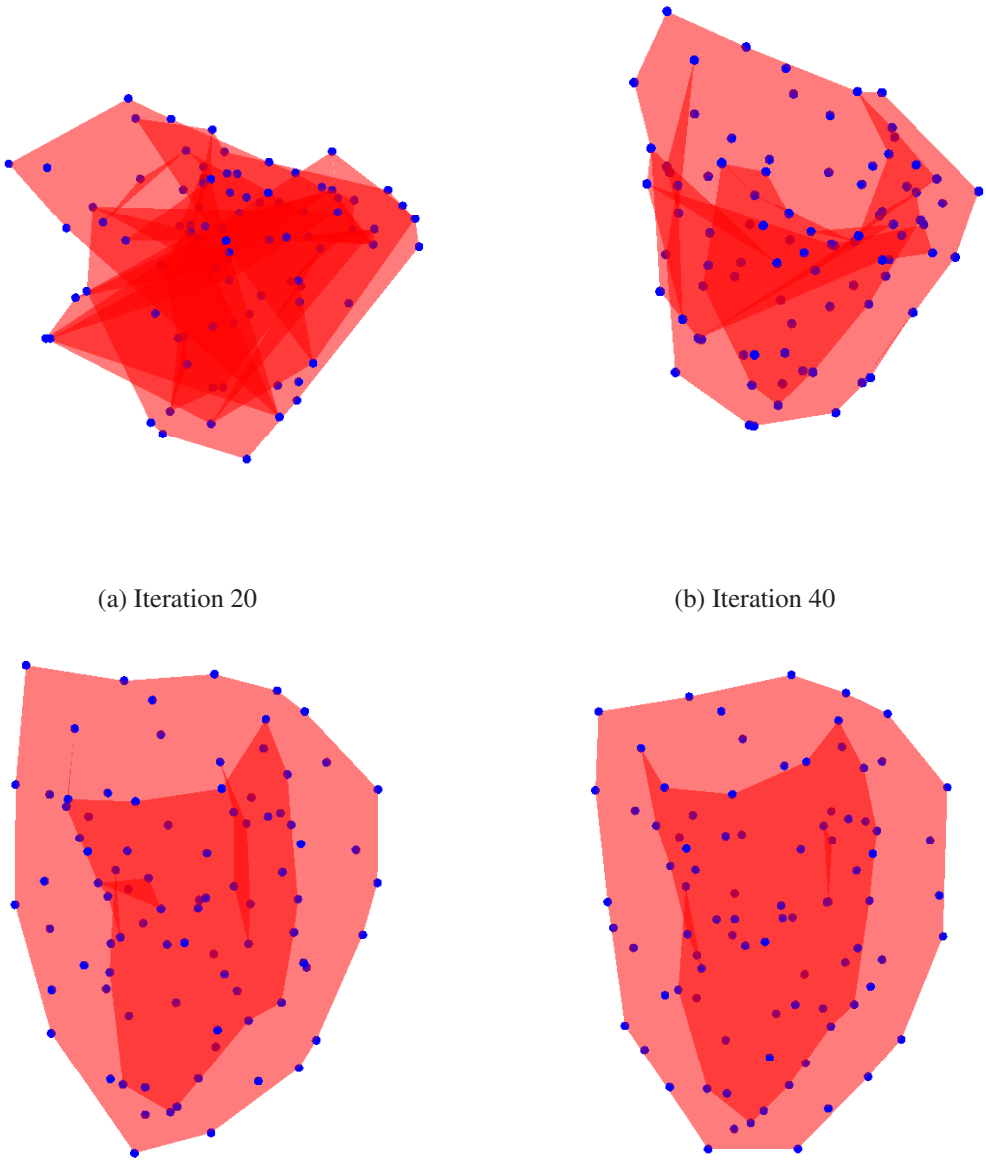
(a) (b)

Figure 3.17: MRF learning with 3D LV dataset. (a) Primal objective function during training. (b) Learned parameters $\mathbf{w}$.

3.17 with the regularization weight $\lambda = 1$ and the step size $\alpha_t = \frac{0.1}{\sqrt{1+t}}$ at iteration t, and a number of 1000 cliques with the largest parameters are selected to construct the spare graph. To test the learned shape manifold, we minimize the MRF energy defined by the shape prior terms. Figure 3.18 shows the shape deformation from an initialization where all the control points are set to the origin of coordinates, (a)-(d) shows the result after 20,40,60,80 iterations respectively. The optimization process for this initialization takes 10 minutes using the 1000 cliques, while the complete graph with 109736 cliques takes hours for the inference.

3.5 Conclusion

We represent the shape model as a point-based graphical model, where each node in the graph corresponds to a control point on the shape boundary, while each clique corresponds to the dependencies of local control points. Choosing the clique size as three, the local spacial constraint of the three points is modeled by the statistics on the angle measurement which inherits invariance to global transformations. The shape manifold is constructed through the L_1 sparse higher-order graph, accumulating the local constraints. The sparse graph consists of a subset of cliques from all possible second-order cliques, and it is learned through MRF training using dual decomposition. The pose-invariant shape prior through sparse higher-order graph can be easily encoded in a higher-order Markov Random Field.



(a) Iteration 20

(b) Iteration 40

(c) Iteration 60

(d) Iteration 80

Figure 3.18: Minimization MRF energy using shape prior of the left ventricle.

Chapter 4

Model-based Segmentation with Shape Priors

In this chapter, we incorporate the prior knowledge in a novel framework for model-based segmentation. We address the segmentation problem as a maximum a posteriori (MAP) estimation in a probabilistic framework. A global MRF energy function is defined to jointly combine regional statistics, boundary support as well as shape prior knowledge for estimating the optimal model. The considered framework is optimized using dual decomposition and applied in 2D and 3D object segmentation with promising result.

4.1 Introduction

The main idea of knowledge-based segmentation is to combine the image information with prior knowledge learned from a training set in order to find the object boundaries, therefore it can cope with occlusions, non-discriminative visual support and noise.

Early approaches adopted snake-based formulations and sought to impose constraints on the interpolation coefficients of the basis functions [Kass 1988]. Active shape models [Cootes 1995] and their visual variance have been a fundamental step towards modeling globally shape variations through principal component analysis on a set of training examples, and they used the associated sub-space for manifold-constrained segmentation. Level set methods have been also endowed with priors either including simple average models [Chen 2002], subspaces [Rousson 2008] or to certain extend pose invariance [Cremers 2006a].

The graph-theoretic approaches were also considered in knowledge-based segmentation. In [Freedman 2005], shape constraints were used iteratively to modify the graph potentials towards imposing prior knowledge by the means of mean shape. Direct mod-

eling of prior knowledge within graphs have been presented either using global priors within the random walker algorithm [Grady 2006] or through modeling the segmentation over the optimization of a graph corresponding to the point distribution model. For example, prior knowledge was modeled through statistical definition of the pairwise constraints in [Seghers 2008, Besbes 2009]. Unfortunately these methods were not pose invariant (*i.e.* invariant to translation, rotation and scale of the global shape) and could not model properly data support. This problem was partially addressed in [Wang 2010] through a fully connected complex graph with computational complexity being the main bottleneck. We also mention that our work is related to the deformable image registration approach [Glocker 2009, Glocker 2011, Zikic 2010] which is based on Markov Random Fields and integrates prior knowledge on the deformation through a set of control points.

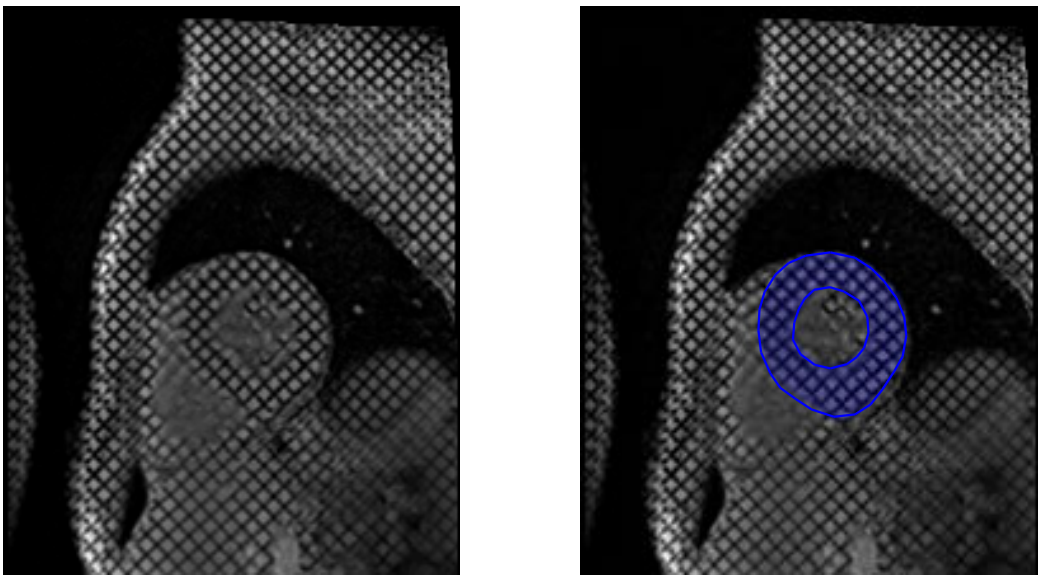
In this chapter, we propose a model-based segmentation using higher-order Markov Random Field. We develop a global approach to jointly encode the regional statistics, boundary support, as well as prior knowledges within a probabilistic framework. The pose-invariant priors are encoded by second-order MRF potentials. The regional statics is exactly factorized into pairwise or second-order terms using Divergence theorem. The proposed segmentation method is robust to noise, partial object occlusions and initializations. It is efficient and does not suffer from bad local minima issues using developed MRF optimization algorithms. Hand-pose segmentation and left ventricle segmentation are used as examples to demonstrate the potential of the method.

The remaining of the chapter is organized as follows. In Section 4.2, we introduce the probabilistic framework for model-based segmentation. Section 4.3 describes the image supports including regional statistics and boundary supports and Section 4.4 defines the Markov Random Filed formulation. Experimental validation is shown in Section 4.5 while Section 4.6 concludes the chapter.

4.2 Probabilistic Framework

We consider the segmentation task as extracting the boundaries between object and background in the observed image (*e.g.* see Figure 4.1), where the object boundaries can be constrained by prior knowledge about the object shape. The method of integrating the shape priors into image segmentation has the advantage of being robust to image noise, object occlusions and complicated background, thus producing reliable segmentation.

In this context, our aim is to estimate an optimal object boundary which is modeled by the learned manifold while being consistent with the visual measurements in an observed image. We formulate the segmentation problem as a *maximum a posteriori* (MAP) estimation in a probabilistic framework. To be specific, given an image $\mathbf{I}$, we estimate the



(a) A tagged cardiac MRI (b) Manual segmentation

Figure 4.1: Left ventricle segmentation.

optimal solution $\mathbf{X}^{\text{opt}}$ (object boundaries) by maximizing the posterior probability over the model space.

$$\mathbf{X}^{\text{opt}} = \arg\max_{\mathbf{X}} p(\mathbf{X}|\mathbf{I}) \tag{4.1}$$

where $p(\mathbf{X}|\mathbf{I})$ denotes the posterior distribution over the unknown $\mathbf{X}$ given the image $\mathbf{I}$. According to Bayes' Rule, the posterior distribution can be obtained by:

$$p(\mathbf{X}|\mathbf{I}) = \frac{p(\mathbf{I}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{I})} \tag{4.2}$$

- $p(\mathbf{I}|\mathbf{X})$ is the conditional probability, or the likelihood of the image $\mathbf{I}$ given a particular model state $\mathbf{X}$.
- $p(\mathbf{X})$ is the prior distribution over the unknown $\mathbf{X}$.
- $p(\mathbf{I})$ is a normalizing constant used to make the $p(\mathbf{X}|\mathbf{I})$ distribution integrate to 1.

Taking the negative logarithm of both sides of Eq.(4.2), we have

$$-\log p(\mathbf{X}|\mathbf{I}) = -\log p(\mathbf{I}|\mathbf{X}) - \log p(\mathbf{X}) + C \tag{4.3}$$

which is the *negative posterior log likelihood*. Since the image $\mathbf{I}$ is a fixed observation, the constant $\log p(\mathbf{I})$ can be dropped during the optimization. The Bayesian framework has two advantages:

1. The conditional probability $p(\mathbf{I}|\mathbf{X})$ of an observation given a model state is often easier to model than the posterior distribution.

2. The prior distribution $p(\mathbf{X})$ allows to introduce prior knowledge in order to produce reliable estimates and to cope with low-level information.

The *maximum a posteriori* (MAP) estimation (4.1) of the most likely solution $\mathbf{X}$ given the image $\mathbf{I}$ is equivalent to minimize the negative posterior log likelihood which can be considered as an energy $E(\mathbf{X}, \mathbf{I})$ of the model $\mathbf{X}$ and the image $\mathbf{I}$.

$$\begin{aligned} \mathbf{X}^{\text{opt}} &= \arg\min_{\mathbf{X}} E(\mathbf{X}, \mathbf{I}) \\ E(\mathbf{X}, \mathbf{I}) &= E_{\text{data}}(\mathbf{X}, \mathbf{I}) + E_{\text{prior}}(\mathbf{X}) \end{aligned} \tag{4.4}$$

where the energy $E(\mathbf{X}, \mathbf{I})$ is the sum of the data energy and the prior energy:

- The data energy $E_{\text{data}}(\mathbf{X}, \mathbf{I}) = -\log p(\mathbf{I}|\mathbf{X})$ measures the negative logarithm of the likelihood of the observed image $\mathbf{I}$ given the model state $\mathbf{X}$. This term attracts the model to locate on the desired image features (*e.g.* the object boundary).

- The prior energy $E_{\text{prior}}(\mathbf{X}) = -\log p(\mathbf{X})$ measures the negative logarithm of the likelihood of the shape configuration. This term imposes the constraint on the geometric shape of the model in order to produce a valid shape.

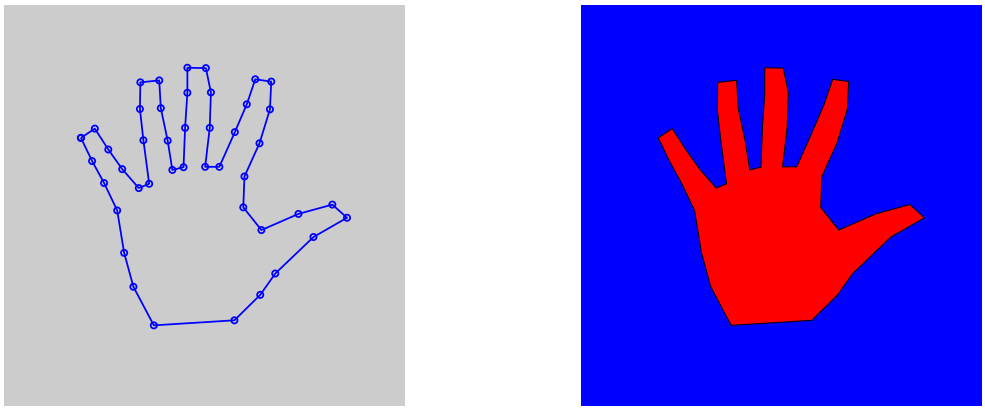
4.3 Image Support

The data energy $E_{\text{data}}(\mathbf{X}, \mathbf{I})$, also called as *image support*, attracts the model to the desired object boundary in terms of image visual properties. To model the image-based attraction, there are two typical image supports: boundary-based and region-based measurements, which are widely used to determine the data likelihood given a configuration of the model. These two image supports are based on the assumption that the populations (objects and background) have their own proper feature (*i.e.* intensity, color, texture) properties which are used to distinguish from each other in the image.

In order to facilitate the presentation, let us consider the bi-modal case where there are two populations in the image, *i.e.* object and background. We define the related notations:

- Let $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ denote the boundary points of the object of interest (*e.g.* see the blue circles in Figure 4.2 (a)), where $\mathbf{x}_i$ indicates the position of i-th point.

- Let $B(\mathbf{X})$ denote the model boundary (see the blue contour in Figure 4.2 (a)) which connects the model points, forming a closed curve in 2D case or a closed surface in 3D case.



(a) Model boundary $B(\mathbf{X})$ (b) Image partition $\Omega(\mathbf{X})$

Figure 4.2: Image measurements of a hand model.

- Let $\Omega(\mathbf{X}) = \{\Omega_{\text{obj}}(\mathbf{X}), \Omega_{\text{bck}}(\mathbf{X})\}$ be the image partition with two non-overlapping regions according to the model. For example in Figure 4.2 (b), the object region $\Omega_{\text{obj}}(\mathbf{X})$ and background region $\Omega_{\text{bck}}(\mathbf{X})$ are shown in red and blue respectively.

We also assume that some prior knowledge regarding the expected properties of the object region, the background region and the boundary between them is available.

- Let p_b denote the boundary probability, which measures how likely a pixel being located on the real boundary between object and background.

- Let p_{obj} and p_{bck} denote the region probabilities, which measure the likelihood of a pixel being part of object and being part of background respectively.

4.3.1 Boundary-based Module

Boundary-based support characterizes the discontinuity properties between different regions. It encourages the model boundary to be located on the real boundary between object and background in the image. Given the model boundary $B(\mathbf{X})$, and we assume that the pixels which belong to the model boundary are independent. Then the boundary-based data likelihood can be considered as the product of the boundary probabilities of the pixels on the model boundary being the real boundary. Equivalently, the negative log of the conditional probability is the sum of the negative log of the boundary probability of each pixel on the boundary.

In the continuous representation, the boundary-based energy is defined as an integration of the negative log of the boundary probabilities along the model boundary.

$$\begin{aligned} E_{\mathrm{B}}^{(1)}(\mathbf{X},\mathbf{I}) &= \oint_{B(\mathbf{x})} G_{\mathrm{B}}^{(1)}(x(s),y(s))ds \\ E_{\mathrm{B}}^{(2)}(\mathbf{X},\mathbf{I}) &= \oiint_{B(\mathbf{x})} G_{\mathrm{B}}^{(2)}(x(s),y(s),z(s))ds \end{aligned} \tag{4.5}$$

where the function G_{B} denotes the negative logarithm of the boundary probability p_b, and $E_{\mathrm{B}}^{(1)}$ and $E_{\mathrm{B}}^{(2)}$ are the boundary energies in 2D and 3D cases respectively.

$$\begin{aligned} p_{\mathrm{B}}(\mathbf{I}|\mathbf{X}) &= \prod_{i\in B(\mathbf{X})} p_b(i) \\ -\log p_{\mathrm{B}}(\mathbf{I}|\mathbf{X}) &= \sum_{i\in B(\mathbf{X})} -\log p_b(i) \end{aligned} \tag{4.6}$$

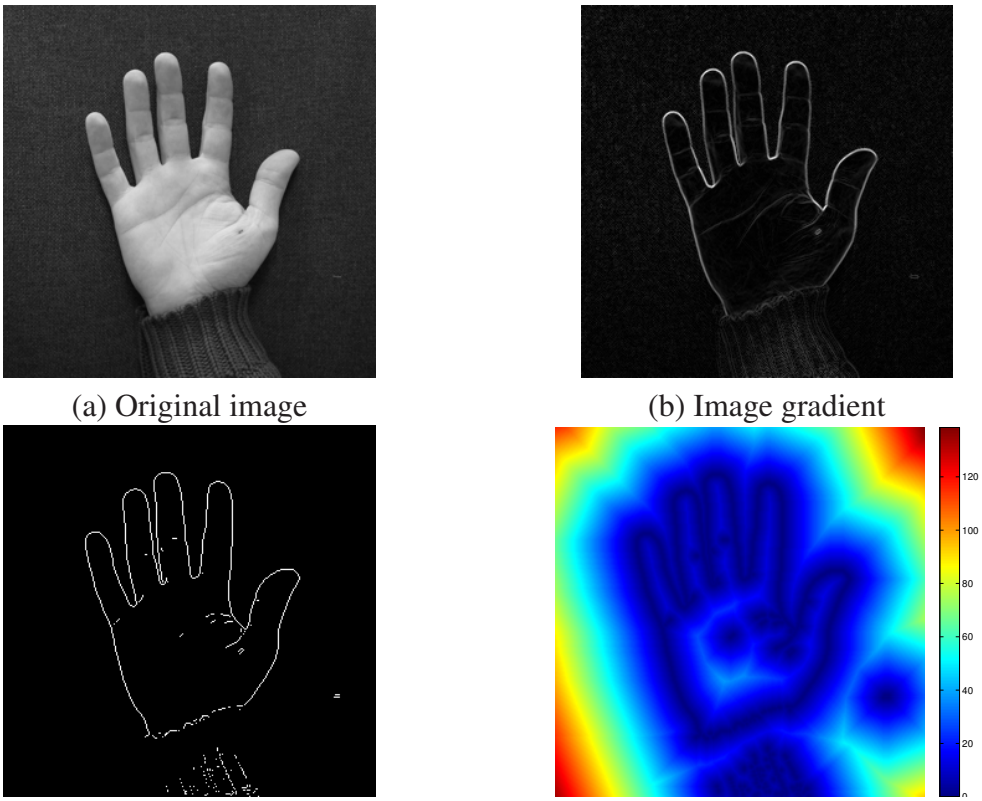
where i is a pixel/voxel on the model boundary $B(\mathbf{X})$ and $p_b(i)$ denotes the probability of the pixel/voxel i being a real boundary.

Given an image $\mathbf{I}$, the boundary function G_{B} determines a value for each position inside the image, such that if the position is highly likely to be on the real boundary, the value is small, otherwise the value is large. In practice, the widely used way to generate the boundary image G_{B} is to use the gradient information of the image (*e.g.* see the scalar field in Figure 4.3 (b): white represents higher value and black represents lower value) because the boundaries between different regions usually exhibit high gradient values. For example, the boundary function G_{B} can be defined directly as the image gradients or a smoothed version of the image Laplacian:

$$G_{\mathrm{B}} = -\left\|\nabla\mathbf{I}\right\|^2 \ or - \left|\left(G_\sigma * \nabla^2\mathbf{I}\right)\right|^2 \tag{4.7}$$

where G_σ is a Gaussian of standard deviation σ.

Alternatively, the boundary function G_{B} can be considered as a distance map to the edges. It is acquired by two steps: (1) We apply an edge detector (*i.e.* Sobel operator) on the observed image to detect the edges. For example, Figure 4.3 (c) shows the edges in the original grayscale image (a) using Sobel operator, and the output is a binary image where the pixels on the edges are assigned to 1 (in white), otherwise they are assigned to 0 (in black). (2) Then we use distance transform of the edge response to generate the distance map. Figure 4.3 (d) shows a distance transform of the binary image of edges. The resulting distance map is shown in a colormap where the higher values are represented in red, and the lower values are represented in blue. The scale value of each pixel equals to



(a) Original image (b) Image gradient

(c) Edges in the image (d) Distance transform

Figure 4.3: Boundary-based information.

its distance to the nearest edge. Thus if the pixel is close to the edges, the function returns a small value.

To minimize the boundary-based energy means that the model boundary is attracted by strong edges corresponding to pixels with local-maxima image gradient values. The edge-based information is easy to implement with low computational cost, but it makes the model sensitive to noise and spurious edges. For this reason, it requires to initialize the model close to the real object boundary in order to avoid getting trapped in local minima.

4.3.2 Region-based Module

Region-based energy captures the homogeneity properties of different populations observed in the image. It encourages the model boundary producing the image partition

which is consistent with the statistics properties of the object and the background. A given model boundary $B(\mathbf{X})$ partitions the image domain into object region $\Omega_{\text{obj}}(\mathbf{X})$ and background region $\Omega_{\text{bck}}(\mathbf{X})$. Assuming that there is no correlation between the regions labeling, the region-based likelihood can be computed as follows.

$$p_{\text{R}}(\mathbf{I}|\mathbf{X}) = p(\mathbf{I}|\Omega_{\text{obj}}(\mathbf{X}))\, p(\mathbf{I}|\Omega_{\text{bck}}(\mathbf{X})) \tag{4.8}$$

where $p(\mathbf{I}|\Omega_{\text{obj}}(\mathbf{X}))$ and $p(\mathbf{I}|\Omega_{\text{obj}}(\mathbf{X}))$ are the posterior probabilities given the object region $\Omega_{\text{obj}}(\mathbf{X})$ and the background region $\Omega_{\text{bck}}(\mathbf{X})$ respectively.

Furthermore, assuming that the pixels within each region are independent, the region probability can be computed by the product of the pixel probabilities. In this context, the a posteriori probability of a image partition $\Omega(\mathbf{X})$ is determined by:

$$p_{\text{R}}(\mathbf{I}|\mathbf{X}) = \prod_{i \in \Omega_{\text{obj}}(\mathbf{X})} p_{\text{obj}}(\mathbf{I}_i) \prod_{i \in \Omega_{\text{bck}}(\mathbf{X})} p_{\text{bck}}(\mathbf{I}_i) \tag{4.9}$$

where $\mathbf{I}_i$ denotes the observed image feature (*e.g.* intensity, RGB values or a feature vector) of the pixel/voxel i, while p_{obj} and p_{bck} are the appearance distribution models of the object and the background respectively. If a pixel/voxel i is in the object region $\Omega_{\text{obj}}(\mathbf{X})$, we calculate the probability of the pixel i being the region, otherwise we calculate the probability of the pixel i being the background. As a result, the regional likelihood encourages that the object region $\Omega_{\text{obj}}(\mathbf{X})$ covers the pixels that exhibit the object properties, and background region $\Omega_{\text{bck}}(\mathbf{X})$ covers the pixels that exhibit the background properties. Now taking the negative logarithm of the above equation, we have:

$$\begin{aligned} -\log p_{\text{R}}(\mathbf{I}|\mathbf{X}) &= \sum_{i \in \Omega_{\text{obj}}(\mathbf{X})} -\log p_{\text{obj}}(\mathbf{I}_i) + \sum_{i \in \Omega_{\text{bck}}(\mathbf{X})} -\log p_{\text{bck}}(\mathbf{I}_i) \\ &= \sum_{i \in \Omega_{\text{obj}}(\mathbf{X})} -\log \frac{p_{\text{obj}}(\mathbf{I}_i)}{p_{\text{bck}}(\mathbf{I}_i)} + \sum_{i \in \Omega} -\log p_{\text{bck}}(\mathbf{I}_i) \\ &= \sum_{i \in \Omega_{\text{obj}}(\mathbf{X})} -\log \frac{p_{\text{obj}}(\mathbf{I}_i)}{p_{\text{bck}}(\mathbf{I}_i)} + \text{constant} \end{aligned} \tag{4.10}$$

The regional energy is originally defined as the sum of the regional likelihood of different populations. Since the sum of the likelihood of the background over the entire image domain $\Omega = \Omega_{\text{obj}}(\mathbf{X}) \cup \Omega_{\text{bck}}(\mathbf{X})$ is a constant value (which will not change during the optimization), it can be ignored in the regional energy. Thus the regional support can be simplified by the integration over only the object region.

In brief, the region-based energy is defined by the integration over the object region $\Omega_{\text{obj}}(\mathbf{X})$ which is determined by the model $\mathbf{X}$. For a pixel/voxel i inside the object region:

- Compute $p_{\text{obj}}(\mathbf{I}_i)$ and $p_{\text{bck}}(\mathbf{I}_i)$ which are the probabilities of the pixel/voxel being the object and being the background respectively.
- If $p_{\text{obj}}(\mathbf{I}_i) > p_{\text{bck}}(\mathbf{I}_i)$ which means the pixel/voxel is more likely being the object, then the integral function returns a negative value *i.e.* $-\log \frac{p_{\text{obj}}(\mathbf{I}_i)}{p_{\text{bck}}(\mathbf{I}_i)} < 0$.
- If $p_{\text{obj}}(\mathbf{I}_i) < p_{\text{bck}}(\mathbf{I}_i)$ which means the pixel/voxel is more likely being the background, then the integral function returns a positive penalty *i.e.* $-\log \frac{p_{\text{obj}}(\mathbf{I}_i)}{p_{\text{bck}}(\mathbf{I}_i)} > 0$.

In this manner, the region-based energy encourages the object model to contain as many object pixels (which are more likely being the object) as possible, and as less background pixels (which are more likely being the background) as possible. The region-based energy in continuous representation is given as follows:

$$\begin{aligned} E_{\text{R}}^{(1)}(\mathbf{X}, \mathbf{I}) &= \iint\limits_{\Omega_{\text{obj}}(\mathbf{X})} -\log \frac{p_{\text{obj}}(\mathbf{I}(x,y))}{p_{\text{bck}}(\mathbf{I}(x,y))} dxdy \\ E_{\text{R}}^{(2)}(\mathbf{X}, \mathbf{I}) &= \iiint\limits_{\Omega_{\text{obj}}(\mathbf{X})} -\log \frac{p_{\text{obj}}(\mathbf{I}(x,y,z))}{p_{\text{bck}}(\mathbf{I}(x,y,z))} dxdydz \end{aligned} \tag{4.11}$$

where $E_{\text{R}}^{(1)}$ and $E_{\text{R}}^{(2)}$ are the regional energies of 2D and 3D cases respectively. A spatial position is represented by its coordinates, *i.e.* (x, y) or (x, y, z) in 2D or 3D cases.

The region-based support uses the homogeneity properties of the image regions to measure how well the model is fitted to the observed image. Given the model $\mathbf{X}$, all the pixels or voxels inside the model boundary are considered in the measurement (global image information), while boundary-based support only takes into account the pixels/voxels on the model boundary (local image information). Thus the regional energy exhibits less local minima than the boundary energy which relies on gradient information along the boundary, and it can make the segmentation less sensitive to noise and initializations. However, the regional support is based on the assumption that different populations can be well distinguished by their appearances in the image. If different objects are not separable in the appearance space, the regional cue will misguide the segmentation. Regarding this fact, it is a critical issue to model the appearance of different objects properly.

4.3.3 Appearance Models

In this subsection, we describe how to model the appearance of different classes of objects (*e.g.* the probability density functions p_{obj} and p_{bck}), using a set of training images. The

appearance attributes (*i.e.* intensity, color, texture) of the object of interest are the most significant visual evidences in computer vision and image processing tasks. The appearance knowledge of the object class is very essential to facilitate the task of object detection, although the image cues are sensitive to noise, partial object occlusions and complicated background. Modeling the object appearance consists of two components:

- Choosing a proper representation of the appearance features describing the object attributes. The selected features should allow a good discrimination between different classes.

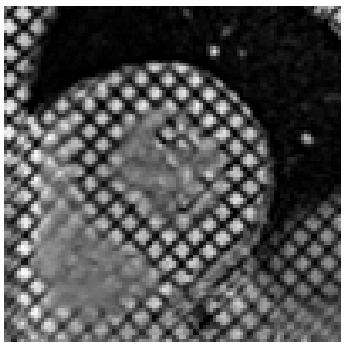
- Learning a discriminative model in order to estimate the class identity or the probability of the new sample.

Regarding the appearance features, the intensity or color of the pixel (same for voxel) is the direct representation, since each pixel is associated with an intensity value or the color at that location. The intensity or color information of the pixel is wildly used to determine the class of the pixel when assuming that the different classes of objects are obviously different in their intensities. However, the information with only pixel intensity at the examined location is often not inadequate to decide if a certain class of object occurs. For example, Figure 4.4 shows a tagged MR cardiac image where a certain pattern of texture occurs. The edges of the original image in Figure 4.4 (c) generate misleading information of real boundary. Moreover, the pixel intensity information is not enough to distinguish the object and the background. Figure 4.4 (d) shows the intensity distribution of the object pixels (in red) and the distribution of the background pixels (in blue) based on ground truth segmentation 4.4 (b), however these two distributions are heavily overlapped.

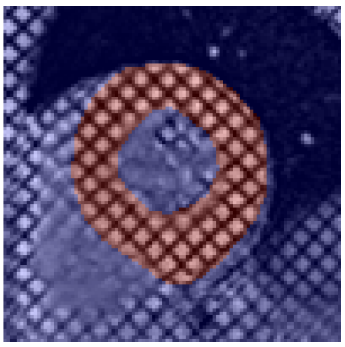
In this case, we need to use the information of a neighborhood of the pixel in order to make the right decision. In particular, it is very necessary when the object class exhibits texture patterns, which makes the appearance modeling more challenging. In the context of computer vision, textures are defined as repeating patterns of local variation of pixel intensities, which means texture features cannot be defined by one single pixel, but by its neighborhood.

Gabor Features

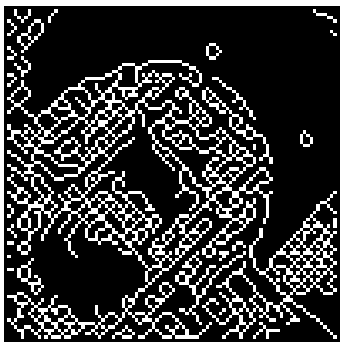
In order to extract the texture features from an image, we employ a multi-resolution representation based on Gabor filters [Manjunath 1996]. Gabor features have been used in various image analysis applications such as texture classification and segmentation. It is motivated by the fact that a set of Gabor filters with different frequencies and orientations are very helpful to characterize the underlying texture information.



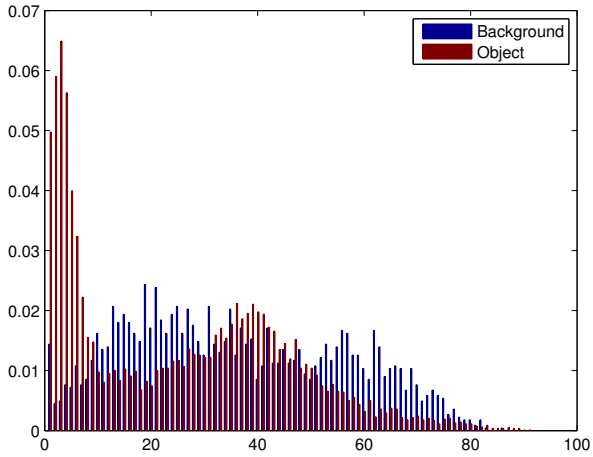
(a) Original image



(b) Ground truth segmentation



(c) Edges



(d) Intensity distributions of object and background

Figure 4.4: Image appearance of a tagged cardiac MRI.

Gabor filter is essentially a Gaussian filter modulated by a sinusoid. A 2D Gabor function $g(x, y)$ and its Fourier transform $G(u, v)$ can be written as:

$$\begin{aligned} g(x,y) &= \frac{1}{2\pi\sigma_x\sigma_y}\exp\Big[-\frac{1}{2}\Big(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\Big)+2\pi jWx\Big] \\ G(u,v) &= \exp\Big[-\frac{1}{2}\Big(\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\Big)\Big] \end{aligned} \tag{4.12}$$

where $\sigma_u = \frac{1}{2\pi\sigma_x}$ and $\sigma_v = \frac{1}{2\pi\sigma_y}$. The parameters σ_x and σ_y characterize the spatial extent and frequency bandwidth of the Gaussian filter, and the parameter W sets the frequency of the sinusoid. Let $g(x, y)$ be the mother function of the Gabor filter family, a set of

Gabor functions $g_{m,n}(x, y)$, referred to as *Gabor wavelets*, can be generated by dilations and rotations of $g(x, y)$ to form a complete but non-orthogonal basis set.

$$\begin{aligned} g_{m,n}(x, y) &= a^{-m} g(x', y') \\ x' &= a^{-m}(x \cos\theta_n + y \sin\theta_n) \\ y' &= a^{-m}(-x \sin\theta_n + y \cos\theta_n) \end{aligned} \tag{4.13}$$

where $a > 1$ and $\theta_n = \frac{n\pi}{K}$, $m = \{0, 1, \cdots, S-1\}$ and $n = \{0, 1, \cdots, K-1\}$. The parameter S is the number of scales and the parameter K is the number of orientations. Given an image $I(x, y)$, the image response of the Gabor wavelet $g_{m,n}$ is defined as:

$$\begin{aligned} J_{m,n}(x, y) &= I(x, y) * g_{m,n}(x, y) \\ &= \sum_k \sum_l I(k, l) g_{m,n}(x - k, y - l) \end{aligned} \tag{4.14}$$

where $*$ denotes the convolution operation. Given the parameters S and K, the image I has a number of $S \cdot K$ Gabor transforms. For example, Figure 4.5 shows the Gabor transforms of the image in Figure 4.4 (a), given the scale parameter $S = 3$ and the orientation parameter $K = 4$. Each image response of the Gabor wavelet $g_{m,n}$ (shown in a colormap where blue/red represents the small/large value) captures the texture properties according to the certain scale and orientation. For each pixel (x, y) in the image, the responses of all the Gabor wavelets at this location can be concatenated to one feature vector $\mathbf{J}(x, y)$ that describes the texture appearance:

$$\mathbf{J}(x, y) = (J_{0,0}(x, y), \cdots, J_{m,n}(x, y), \cdots, J_{S-1,K-1}(x, y))^T \tag{4.15}$$

To extract the texture features in 3D images, we use the method in [Zhan 2003] which approximates the complete set of 3D Gabor features by using two banks of 2D Gabor filters located at the orthogonal planes in order to save computation time. Moreover, [Han 2007] introduced a rotation-invariant and a scale-invariant Gabor representations, where each representation only requires few summations over the filter responses of the conventional Gabor filter family.

In general, various appearance features can be integrated into one feature vector which is a D-dimensional vector of numerical features that represent the object. Given an original image I, each pixel i of the image can be represented by its feature vector:

$$\mathbf{I}(i) = (I_0(i), \cdots, \cdots, I_{D-1}(i))^T \tag{4.16}$$

where the component $I_k(i)$ can be a particular appearance attribute property, such as the intensity of the pixel, the intensity of a neighboring pixel, the gradient, and the texture

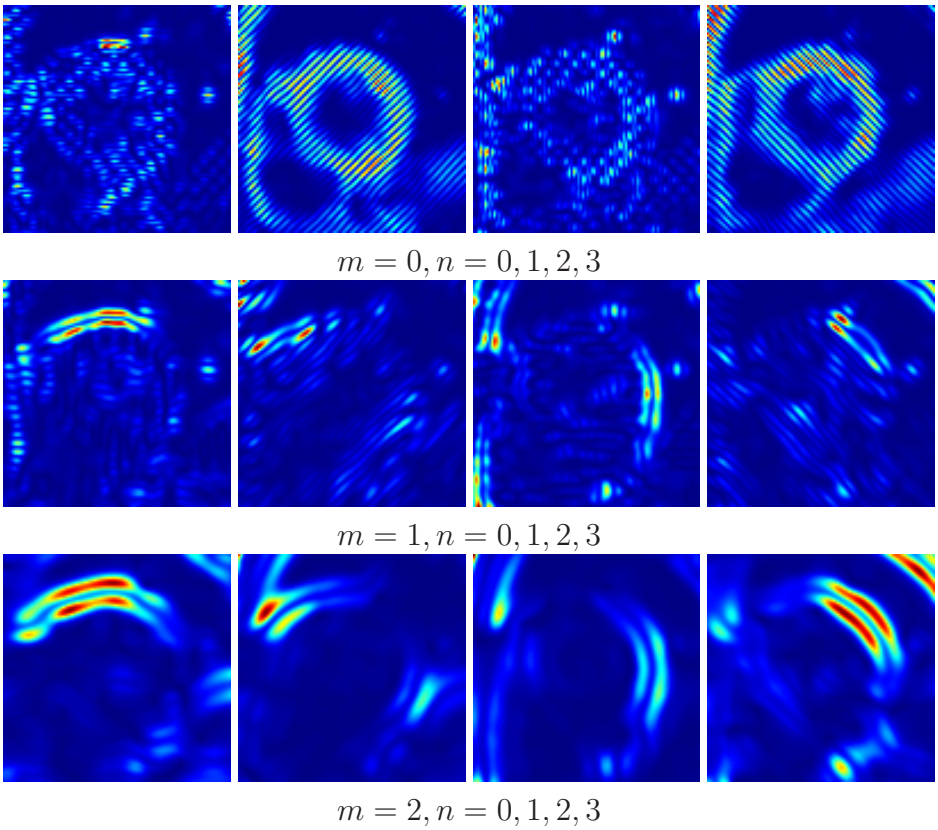


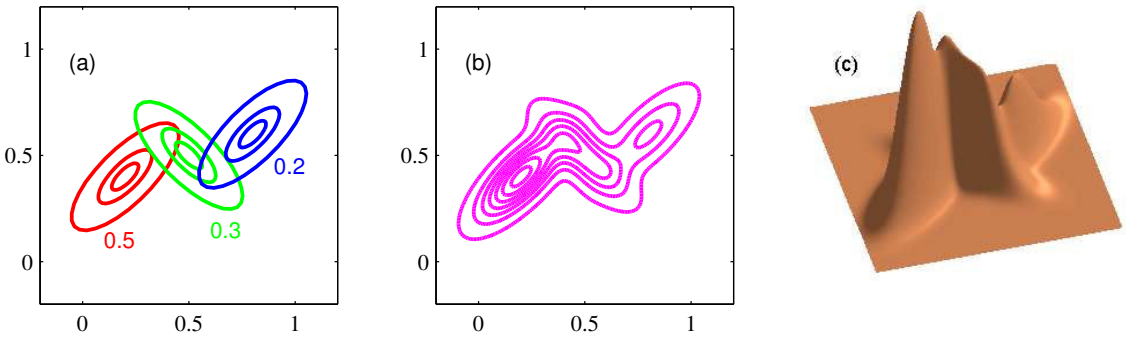
Figure 4.5: Gabor features with 3 scales and 4 orientations.

feature. In our experiments, we used a patch of intensities centered at the pixel and the Gabor features to compose the feature vector of the pixel.

Based on the feature representation, a learning phase is performed to find the appearance feature models of different classes objects. The feature models should have the power to discriminate between different classes given the observed features. We discuss two supervised learning techniques for this purpose: Gaussian Mixture Model and AdaBoost learning algorithm.

Gaussian Mixture Model

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a linear combination of Gaussian component densities. By choosing a sufficient number of Gaussians, and by estimating their means and covariances as well as the coefficients, the mixture of Gaussians is able to capture the distribution of a dataset. Given a



(a) 3 Gaussian components. (b) The mixture distribution. (c) A surface plot.

Figure 4.6: A Gaussian mixture model using 3 components in 3D space [Bishop 2006].

set of training images and their corresponding ground truth segmentation (*i.e.* labeling the image pixels into L classes), each class $c_{i\in L}$ is associated with a number N of appearance samples $\{\mathbf{f}_i^j\}_{j=1:N}$, where $\mathbf{f}$ denotes a feature vector of the pixel. We model each class $c_{i\in L}$ by using a Gaussian Mixture Model of K components of Gaussian densities. The probability of a feature vector $\mathbf{f}$ belonging to class c_i is given by:

$$p(\mathbf{f}|c_i) = \sum_{k=1}^{K} \pi_k^i \mathcal{N}(\mathbf{f}|\mu_k^i, \Sigma_k^i) \tag{4.17}$$

where each Gaussian density $\mathcal{N}(\mathbf{f}|\mu_k^i, \Sigma_k^i)$ is defined by its own mean μ_k^i and covariance matrix Σ_k^i. The parameters π_k^i are the mixing coefficients, and they satisfy that $0 \leq \pi_k^i \leq 1$ and $\sum_{k=1}^{K} \pi_k^i = 1$. The number of the components is fixed, often chosen as $K = 3$ in practice. Figure 4.6 illustrates a Gaussian mixture of 3 components. The 3 components are represented by their contours of constant density in red, blue and green with the mixing coefficients in (a), while contours of the probability density and a surface plot of the mixture distribution are shown in (b) and (c).

For each class $c_{i\in L}$, the mean $\boldsymbol{\mu}_i = \{\mu_1^i, \cdots, \mu_K^i\}$, covariance matrix $\boldsymbol{\Sigma}_i = \{\Sigma_1^i, \cdots, \Sigma_K^i\}$ and the mixing coefficients $\boldsymbol{\pi}_i = \{\pi_1^i, \cdots, \pi_K^i\}$ are estimated by using Expectation Maximization algorithm [Bishop 2006], which uses integrative optimization techniques to maximizing the log of the likelihood functions:

$$\log p(\mathbf{f}_i^1, \cdots, \mathbf{f}_i^N | \boldsymbol{\pi}_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \sum_{n=1}^{N} \log \Big\{ \sum_{k=1}^{K} \pi_k^i \mathcal{N}(\mathbf{f}_i^n | \mu_k^i, \Sigma_k^i) \Big\} \tag{4.18}$$

Therefore, we obtain a probability density function $p(\mathbf{f}|c_i)$ for each class c_i from its training data. Given a new observed data (*e.g.* a pixel described by a feature vector $\mathbf{f}$), we

can compute the probability of the element being the class c_i. When two classes (*i.e.* object and background) are considered in the image, we denote p_{obj} and p_{obj} being the Gaussian Mixture Models of the object and the background respectively.

Boosting

Boosting was proposed in the machine learning literature [Freund 1995], and "boosting" describes a procedure of combining multiple "weak" classifiers to produce a powerful "committee" whose performance can be significantly better than any weak classifier. We describe the most widely used version of the boosting algorithm called *AdaBoost* algorithm [Freund 1996], short for adaptive boosting. AdaBoost trains the base classifiers in sequence, and each new classifier adjusts the weights associated with each data point according the performance of the previous classifiers. After all the weak classifiers have been trained, the algorithm constructs a strong classifier which is a linear combination of the weak classifiers balanced by their weights.

Considering a two-class classification setting, the training data is composed by N input feature vectors $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ and their corresponding ground truth binary labels $\{y_1, \cdots, y_N\}$ where $y_{i=1:N} \in \{-1, 1\}$. Each data point is associated with a weight $w_{i=1:N}$ which is initially set equally to $\frac{1}{N}$. The learning is an iterative process and each iteration stage has three steps: (1) Choosing a new weak binary classifier h_t, using training data by minimizing the weighted error function; (2) Computing the error ϵ_t with respect to the distribution of $\{w_i^{(t)}\}$, and then using it to define the weight α_t of the weak classifier h_t; (3) Adjusting the weights of the samples according to the performance of the previous classifier h_t, so that the misclassified samples by the weak classifier h_t are given larger weight in training the next classifier in the sequence. The new weights are normalized by Z_t in order to keep a distribution. After the desired number T of training rounds, the final model is built by combining the weighted weak classifiers, and the function sign constraints the output of the classifier to be either -1 or $+1$. The learning process of AdaBoost algorithm is illustrated in Algorithm 4.1.

This version of AdaBoost algorithm is also called *Discrete AdaBoost* because the weak classifier $h_t(\mathbf{x}) \in \{-1, +1\}$ produces a binary classification. A generalized version of AdaBoost called *Real AdaBoost* [Schapire 1999] appeared to improve the boosting algorithms. The weak learner returns a class probability estimates $p_t(\mathbf{x}) \in [0, 1]$, and the final classifier is defined as the sum of half the logit-transform of the probability estimate corresponding to each weak classifier. From a statistical perspective [Freund 2000], the AdaBoost algorithms can be interpreted as fitting an additive logistic regression model $H(\mathbf{x}) = \sum_{t=1}^{T} h_t(\mathbf{x})$ by minimizing an exponential criterion $J(H)$:

$$J(H) = E(\exp\{-yH(\mathbf{x})\}) \tag{4.19}$$

Algorithm 4.1 AdaBoost.

Input: A training set with labeled pairs $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\}$.

Initialize the weights of the samples: $\forall i \in \{1, \cdots, N\}$,$w_i^{(1)} = \frac{1}{N}$.

for $t = 1$ **to** T **do**

(1) Fit the classifier $h_t(\mathbf{x}) \in \{-1, +1\}$ using the weights $\{w_i^{(t)}\}$ on the training data.

(2) Compute the error ϵ_t and the weight α_t of the weak classifier h_t:

$$\epsilon_t = \sum_{i=1}^{N} w_i^{(t)} \left[(h_t(\mathbf{x}_i) \neq y_i)\right]$$

$$\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

(3) Update the weights of the samples.

$$w_i^{(t+1)} = \frac{1}{Z_t} w_i^{(t)} \exp\left\{\alpha_t \left[(h_t(\mathbf{x}_i) \neq y_i)\right]\right\}$$

end for

return the final classifier $H(\mathbf{x})$:

$$H(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right)$$

where $E(\cdot)$ represents the expectation. The function $J(H)$ is proved to be optimized at:

$$H(\mathbf{x}) = \frac{1}{2} \log \frac{P(y=1|\mathbf{x})}{P(y=-1|\mathbf{x})} \tag{4.20}$$

As a result, we obtain the relations between the classifier and the posterior or conditional class probabilities which we are interested in and we want to use for modeling the appearance features of different classes:

$$\begin{aligned} P(y=1|\mathbf{x}) &= \frac{e^{H(x)}}{e^{-H(x)} + e^{H(x)}} \\ P(y=-1|\mathbf{x}) &= \frac{e^{-H(x)}}{e^{-H(x)} + e^{H(x)}} \end{aligned} \tag{4.21}$$

A modified version of the Real AdaBoost algorithm, named *Gentle AdaBoost* algorithm was proposed in [Freund 2000], using Newton steps for minimizing $J(H)$. Given an imperfect $H_t(\mathbf{x})$, an update $H_t(\mathbf{x}) + h_t(\mathbf{x})$ is proposed to optimize the exponential

Algorithm 4.2 Gentle AdaBoost.

Input: A training set with labeled pairs $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\}$.

Initialize the weights of the samples: $\forall i \in \{1, \cdots, N\}$,$w_i^{(1)} = \frac{1}{N}$, and $H_1(\mathbf{x}) = 0$.

for $t = 1$ **to** T **do**

(1) Fit the regression function $h_t(\mathbf{x})$ by weighted least-squares of y_i to $\mathbf{x}_i$ with weights $\{w_i^{(t)}\}$.

(2) Update $H_{t+1}(\mathbf{x}) = H_t(\mathbf{x}) + h_t(\mathbf{x})$

(3) Update $w_i^{(t+1)} = \frac{1}{Z_t} w_i^{(t)} \exp\{-y_i h_t(\mathbf{x}_i)\}$

end for

return the final classifier:

$$\text{sign}(H(\mathbf{x})) = \text{sign}\big(\sum_{t=1}^{T} h_t(\mathbf{x})\big)$$

criterion (the population version) with respect to $h_t(\mathbf{x})$ in each iteration.

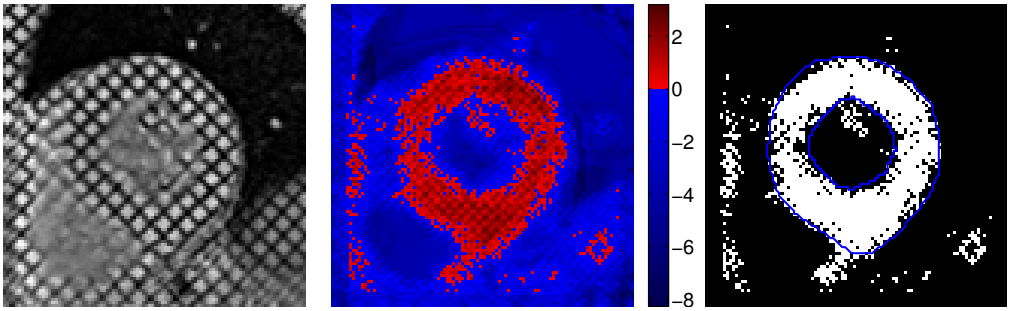
$$J(H_t(\mathbf{x}) + h_t(\mathbf{x})) = E(\exp\{-y(H_t(x) + h_t(x))\}) \tag{4.22}$$

The update $h_t(\mathbf{x})$ is given by the estimates of the weighted class probabilities:

$$h_t(\mathbf{x}) = P_w(y = 1|\mathbf{x}) - P_w(y = -1|\mathbf{x}) \tag{4.23}$$

This update $h_t(\mathbf{x})$ produces the values in the range $[-1, 1]$, while the update $h_t(\mathbf{x}) = \frac{1}{2} \log \frac{P_w(y=1|\mathbf{X})}{P_w(y=-1|\mathbf{X})}$ used in the Real AdaBoost algorithm can be numerically unstable because of the log-ratios. This modification makes the Gentle AdaBoost outperform the Real AdaBoost in practice. The Gentle AdaBoost algorithm is illustrated in Algorithm 4.2. The boosting with multiple classes is discussed in [Freund 2000].

Figure 4.7 illustrates how the AdaBoost algorithm is useful in appearance modeling. First, the strong classifier $\sum_{t=1}^{T} h_t(\mathbf{x})$ is learned from a training set where each data point is represented by a feature vector and its ground truth labeling is available. Then, we apply the obtained classifier on a test image (a) and the image response is shown in (b). Each pixel is labeled with a signed value, positive value (shown in red) representing the class of object and negative value (shown in blue) representing the class of background. The absolute value represents the scores of being the corresponding class, *e.g.* higher score is shown in dark color and lower score is shown in light color. The binary output $H(\mathbf{x})$ is shown in (c) (white/black represents the object/background), and the ground truth is outlined by blue contours in order to compare with the classification result. As we can see, the AdaBoost algorithm can provide reliable models for different classes.



(a) Test image (b) Classifier response (c) Binary output

Figure 4.7: Appearance modeling using Gentle AdaBoost classifier.

4.4 Markov Random Fields Formulation

Now we implement the above probabilistic framework within a higher-order Markov Random Fields (MRF) formulation, so that we can employ the recent developed MRF inference algorithms to achieve a good optimum with a very fast speed.

The Markov Random Filed describes a set of random variables and their dependencies by a graph. Let $\mathcal{G} = (\mathcal{V}, \mathcal{D})$ denote a hypergraph which consists of a set $\mathcal{V}$ of nodes and a set $\mathcal{D}$ of cliques (*e.g.* see Figure 4.8 bottom).

- The node set $\mathcal{V}$ represents the point-based model, where each node corresponds to the variable of a control point (*i.e.* the position of the control point).
- The clique set $\mathcal{D}$ represents the local interactions between the nodes (random variables), where each clique consists of a subset of the nodes.

In particular, the clique set $\mathcal{D} = \mathcal{E} \cup \mathcal{F}$ is composed by two types of cliques in the graph:

- The set $\mathcal{E}$ represents the boundary cliques (*e.g.* see Figure 4.8 right). It defines the connectivity information of the control points in order to recover the boundary of the model. In 2D cases, each clique represents a line segment determined by two end points on the closed curve, while in 3D cases each clique represents a triangulated face of the mesh which are determined by three points.
- The set $\mathcal{F}$ represents the prior cliques (*e.g.* see Figure 4.8 left). It expresses the prior knowledge of the shape, where each clique represents the local interactions of three control points.

With respect to the graph $\mathcal{G}$, let $\mathbf{x}_{i \in \mathcal{V}}$ denote the random variable (*i.e.* the coordinates of point i) of each node, and $\mathbf{X} = (\mathbf{x}_i)_{i \in \mathcal{V}}$ indexed by $\mathcal{V}$ denotes all the node variables of

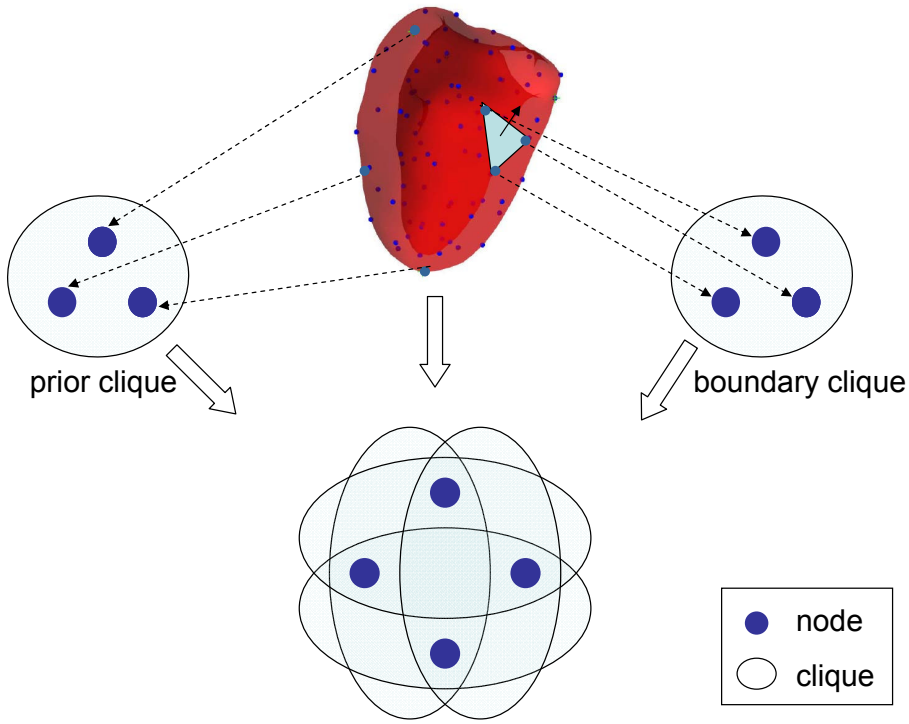


Figure 4.8: The relation between the object model (top) and the graphic model (bottom).

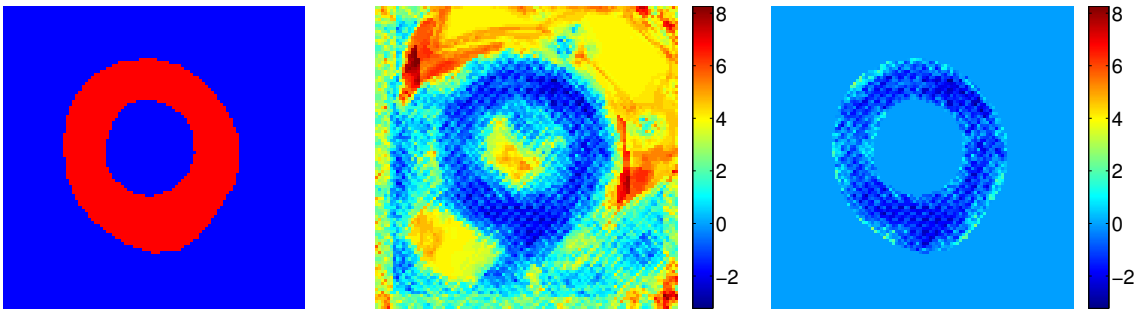
the MRF. Now we formulate the probabilistic framework of the segmentation problem in Eq.(4.4) as an MRF energy minimization.

$$\begin{aligned} \mathbf{X}^{\text{opt}} &= \arg\min_{\mathbf{X}} E(\mathbf{X}) \\ E(\mathbf{X}) &= E_{\text{data}}(\mathbf{X}) + E_{\text{prior}}(\mathbf{X}) \end{aligned} \tag{4.24}$$

where the MRF energy $E(\mathbf{X})$ integrates both visual support and shape prior constraints. Furthermore, the MRF energy $E(\mathbf{X})$ is a factorization of higher-order terms of the hypergraph, while the data energy $E_{\text{data}}(\mathbf{X})$ is defined on the boundary clique set $\mathcal{E}$ and the prior energy $E_{\text{prior}}(\mathbf{X})$ is defined on the prior clique set $\mathcal{F}$.

4.4.1 Regional Energy

As we stated before, the regional energy takes into account the homogeneity properties of the entire inner region of the shape model in an observed image $\mathbf{I}$. Given a shape configuration $\mathbf{X} = X$, the image domain Ω is partitioned into the object region $\Omega_{\text{obj}}(\mathbf{X} =$



(a) Image partition (b) Image likelihood (c) Likehood of Ω_{obj}

Figure 4.9: Regional energy.

X) and the background region $\Omega_{\text{bck}}(\mathbf{X} = X)$. In Figure 4.9, the object region and the background region are shownin red and blue respectively. We denote the function $f(\cdot)$ as the likelihood $-\log \frac{p_{\text{obj}}(\mathbf{I}(\cdot))}{p_{\text{bck}}(\mathbf{I}(\cdot))}$ (*e.g.* see Figure 4.9 (b)), while $p_{\text{obj}}, p_{\text{bck}}$ are the appearance distribution models of object and background. The regional energy (4.11) is computed as an integral of likelihood over the object region:

$$\begin{aligned} E_{\text{R}}^{(1)}(\mathbf{X}) &= \iint_{\Omega_{\text{obj}}(\mathbf{X})} f(x,y)dxdy \\ E_{\text{R}}^{(2)}(\mathbf{X}) &= \iiint_{\Omega_{\text{obj}}(\mathbf{X})} f(x,y,z)dxdydz \end{aligned} \tag{4.25}$$

where $E_{\text{R}}^{(1)}$ and $E_{\text{R}}^{(2)}$ denote the regional energies in 2D and 3D cases respectively. Given the coordinates of a point position in space *i.e.* (x, y) or (x, y, z), the likelihood function $f(x, y)$ or $f(x, y, z)$ gives a negative cost if the probability of this position being the object is larger than its probability being the background, otherwise it gives a positive cost. The regional energy can be interpreted in Figure 4.9: denoting A for the binary image of image partition (a), and B for the image likelihood (b), then the image (c) can be computed by $C(x, y) = A(x, y) \cdot B(x, y)$. As the result, the regional energy equals to the sum of the values of all the pixels in the image (c) *i.e.* $E_{\text{R}} = \sum_i C(i)$. Since the lower the value of the pixel in C, the higher probability of the pixel being the object, the regional energy encourages to include as many likely object pixel as possible.

However, all the variables of the model are required in the above definition in order to determine the region of interest, while in a Markov random field we assume the local dependencies of the variables (*i.e.* the variables are dependent only if they are included in the same clique.) As a consequence, the above equation of the regional energy can not be directly encoded in the MRF formulation. To deal with this difficulty, we propose the

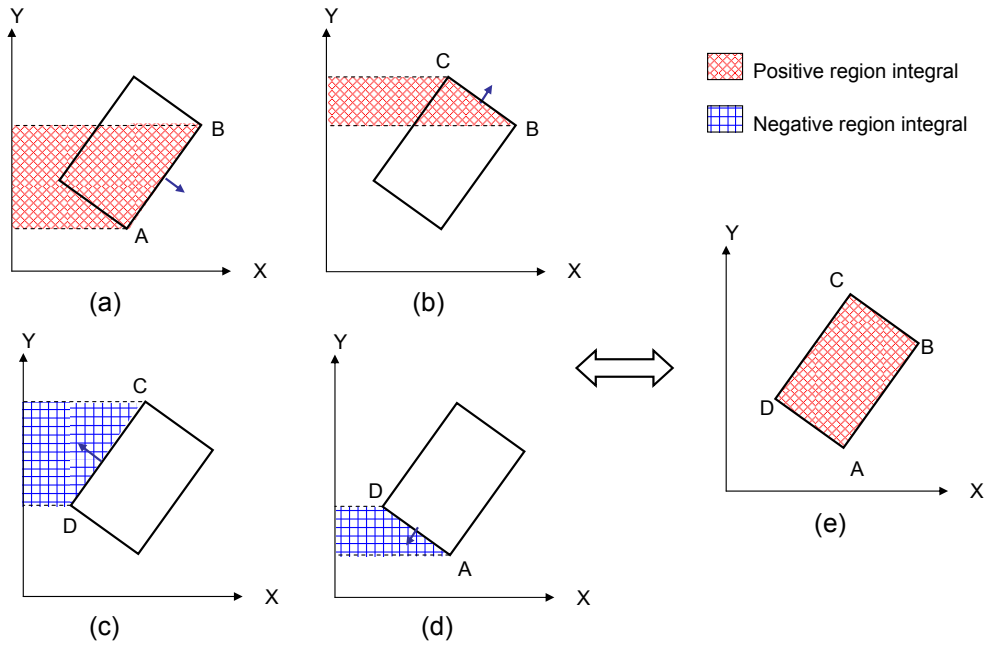


Figure 4.10: A 2D example using Divergence Theorem. (a-d) Line integrals around the closed curve. (e) Double integral over the bounded region.

exact factorization of the regional energy term (4.25) into higher order potentials in MRFs by using Divergence Theorem.

Factorization of 2D Cases

To understand the factorization method, let us first consider the situation in 2D cases. In mathematics, the 2D version of Divergence Theorem (equivalent to Green's Theorem) states the equivalence between a line integral around a simple closed curve and a double integral over the bounded plane region:

$$\iint_{\mathcal{D}} (\nabla \cdot \mathbf{F}) dA = \oint_{\mathcal{C}} (\mathbf{F} \cdot \mathbf{n}) ds \tag{4.26}$$

Let $\mathcal{C}$ be a positively oriented simple closed curve in a plane, and let $\mathcal{D}$ be the region bounded by $\mathcal{C}$, where the path of integration along $\mathcal{C}$ is counterclockwise. The two-dimensional vector field $\mathbf{F} = (F_x, F_y)$ is defined on an open region containing $\mathcal{D}$, where $\nabla \cdot \mathbf{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y}$ is the divergence on $\mathbf{F}$, and $\mathbf{n}$ is the outward-pointing unit normal vector

on the boundary. Figure 4.10 illustrates the 2D Divergence Theorem with an example of a quadrangle region $ABCD$. The line integrals along each segment of the curve are shown in (a-d), where the outward-pointing normal $\mathbf{n}$ is shown with a blue arrow. Each component of the line integral equals to a double integral over the region shown in red or blue, where red represents the equivalent integral over that region is positive since $\mathbf{F} \cdot \mathbf{n} > 0$ and blue represents the integral over the region is negative since $\mathbf{F} \cdot \mathbf{n} < 0$. The sum of the line integrals along the segments $\{AB, BC, CD, DA\}$ equals the double integral over the bounded region $ABCD$ shown in (e).

Now we associate the divergence theorem with the computation of regional energy $E_{\text{R}}^{(1)}$ (4.25). In this context, the closed curve $\mathcal{C}$ is the boundary of the shape model $B(\mathbf{X})$, and the bounded region is the inner region of the model $\Omega_{\text{obj}}(\mathbf{X})$. Let $f(x, y)$ denote the integral function over the region $\Omega_{\text{obj}}(\mathbf{X})$ on the left side of the equation:

$$E_{\text{R}}^{(1)} = \iint_{\Omega_{\text{obj}}(\mathbf{X})} f(x, y)dxdy = \oint_{B(\mathbf{X})} (\mathbf{F} \cdot \mathbf{n})ds \tag{4.27}$$

Let us choose $F_y = 0$, according to the divergence theorem, the function $f(x, y) = \frac{\partial F_x}{\partial x}$ is the derivative of the function F_x with respect to x, thus we can compute the line integral function F_x as follows:

$$F_x(x, y) = \int_0^x f(t, y)dt = \int_0^x -\log \frac{p_{\text{obj}}(\mathbf{I}(t, y))}{p_{\text{bck}}(\mathbf{I}(t, y))}dt \tag{4.28}$$

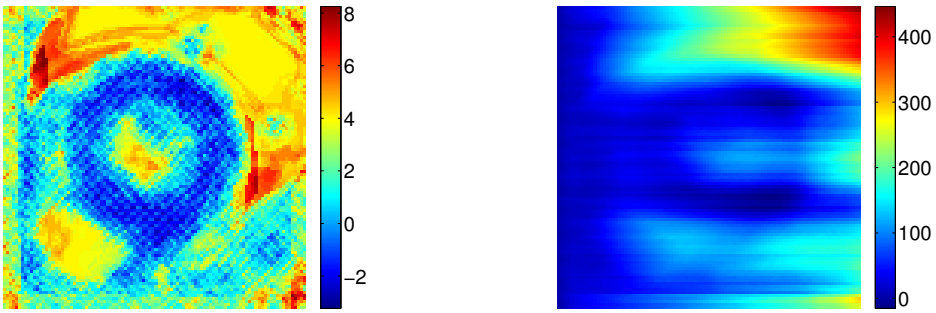
where t denotes the variable. We interpret the physical meaning of $F_x(x, y)$ as follows, if we consider the likelihood function $f(x, y)$ over the image domain as an image (*e.g.* see Figure 4.11 (a)) by assuming the probabilities are all equal outside the image, then the function $F_x(x, y)$ over the image domain corresponds to the related integral image with respect to the x axis (*e.g.* see Figure 4.11 (b)). Moreover, we denote $G_{\text{R}}^{(1)}(x, y)$ as the integral function along the curve:

$$G_{\text{R}}^{(1)}(x, y) = \mathbf{F} \cdot \mathbf{n} = F_x(x, y)n_x(x, y) \tag{4.29}$$

where n_x is the component of the outward pointing unit normal $n = (n_x, n_y)$ of the boundary $B(\mathbf{X})$ corresponding to the x axis.

Furthermore, the line integral around the closed curve in Eq.(4.27) can be factorized into the integrals along the segments of the curve which are determined by two end points. Thus, the regional energy in 2D cases can be encoded with pairwise potentials computed by line integrals as follows:

$$E_{\text{R}}^{(1)}(\mathbf{X}) = \sum_{(i,j)\in\mathcal{E}} \int_{\mathbf{x}_i\mathbf{x}_j} G_{\text{R}}^{(1)}(x(s), y(s))ds \tag{4.30}$$



(a) (b)

Figure 4.11: Computation of regional energy using Divergence theorem. (a) Image likelihood f. (b) Function F_x.

where the boundary clique set $\mathcal{E}$ consists of pairs of the node variables in the graph $\mathcal{G}$. In particular, each pair $(i,j) \in \mathcal{E}$ represents a segment of two control points $\{\mathbf{x}_i, \mathbf{x}_j\}$ connected with a direction such that all the segments compose a closed curve in counter-clockwise. The unit normal $\mathbf{n}$ of the directed segment can be computed as: $\mathbf{n} = \mathbf{i}\frac{dy}{ds} - \mathbf{j}\frac{dx}{ds}$. Thus the regional energy can be written as:

$$E_{\mathrm{R}}^{(1)}(\mathbf{X}) = \sum_{(i,j)\in\mathcal{E}} \int_{\mathbf{x}_i}^{\mathbf{x}_j} F_x(x(y), y)dy \tag{4.31}$$

Factorization of 3D Cases

Now we extend the method to 3D cases. In 3D space, the divergence theorem states that the outward flux of a vector field through a closed surface is equal to the volume integral of the divergence over the region inside the surface.

$$\iiint_V (\nabla \cdot \mathbf{F})dV = \oiint_S (\mathbf{F} \cdot \mathbf{n})ds \tag{4.32}$$

Suppose V is a 3D volume which is compact and has piecewise smooth boundary S (also indicated with $\partial V = S$), where $\mathbf{F}$ is a continuously differentiable vector field defined on a neighborhood of V and $\mathbf{n}$ is the outward pointing unit normal field of the boundary S. In other words, the left side of the equation is a volume integral over the volume V and the right side is the surface integral over the boundary S of the volume. The boundary S has to satisfy the conditions: (1) it is a closed surface and (2) it is the generally the boundary of the volume V oriented by outward-pointing normals.

Next, we associate the 3D divergence theorem with the computation of the regional energy $E_{\mathrm{R}}^{(2)}$ (4.25). The volume V in 3D space corresponds to the inner region of the model

$\Omega_{\text{obj}}(\mathbf{X})$ and the closed boundary S corresponds to the surface boundary of the model $B(\mathbf{X})$. Let the likelihood $f(x,y,z) = \nabla \cdot \mathbf{F}$ denote the divergence of the differentiable vector filed $\mathbf{F} = (F_x, F_y, F_z)$, the regional energy which involves volume integral can be transformed into surface integral:

$$E_{\text{R}}^{(2)}(\mathbf{X}) = \iiint\limits_{\Omega_{\text{obj}}(\mathbf{X})} f(x,y,z)dxdydz = \oiint\limits_{B(\mathbf{X})} (\mathbf{F} \cdot \mathbf{n})ds \tag{4.33}$$

Let us choose $F_x = F_y = 0$, we have the relation between the scale-valued function $f(x,y,z)$ and $F_z(x,y,z)$:

$$F_z(x,y,z) = \int_0^z f(x,y,t)dt = \int_0^z -\log \frac{p_{\text{obj}}(\mathbf{I}(x,y,t))}{p_{\text{bck}}(\mathbf{I}(x,y,t))}dt \tag{4.34}$$

where t denotes the variable. We interpret the physical meaning of $F_z(x,y,z)$ as follows: if we consider the likelihood function $f(x,y,z)$ over the image domain as an image by assuming the probabilities are all equal outside the image, then the function $F_z(x,y,z)$ over the image domain corresponds to the integral image with respect to the z axis. Moreover, we use $G_{\text{R}}^{(2)}(x,y,z)$ to denote the surface integral function in Eq.(4.33).

$$G_{\text{R}}^{(2)}(x,y,z) = \mathbf{F} \cdot \mathbf{n} = F_z(x,y,z)n_z(x,y,z) \tag{4.35}$$

where n_z is the component of the outward pointing unit normal $\mathbf{n} = (n_x, n_y, n_z)$ of the boundary $B(\mathbf{X})$ corresponding to the z axis.

Furthermore, since the boundary surface S is represented by a triangulated mesh $B(\mathbf{X})$ in our case, then the surface integral (4.33) over the closed surface of the volume can be factorized into the integrals over each triangle face of the mesh:

$$E_{\text{R}}^{(2)}(\mathbf{X}) = \sum_{c \in \mathcal{E}} \iint\limits_{\mathbf{x}_c} G_{\text{R}}^{(2)}(x(s), y(s), z(s))ds \tag{4.36}$$

where the boundary clique set $\mathcal{E}$ consists of triplets of control points which compose the triangulated mesh. In particular, a triplet clique $c = (i,j,k) \in \mathcal{E}$ represents a triangulated face $\mathbf{x}_c = \{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k\}$ oriented by the outward-pointing normal $\mathbf{n}$ which can be computed by the cross product of two vectors $\mathbf{n} = \mathbf{x}_i\mathbf{x}_j \times \mathbf{x}_j\mathbf{x}_k$.

4.4.2 Boundary Energy

Boundary-based energy uses the discontinuity properties between different regions. It is defined as the integral of the appearance discontinuities along the model boundary $B(\mathbf{X})$

in Eq.(4.5). Since the model boundary is composed by a set of segments of a closed curve in 2D cases or a set of triangulated faces of a mesh in 3D cases, the integral along the model boundary can be decomposed into higher-order terms:

$$\begin{aligned} E_{\mathrm{B}}^{(1)}(\mathbf{X}) &= \sum_{c\in\mathcal{E}} \int_{\mathbf{x}_c} G_{\mathrm{B}}^{(1)}(x(s), y(s))ds \\ E_{\mathrm{B}}^{(2)}(\mathbf{X}) &= \sum_{c\in\mathcal{E}} \iint_{\mathbf{x}_c} G_{\mathrm{B}}^{(2)}(x(s), y(s), z(s))ds \end{aligned} \tag{4.37}$$

where the energy $E_{\mathrm{B}}^{(1)}$ of 2D cases are expressed by pairwise terms and the energy $E_{\mathrm{B}}^{(2)}$ of 3D cases are expressed by second-order terms (using triplet cliques). The discontinuity function G_{B} represents a distance map to the edges over the image domain. In particular, the discontinuity function G_{B} labels each position in the image space with a non-negative real value as the distance of this position to its nearest edge.

4.4.3 Shape Prior Energy

The shape prior energy $E_{\text{prior}}(X)$ imposes the geometric constraints of the model in order to produce a valid shape. Based on our sparse graphic shape prior which is modeled by local interactions, the prior energy can be encoded using higher order potentials.

$$E_{\text{prior}}(\mathbf{X}) = \sum_{c\in\mathcal{F}} -w_c \cdot \log p_c(\alpha_c(\mathbf{x}_c), \beta_c(\mathbf{x}_c)) \tag{4.38}$$

where $\mathcal{F}$ consists of a set of triplet cliques. Each clique $c \in \mathcal{F}$ is associated with a weight w_c and the probability density p_c of two inner angles from learning.

4.4.4 Higher-order MRF Inference

To this end, the total MRF energy (4.4) is a summary of the data energy $E_{\text{data}}(\mathbf{X})$ and the prior energy $E_{\text{prior}}(\mathbf{X})$:

$$E(\mathbf{X}) = \sum_{c\in\mathcal{E}} \psi(\mathbf{x}_c) + \sum_{c\in\mathcal{F}} \phi(\mathbf{x}_c) \tag{4.39}$$

where ψ and ϕ encode respectively the data potential and the prior potential:

$$\begin{cases} \psi^{(1)}(\mathbf{x}_c) = \int_{\mathbf{x}_c} \left(\lambda_1 \cdot G_{\mathrm{R}}^{(1)}(s) + \lambda_2 \cdot G_{\mathrm{B}}^{(1)}(s)\right) ds \\ \psi^{(2)}(\mathbf{x}_c) = \iint_{\mathbf{x}_c} \left(\lambda_1 \cdot G_{\mathrm{R}}^{(2)}(s) + \lambda_2 \cdot G_{\mathrm{B}}^{(2)}(s)\right) ds \\ \phi(\mathbf{x}_c) = -w_c \cdot \log p_c(\alpha_c(\mathbf{x}_c), \beta_c(\mathbf{x}_c)) \end{cases} \tag{4.40}$$

Note that $\psi^{(1)}$ and $\psi^{(2)}$ denote the data potentials of 2D and 3D cases respectively, $\lambda_1 > 0$ and $\lambda_2 > 0$ being two weight coefficients.

Given the MRF energy in Eq.(4.39) and Eq.(4.40), we adopt a dual-decomposition optimization framework [Komodakis 2007a] to perform the Maximum a Posteriori (MAP) inference for the proposed higher-order MRF. The dual-decomposition strategy is considered to be the state-of-the-art for MAP-MRF inference, in particular when dealing with higher-order MRFs. Based on such a framework, we decompose the original problem (which is difficult to solve directly) into a set of factor trees [Wang 2010] which can be solved very efficiently within polynomial time using max-product belief propagation algorithm [Bishop 2006]. A projected subgradient method [Komodakis 2007a] is employed to combine the solutions of the sub-problems in order to obtain the solution of the original problem.

4.5 Experimental Validation

We validate the proposed method in both 2D and 3D segmentation. Manual segmentations on the database are available and are considered as ground truth for both learning and validation purposes. An iterative scheme is employed to search for the optimal model instance in the test image. Given an initialized model, the label space of each node is composed by a set of displacements of the current position. The model is updated by the optimal displacements in each iteration, while the displacement set is adapted to a coarse to fine setting during the model deformation. The experiments (programmed in C++) were run on a 2.8GHz, Quad Core, 12GB RAM computer.

4.5.1 2D Hand Segmentation

Our 2D hand dataset consists of 40 right hand examples with different poses and movements between the fingers. The shape model consists of 23 control points, and a number of 100 cliques with the largest parameters to represent the shape prior.

Some segmentation results of our knowledge-based method are shown in Fig.4.15, where the red solid contours represent our results and the yellow dashed contours represent the initializations. As can be seen, our results are robust to the noise, partial occlusions and complicated background. For example, in the second row where the fingers are partially self-occluded, our method shows the ability to deal with the shapes which have not been seen during training. In the third row, the same images from the first row are artificially added with Gaussian noise and black obstructions, while we deal with these cases with a larger weight of the prior energy, which is also the reason why a part of sleeve is mis-

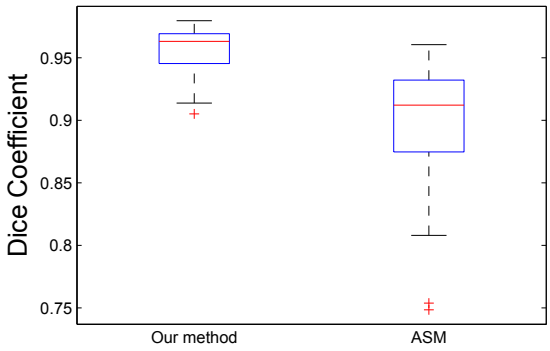


Figure 4.12: Dice coefficients of 2D hand segmentation.

labeled as the hand in the second image. The fourth row shows our results on a set of video images with complicated background.

For both quantitative and comparison purposes, we compare our method with Active Shape Model (ASM) using dice coefficient in Fig.4.12. Dice coefficient is a statistical measurement used for comparing the similarity between two samples and it is formulated as:

$$s = \frac{2\,|A \cap B|}{|A| + |A|} \tag{4.41}$$

where A and B are two segmentation solutions in our case. $|A|$ is the number of the object pixels in segmentation A, and $|A \cap B|$ is the number of the common object pixels in both segmentations. The value of dice coefficient s is in $[0, 1]$ range, and the higher value indicates that the two solutions are more similar. We compute the dice coefficient of our segmentation result and the ground truth, and the dice coefficient of ASM result and the ground truth. In each box of Fig.4.12, the central mark in red is the median, the edges of the box are the 25th and 75th percentiles. It verified that our method is more similar to the ground truth than ASM result. Moreover, benefit from the sparse graphic shape prior, our segmentation takes 20 seconds per image while the one using complete graph takes more than 4 minutes.

4.5.2 2D Left Ventricle Segmentation

We validate our method on a dataset which consists of 60 tagged cardiac MR images. Standard of reference was available, consisting of annotations of epicardium and endocardium boundaries provided by experts. These MR images were acquired by a 3-T Siemens MR imaging system equipped with a high-performance gradient system (maximum amplitude:

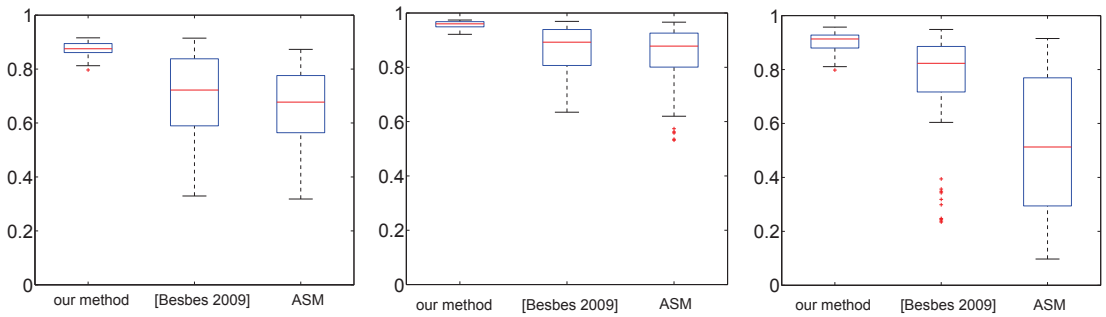
40 mT/m; minimum rise: slew rate 200 mT.m^{-1}/s) using a 32-channel phased-array cardiac coil. Images were acquired in the short axis plane at basal, mild and apical ventricular levels. An ECG-triggered segmented k-space fast gradient echo sequence with spatial modulation of magnetization was performed with the following parameters: grid tag spacing: 8 mm; echo time=2.54 ms; repetition time=48 ms; number of frames: 20-25 (depending of heart rate); pixel size:1.8 $\times$ 1.4 $\times$ 7 mm; bandwidth 446 Hz/pixel; flip angle: 10^o; acquisition time: 19 seconds (during one breathhold).

We performed a leave-one-out cross validation on the whole dataset. The computational time of segmenting an image is 0.781 second on the average. For all the images in the dataset, we used the same parameters and the same initialization. Segmentation results on test images from different sequences of different patients are presented in Fig.4.16, which shows that our shape model can represent well the contraction of myocardium during the cardiac cycle and can deal with different scales of the myocardium boundaries. Furthermore, Fig.4.16 (b) shows the results obtained with different initializations (shown in green contours) with respect to the location and scale on the same test image. The consistent results demonstrate the robustness of our method with respect to the initialization.

For both quantitative evaluation and comparison purposes, we present in Fig.4.13 the distributions of the Dice coefficients of the segmentation results of the endocardium (region bounded by the inner contour), the epicardium (region bounded by outer contour) and the myocardium (region bounded by both contours), respectively. Each sub-figure of Fig.4.13 contains three boxes which present the Dice coefficients obtained by our method, the method of [Besbes 2009] and standard ASM method, respectively. Note that a higher Dice coefficient implies a better segmentation performance. Therefore, the obtained Dice coefficients demonstrate that our segmentation approach performs significantly better than the other two methods. In particular, the better performance with respect to [Besbes 2009] demonstrates the power of the exact factorization of the regional data likelihood.

4.5.3 3D Left Ventricle Segmentation

A dataset of 20 3D CT cardiac images is used to validate the proposed method in 3D segmentation application. The point-based model consists of 88 control points both on the myocardium surface as well as the atrium surface. The coarse triangulated mesh consists of 172 triangle faces. A number of 1000 triplet cliques are selected from the MRF learning to encode the shape prior. Regarding the image support, a feature vector is used for each voxel instead of intensity values. The feature vector consists of patches of intensities and Gabor features. Then the learning is performed using an Adaboost classifier for the object and the background, and we apply the classifier responses to obtain a likelihood image for the test image.



(a) Endocardium (b) Epicardium (c) Myocardial

Figure 4.13: Dice coefficients of 2D left ventricle segmentation.

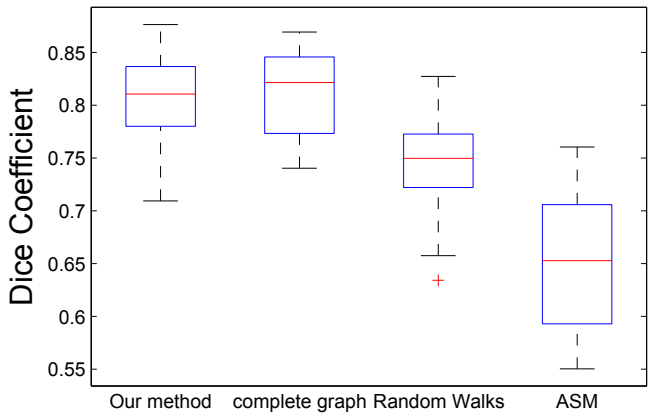


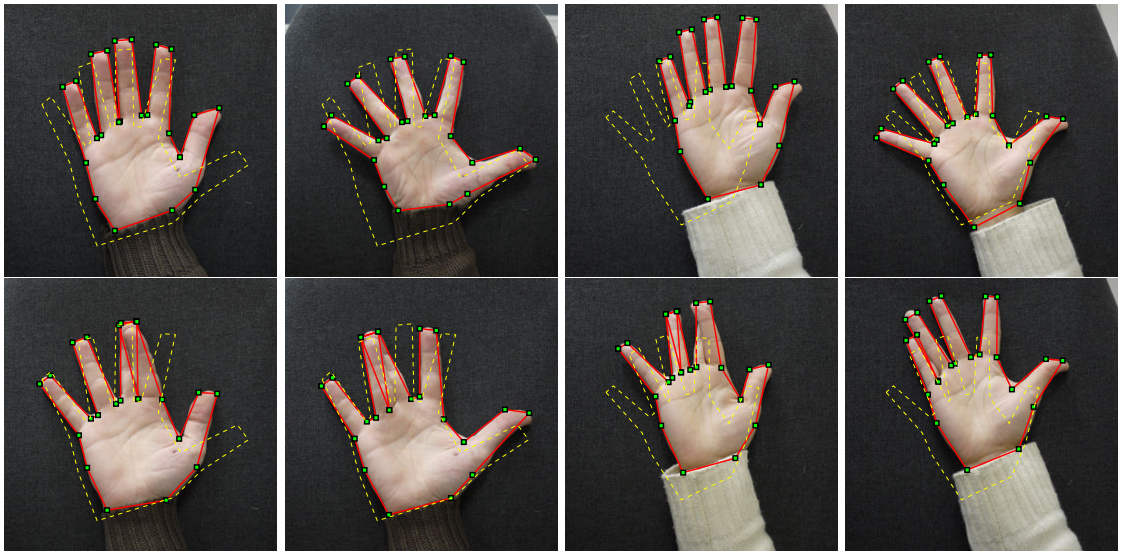
Figure 4.14: Dice coefficients of 3D left ventricle segmentation.

We perform a leave-one-out cross-validation on the dataset. Some results are shown in Fig.4.17 where the yellow contours represent our results, while the green contours represent the results of ASM models. As can be observed, our model exhibits better accuracy on the boundary (the first two columns) and robustness to the papillary muscles in the blood pool (the last column). Fig.4.14 presents the Dice coefficients obtained by our method with sparse graph, our method with complete graph [Xiang 2012], the Random Walks algorithm [Grady 2006] and ASM [Cootes 1995]. Although the performance of our previous method with complete graph is competitive to the one with sparse graph, it introduces a higher computational complexity (linear to the number of cliques in the graph) and takes hours to segment one volume. On the contrary, our recent method is more efficient with decreased computation complexity in both energy computation and optimization process,

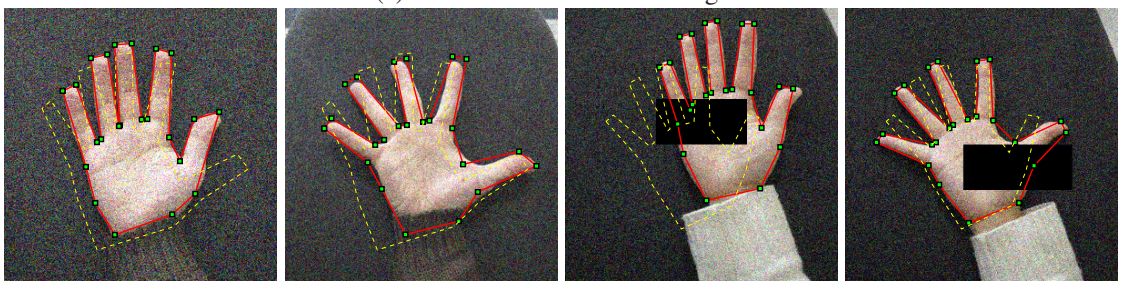
and it takes about 15 minutes per volume.

4.6 Conclusion

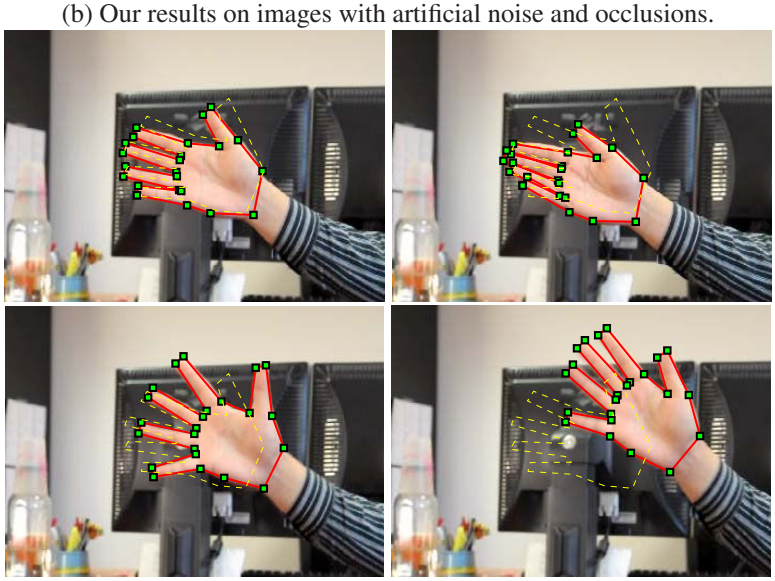
In this chapter we have studied the problem of knowledge-based object segmentation. We develop a global approach to jointly encode the regional statistics, boundary support, as well as prior knowledges within a probabilistic framework. The pose-invariant priors are encoded by second-order MRF potentials. The regional statistics is exactly factorized into pairwise or second-order terms using Divergence theorem. The proposed segmentation method is robust to noise, partial object occlusions and initializations. It is efficient and does not suffer from bad local minima issues using developed MRF optimization algorithms.



(a) Our results on standard images.

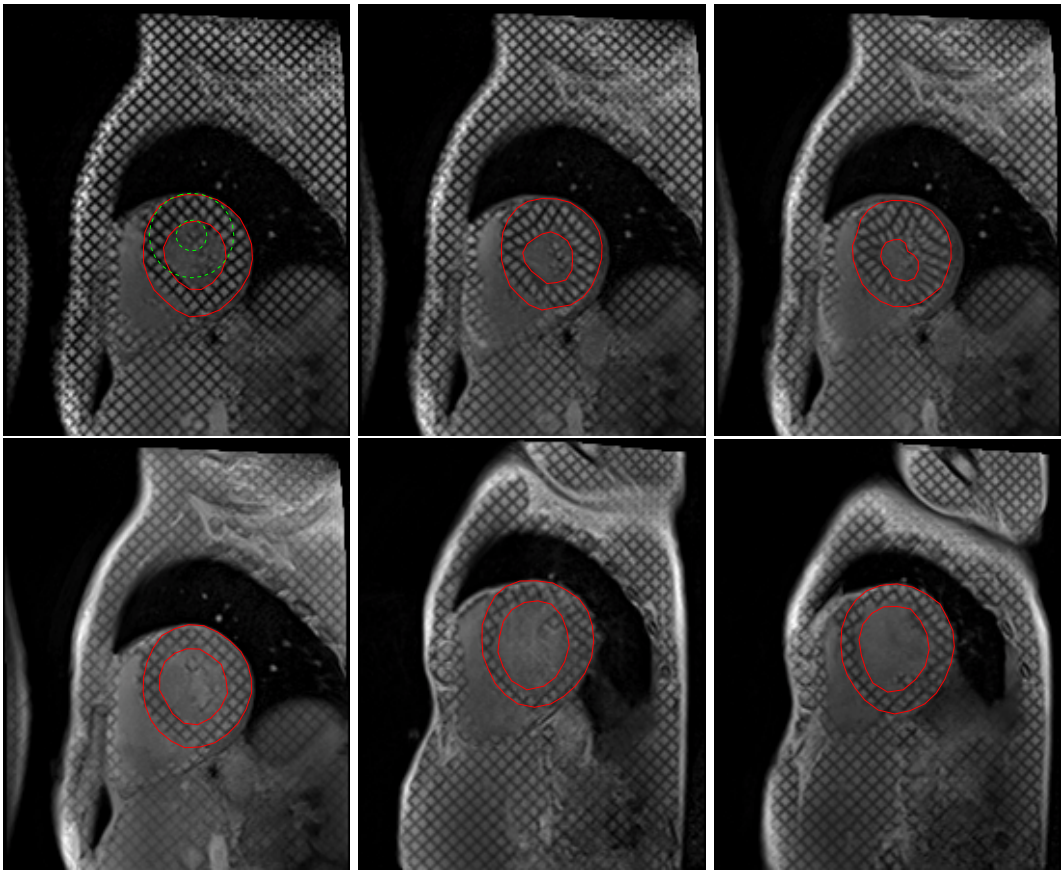


(b) Our results on images with artificial noise and occlusions.

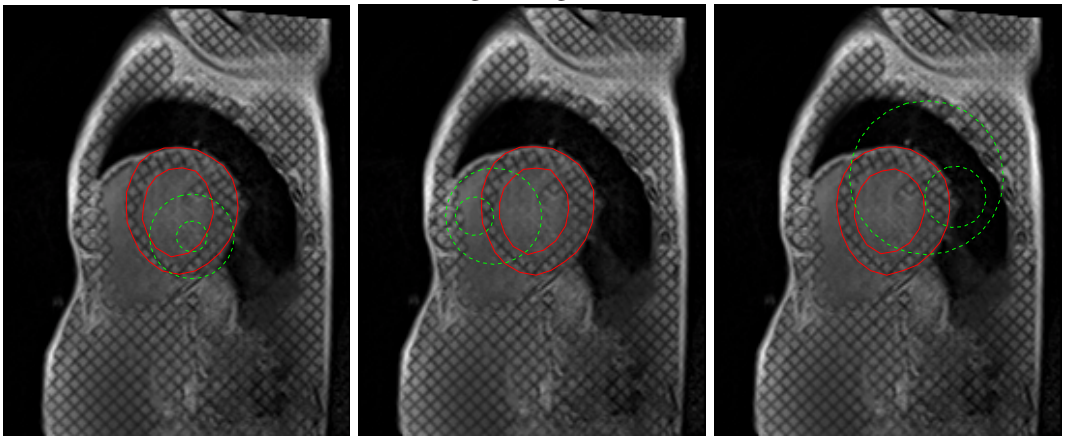


(c) Our results on video images with cluttered background.

Figure 4.15: 2D hand segmentation results.

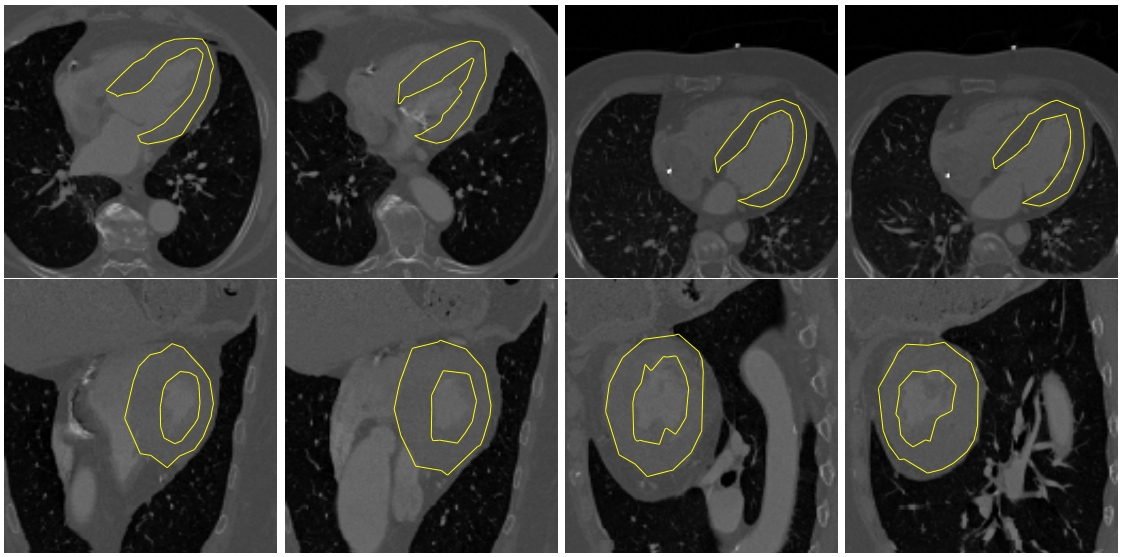


(a) Different test images using the same initialization.

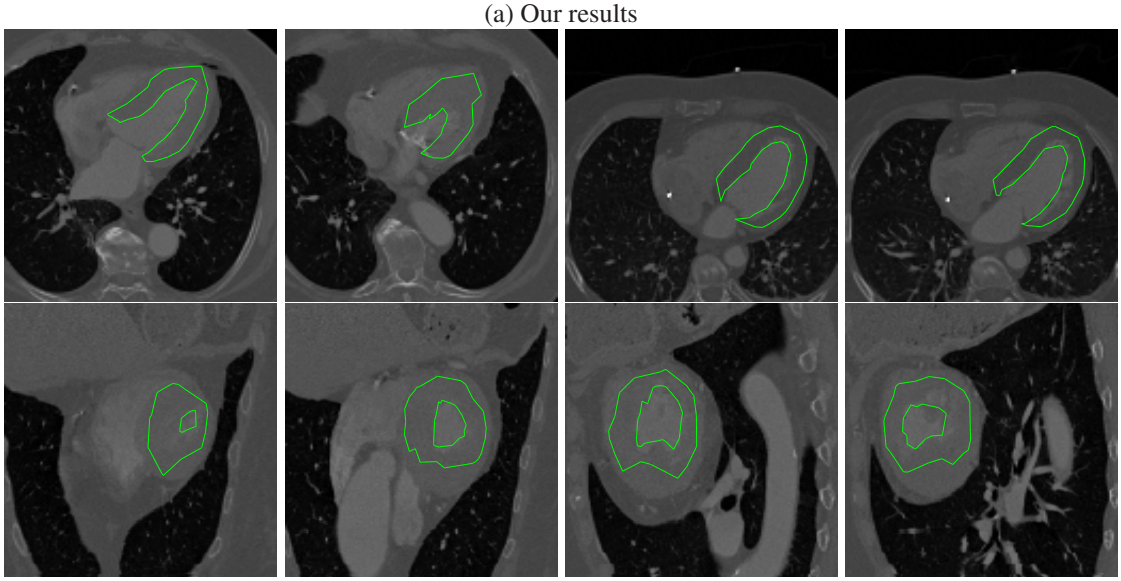


(b) Same test image with different initializations.

Figure 4.16: 2D left ventricle segmentation results.



(a) Our results



(b) ASM results

Figure 4.17: 3D left ventricle segmentation results on cardiac CT volumes.

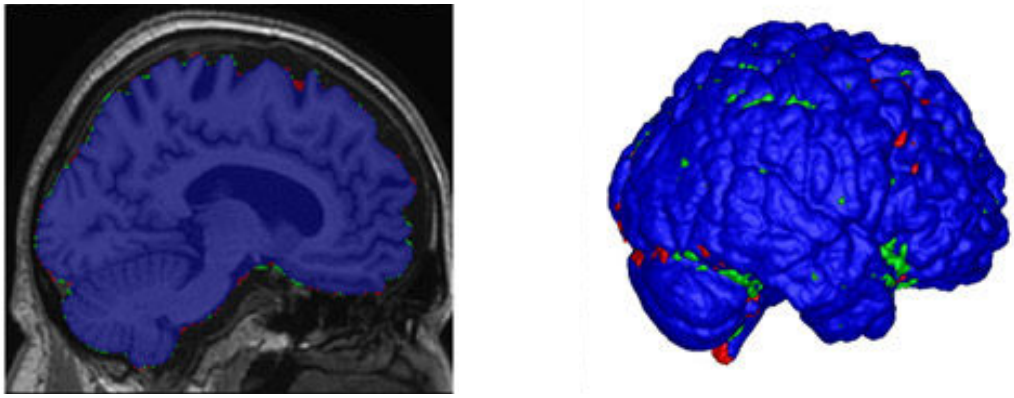
Chapter 5

Joint Model-Pixel Segmentation

In the previous chapter, we have proposed a top-down approach for class-specific segmentation. The model-based segmentation is formulated as estimating the object boundary model in the observed image, combining the prior knowledge regarding the object shape as well as the image cues. In this chapter, we are going to integrate both top-down and bottom-up approaches in a unified framework towards a more refined segmentation, estimating the pixel-level labeling and the model localization simultaneously.

5.1 Introduction

The earlier segmentation approaches can be generally classified into two types: (1) Bottom-up ones which label each image pixel as object or background using low-level information; (2) Top-down ones which delineate the object boundary based on high-level information of the object class. The latter ones have gained increasing popularity since they are able to incorporate prior knowledge and thus are robust to low-level variations. However, the segmentation performance of the top-down approaches is highly dependent on the choice of the model representation which defines the boundary of the object of interest. For instance, the point-based model is widely chosen for the convenience to study the statistics of the shape, but due to its discrete representation, it produces the piecewise linear boundary of the object and thus generates segmentation errors in the local boundary area. These coarse segmentation and missing details can not be accepted in many applications (*e.g.* medical image applications). In Fig.5.1, we show an example of brain extraction where the exact delineation of the object boundary is necessary. In the segmentation, blue voxels are correctly segmented compared to the gold standard, while green voxels are false positives and red voxels are false negatives. Obviously, a top-down model is not capable to capture all the details on the brain surface, otherwise the complexity of the model is significantly



(a) Segmentation on a sagittal slice (b) 3D rendering

Figure 5.1: Brain extraction [Eskildsen 2012].

increasing with respect to the boundary details. In fact, this issue can be refined by bottom-up approaches using low-level cues. Based on the above, a natural direction to improve the existing approaches is to combine both top-down and bottom-up cues in a principled manner for class-specific segmentation problem. In Fig.5.2 given the input image (a), we show the examples of bottom-up segmentation in (b) and top-down segmentation in (c), where bottom-up results can detect salient image discontinuities and the top-down results can capture the spatial relationships between the object parts.

Over the past decade, many efforts have been made in combining top-down and bottom-up segmentation. Kumar *et al.* propose an OBJ CUT method [Kumar 2005] which combines an Markov Random Field (representing bottom-up information) and the Layered Pictorial Structures (LPS) [Kumar 2004] (representing top-down information) for segmentation, while the former biases the segmentation to follow image discontinuities and the latter provides the prior knowledge of the object shape. An EM framework is used to solve this combined method: (1) in E step, LPS model is matched to the given image and a number of samples each corresponding to a probable pose of the object are obtained; (2) in M step, given the model samples, the segmentation can be obtained using a single graph cut.

Bray *et al.* propose a POSE CUT method [Bray 2006] for combining object segmentation and pose estimation of a human body simultaneously. Similar to the OBJ CUT method, they also include the shape prior in a Markov Random Field (MRF) for object segmentation, while instead of learning exemplars of the object as in the former method, they use a simple articulated stick-man model as the pose-specific shape prior. Given an image, the optimization of the pose-specific MRF with respect to segmentation measures the quality of a pose, then the pose inference is formulated as minimizing this cost function

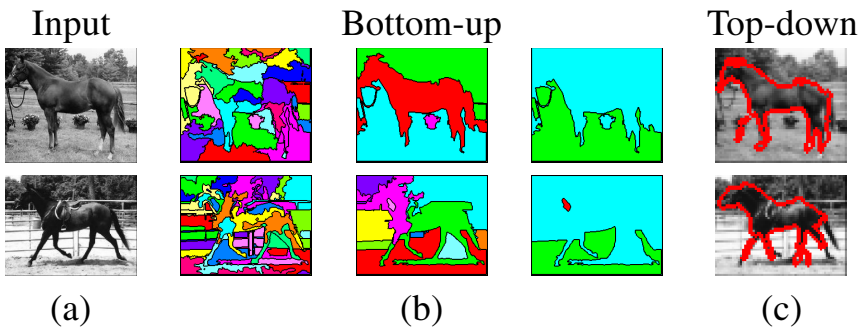


Figure 5.2: Bottom-up and top-down segmentation [Borenstein 2008]. (a) Input image. (b) Bottom-up segmentation at three scale. (c) Top-down segmentation.

over all pose parameters using dynamic graph cuts.

Levin and Weiss propose a Conditional Random Fields (CRF) framework [Levin 2006] to learn combined low-level and high-level cues from segmented images. The simultaneous learning procedure yields a novel fragment selection algorithm, which allows them to efficiently learn models with a smaller number of fragments, whereas pure top-down algorithms often requires hundreds of fragments to represent the top-down model. Then given a new image, a combination of a low-level term and a local class-dependent term regarding the learned fragments is used to define the CRF energy towards image segmentation.

However, these above methods combine top-down with bottom-up processing in a strictly feed-forward manner to produce segmentation. Borenstein and Ullman propose an intertwined scheme for segmentation and recognition [Borenstein 2008]. The top-down part [Borenstein 2002, Borenstein 2004] learns a bank of fragments and their automatic labeling to represent a class. Given a novel image, the stored fragments are first used to recognize the object and to create a complete cover of the object shape, then the resulting top-down segmentation is integrated with multi-scale hierarchical bottom-up segmentation to better delineate the object boundaries.

Wang *et al.* propose a unified graphical-model framework for simultaneous segmentation, ordering and multi-object tracking [Wang 2009]. A single pairwise Markov Random Field (MRF) is used to jointly estimate all variables of interest with respect to both pixel-level (pixel class label and pixel depth) and object-level (model parameters and object depth), while interaction between these variables are expressed as cost terms in the MRF. The contribution of this approach is its single-shot optimization MRF framework for joint segmentation, depth ordering and tracking with occlusion handling. However, they use a simple rectangle representation to model the object and no prior knowledge about the object shape is included, thus the resulting segmentation is less accurate especially when

dealing with the objects with large deformations.

Packer *et al.* combine a contour-based LOOPS model [Heitz 2009] (top-down module) with the CRF-based segmentation (bottom-up module) to form a coherent energy function over both model parameters and pixel labels [Packer 2010]. The energy function includes the terms for each separate task with an interaction term that encourages the contour and pixel-level segmentation to agree. Specifically, they introduce landmark-segment masks which are learned to capture outline detail of each part of the object, in order to connect the model landmarks and the local pixel labels. An efficient method is proposed for joint inference which can avoid local minima found in each task separately.

There are two main limitations among the existing methods: (1) The combined problem is addressed within an alternating minimization approach where no guarantees on the optimality properties of the obtained solution could be satisfied. (2) The object model is either simple or does not include statistic priors on the global shape. The method in [Heitz 2009] solves the both problems, but they only consider a subset of the image pixels in the unified energy, thus a post-processing step is necessary to label all the pixels.

In this chapter, we propose a novel framework for image segmentation through a unified model-based and pixel-driven integrated graphical model, similar to [Heitz 2009]. Prior knowledge of the object shape is expressed through the deformation of a discrete model that consists of decomposing the shape of interest into a set of higher order cliques (triplets). Such decomposition allows the introduction of region-driven image statistics as well as pose-invariant (i.e. translation, rotation and scale) constraints whose accumulation introduces global deformation constraints on the model. Regional triangles are associated with pixels labeling which aims to create consistency between the model and the image space. The proposed pose-invariant framework simultaneously solves the problem in both model space and image space. It is achieved by the definition of an objective function aiming to: (i) assign labels to image pixels in order to maximize the image likelihood [Boykov 2006], (ii) deform a point-based model in order to maximize the geometric likelihood of the model as well as the model-to-image likelihood (our model-based segmentation which is introduced in the previous chapter), (iii) impose consistency between the two label spaces. The resulting higher order graphical model formulation is solved by using a state of the art message passing algorithm [Kolmogorov 2006]. Promising results on a challenging clinical setting demonstrate the potentials of our method.

The remainder of the chapter proceeds as follows. We first present the probabilistic framework in Section 5.2. Based on the shape representation in Section 5.3, the Markov Random Field formulation is defined in Section 5.4. Experimental validation is shown in Section 5.5 while Section 5.6 concludes the chapter.

5.2 Probabilistic Framework

In this section, we propose a framework to combine both model-based and pixel-based segmentation. The aim is to simultaneously deform the shape model in an observed image and label the image pixels to object/background using an interconnected graphical model.

Model-based segmentation aims to partition the image domain by searching for an optimal model configuration to best compromise between data-attraction and shape-fitness with the prior. It can be formulated as a maximization of the posterior probability (MAP) in a probabilistic framework. Given an image $\mathbf{I}$, let us denote $\mathbf{X}$ for the model variables and $\mathrm{dom}(\mathbf{X})$ for the model space, then the model variables can be optimized by:

$$\mathbf{X}^{\mathrm{opt}} = \arg \max_{\mathbf{X} \in \mathrm{dom}(\mathbf{X})} p(\mathbf{X}|\mathbf{I}) \tag{5.1}$$

Using Bayes' rule, the posterior distribution $p(\mathbf{X}|\mathbf{I})$ is proportional to the product of $p(\mathbf{I}|\mathbf{X})$ and $p(\mathbf{X})$ since $p(\mathbf{I})$ is a normalizing constant.

$$p(\mathbf{X}|\mathbf{I}) = \frac{p(\mathbf{X}, \mathbf{I})}{p(\mathbf{I})} \propto p(\mathbf{X}, \mathbf{I}) = p(\mathbf{I}|\mathbf{X}) \cdot p(\mathbf{X}) \tag{5.2}$$

The conditional distribution $p(\mathbf{I}|\mathbf{X})$ encodes the data likelihood of an observing image $\mathbf{I}$ given a particular model configuration $\mathbf{X}$, and the probability distribution $p(\mathbf{X})$ encodes the prior knowledge of the model $\mathbf{X}$ regarding the object shape. A model-based segmentation approach has been discussed with details in the previous chapter.

Pixel-based segmentation aims to group the pixels which are consistent in the appearance (*e.g.* intensity, color or texture) by assigning a label to each pixel in the image, so that the pixels which belong to the same object should be assigned with the same label. Let us denote $\mathbf{Y}$ as a vector of which each component represents a label variable of a pixel, and $\mathrm{dom}(\mathbf{Y})$ is denoted as the image labeling space. The pixel-based segmentation can be formulated as an MAP estimation of labeling $\mathbf{Y}$ over the labeling space $\mathrm{dom}(\mathbf{Y})$:

$$\mathbf{Y}^{\mathrm{opt}} = \arg \max_{\mathbf{Y} \in \mathrm{dom}(\mathbf{Y})} p(\mathbf{Y}|\mathbf{I}) \tag{5.3}$$

Similarly, the posterior distribution $p(\mathbf{Y}|\mathbf{I})$ can be expressed by two terms $p(\mathbf{I}|\mathbf{Y})$ and $p(\mathbf{Y})$ using Bayes' rule.

$$p(\mathbf{Y}|\mathbf{I}) \propto p(\mathbf{Y}, \mathbf{I}) = p(\mathbf{I}|\mathbf{Y}) \cdot p(\mathbf{Y}) \tag{5.4}$$

The conditional distribution $p(\mathbf{I}|\mathbf{Y})$ encodes the data likelihood of the image $\mathbf{I}$ given a particular image labeling configuration $\mathbf{Y}$, and the distribution $p(\mathbf{Y})$ encodes the prior of the pixel labels (*i.e.* dependencies of neighboring pixels).

Now we couple the model estimation and the pixel labeling tasks, and we formulate the segmentation problem within a joint MAP estimation, seeing that each separate task can be viewed as a MAP estimation problem. Given an image $\mathbf{I}$, the model parameters $\mathbf{X}$ and the image labeling $\mathbf{Y}$ are optimized at the same time:

$$(\mathbf{X}, \mathbf{Y})^{\text{opt}} = \arg \max_{(\mathbf{X},\mathbf{Y})} p(\mathbf{X}, \mathbf{Y}|\mathbf{I}) \tag{5.5}$$

The posterior distribution $p(\mathbf{X}, \mathbf{Y}|\mathbf{I})$ is a Gibbs distribution which can be written as:

$$p(\mathbf{X}, \mathbf{Y}|\mathbf{I}) \propto p(\mathbf{X}, \mathbf{Y}, \mathbf{I}) = \frac{1}{Z} \cdot \exp\{-E(\mathbf{X}, \mathbf{Y}, \mathbf{I})\} \tag{5.6}$$

where Z is a normalizing constant known as the partition function, and $E(\mathbf{X}, \mathbf{Y}, \mathbf{I})$ is an energy function of the configuration $\mathbf{X}, \mathbf{Y}$ and the observed image $\mathbf{I}$, which can be defined as the sum of the model-based energy, the pixel-based energy and the interaction energy:

$$E(\mathbf{X}, \mathbf{Y}, \mathbf{I}) = E^{(1)}(\mathbf{X}, \mathbf{I}) + E^{(2)}(\mathbf{Y}, \mathbf{I}) + E^{(3)}(\mathbf{X}, \mathbf{Y}) \tag{5.7}$$

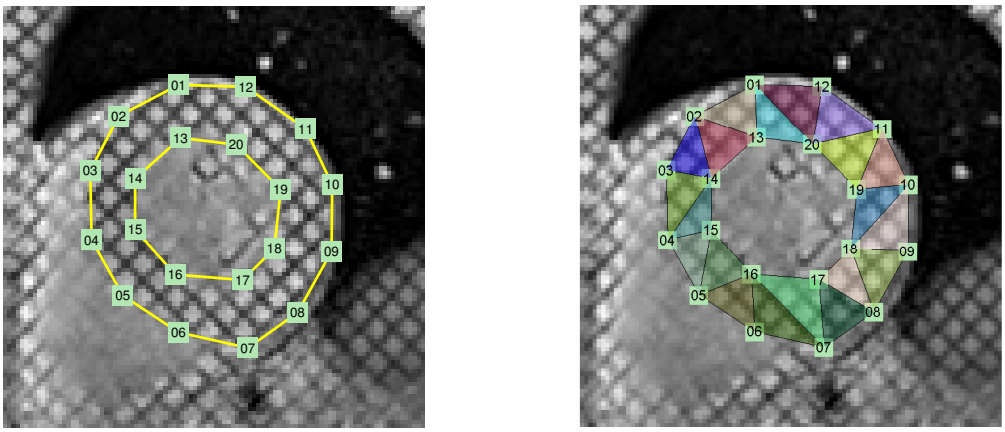
The model-based energy $E^{(1)}$ and the pixel-based energy $E^{(2)}$ can be inherited from the separate module, while the interaction energy $E^{(3)}$ is introduced as the key to couple the model fitting and image labeling in the joint framework. This energy function $E(\mathbf{X}, \mathbf{Y}, \mathbf{I})$ will be served as the objective function in the Markov Random Fields formulation, and the definitions of each energy term will be given in Section 5.4.

5.3 Shape Representation

Before we address the Markov Random Fields formulation of the joint model-pixel segmentation problem, we introduce a shape decomposition based on our model representation in Chapter 3. It is the key to combine the two tasks at the same time, since the shape decomposition brings the access to produce the interaction between model-based segmentation and pixel-based segmentation.

5.3.1 Shape Decomposition

As we described before, we represent the object of interest as a point-based model $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ with n control points lying on the boundary, where $\mathbf{x}_{i\in\{1,\cdots,n\}}$ denotes the coordinates of point i. For example, we show the point-based model of the left ventricle in a tagged cardiac image in Fig.5.3 (a), where the control points are marked in green and the reconstructed object contour by connecting the control points are shown in yellow.



(a) Point-based model (b) Model triangulation

Figure 5.3: Shape representation of 2D left ventricle.

Based on the point representation, we introduce a shape decomposition which partitions the object region into a number of triangle parts. The shape decomposition should satisfy the following conditions:

- Each component is composed of three control points, and its corresponding triangle region should be a part of the object region.
- These triangle regions should not overlap.
- The union of all the triangle regions covers the entire object region.

For example, we show a shape decomposition of the left ventricle model in Fig.5.3 (b), where each triangle part is represented in a unique color. In this example, we simply define the shape decomposition manually. Without loss of generality, such shape decomposition or polygon triangulation can be applied to any shape (heart, liver *etc.*) which can be represented as a polygonal area. It can be achieved automatically by shape decomposition algorithms (*i.e.* [Latecki 1999, De Berg 2000]). As a result of model triangulation, it produces a set of cliques where each clique consists of three points. We call the resulting clique set $\mathcal{A}$ as *data cliques* since they are used for calculating the image support. Using model triangulation can facilitate factorizing the regional-driven energy as well as introducing pixel and model interactions.

5.3.2 Shape Priors

Moreover, a set of *prior cliques* are considered to encode the local interactions of the model with respect to the shape priors. In Chapter 3, we proposed the L_1 sparse graphic model through MRF learning to obtain the set of prior cliques, whereas each clique encodes the dependencies of a triplet of points by two inner angles. Alternatively, we present another choice of the shape prior construction from a different perspective. It comes from the context of shape decomposition where the object is divided into several components, then the shape of the object can be described by these components themselves and the spatial relations between these components.

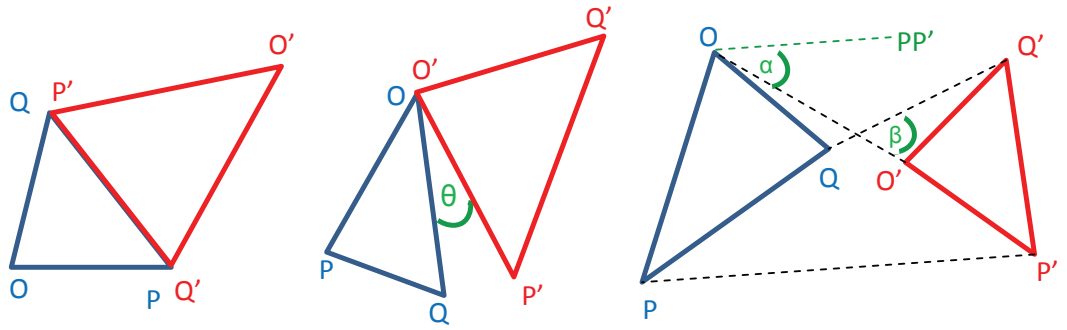
Based on the model triangulation, each component of the object is represented as a triangle area. Considering a triplet clique $a = (o, p, q) \in \mathcal{A}$, the triangle shape $\mathbf{x}_a = \{(O, P, Q)|O = \mathbf{x}_o, P = \mathbf{x}_p, Q = \mathbf{x}_q\}$ can be represented in a pose-invariant manner using two inner angles (α_a, β_a):

$$\alpha_a = \arccos \frac{\overrightarrow{OP} \cdot \overrightarrow{OQ}}{\|OP\| \, \|OQ\|}, \quad \beta_a = \arccos \frac{\overrightarrow{PO} \cdot \overrightarrow{PQ}}{\|PO\| \, \|PQ\|} \tag{5.8}$$

Given a training set $\mathcal{X} = \{\mathbf{X}_1, \cdots, \mathbf{X}_K\}$ composed of K object shapes where the control points have the same correspondence, for $\forall a \in \mathcal{A}$, we have a set of samples $\mathcal{X}_a = \{\mathbf{x}_a^1, \cdots, \mathbf{x}_a^K\}$ with respect to its triangle shape. A statistical model can be used to learn the probability density distributions $p_a(\alpha, \beta)$ of the inner angles of triplet a. For example, we can use Gaussian distribution $\mathcal{N}$ as statistical model, and the mean$\boldsymbol{\mu}_a$ and the variance matrix $\boldsymbol{\Sigma}_a$ are learned from the training set, then the probability distribution of the object component can be written as:

$$p_a(\alpha, \beta) = \mathcal{N}(\alpha, \beta|\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a), \quad a \in \mathcal{A} \tag{5.9}$$

Now let us consider the spatial relations between the components. Assuming two components are independent of a third one, we model the constraints between the object components by pairs of components. Taken any two different components $a, b \in \mathcal{A}$ of the object (*i.e.* two triplet cliques), we denote $\mathbf{x}_a = \{O, P, Q\}$ and $\mathbf{x}_b = \{O', P', Q'\}$ as the two corresponding triangles, while we make sure that the triplets (*i.e.* the order of O, P, Q) are oriented in the counter-clockwise direction. There are three situations of the two triplet cliques: they share (i) two common points, *i.e.* $|a \cap b| = 2$; (ii) one common point, *i.e.* $|a \cap b| = 1$; (iii) no common points, *i.e.* $|a \cap b| = 0$. We illustrate the three cases of any triplet pair in Fig.5.4. The dependencies of a pair of triple cliques can be defined as follows: (i) When there exists two common points, no more constraints need to be added since the spacial information is already included by the angles in both separate component; (ii) When there exists one common point, one angle θ determined by the two



(a) (b) (c)

Figure 5.4: Dependencies of triplet pair. (a) With 2 common points. (b) With 1 common point. (c) No common points.

clique is needed to measure the spatial constraint of the two components (see Fig.5.4 (b)); (iii) When there are no common points, two angles are required to measure the spatial relation of the two components (see Fig.5.4 (c)). We mention that the angle measurements are invariant to the global pose of the object (*i.e.* translation, rotation and scale). As a result, no shape alignments to the same referential are needed for both training samples and testing shapes. Specifically, the pose-invariant dependencies of two component can be written as:

$$\begin{cases} (P = Q')\&\&(Q = P') & (i) \\ O = O', \theta = \arccos \dfrac{\overrightarrow{OQ} \cdot \overrightarrow{O'P'}}{\|OQ\| \, \|O'P'\|} & (ii) \\ \alpha = \arccos \dfrac{\overrightarrow{OO'} \cdot \overrightarrow{PP'}}{\|OO'\| \, \|PP'\|}, \beta = \arccos \dfrac{\overrightarrow{OO'} \cdot \overrightarrow{QQ'}}{\|OO'\| \, \|QQ'\|} & (iii) \end{cases} \tag{5.10}$$

In other words, a pair of triangle components may contain four, five or six control points, according to three spatial relations respectively. Since we already defined the prior for each triangle shape, we only need the angles defined by the pairs to describe the pair constraints. We denote $\mathcal{B}$ as a set of cliques where each clique consists of five control points corresponding to all possible component pairs of case (ii), while $\mathcal{C}$ denotes a set of cliques where each clique consists of six control points corresponding to all possible component pairs of case (iii). Similarly to the prior for a single triplet, given a training set, we learn the probability density distributions of the angles $p_{b\in\mathcal{B}}(\theta), p_{c\in\mathcal{C}}(\alpha, \beta)$ in order to model the dependencies of the component pairs in a statistical manner.

To this end, the pose-invariant prior model is constructed by two types of priors: priors

of single triangle components and priors of pairs of triangle components. The global shape probability distribution $p(\mathbf{X})$ of the unknown model variables $\mathbf{X}$ is defined as the accumulation of the local interactions.

$$p(\mathbf{X}) = \frac{1}{Z'} \prod_{a \in \mathcal{A}} p_a(\alpha(\mathbf{x}_a), \beta(\mathbf{x}_a)) \prod_{b \in \mathcal{B}} p_b(\theta(\mathbf{x}_b)) \prod_{c \in \mathcal{C}} p_c(\alpha(\mathbf{x}_c), \beta(\mathbf{x}_c)) \tag{5.11}$$

As we defined before, $\mathcal{A}$ denotes the triplet cliques produced by model triangulation, $\mathcal{B}$ denotes all possible cliques produced by pairs of components including five control points, and $\mathcal{C}$ denotes all possible cliques produced by pairs of components including six control points. The corresponding probability density distributions p_a, p_b and p_c are learned from a training set, and Z' is a normalizing constant.

Now we summarize the shape representation as follows:

- Let $\mathcal{A}$ denote the *data cliques* which decompose the object model $\mathbf{X}$ into a number of triangle components.
- Let $\mathcal{C}_m = \{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ denote the *prior cliques*, including the triplet components and the related pairs of components.
- Let $p(\mathbf{X}) \propto \prod_{k \in \mathcal{C}_m} p_k(\mathbf{x}_k)$ denote the *shape priors* based on local interactions, where if $k \in \mathcal{A}, p_k = p_a$; if $k \in \mathcal{B}, p_k = p_b$ and if $k \in \mathcal{C}, p_k = p_c$.

This representation of the shape model introduces the concept of the object components (or object parts), and it models the shape priors by the prior of the single components and the spatial dependencies of the pairs of components. It brings two significant advantages: (1) The data cliques can facilitate the computation of the regional energy regarding to model to image likelihood as well as provide the pixel and model interactions. (2) The prior cliques defined by the components and their spatial dependencies produce a graph with less number of cliques, compared to the complete graph with all triplets of the control points. We suppose that the model has n control points, and it can be decomposed into m parts, usually we have $m < n$. In order to construct the pose-invariant priors, a number of all possible triplets C_n^3 is required without the MRF learning as described in Chapter 3, while the number of all possible component pairs is C_m^2. Building the shape prior based on the components rather than the single points can capture the shape properties without redundancies.

5.4 Markov Random Fields Formulation

Now we address the segmentation problem of joint model estimation and pixel labeling within a higher order Markov Random Field (MRF) formulation. The proposed graph

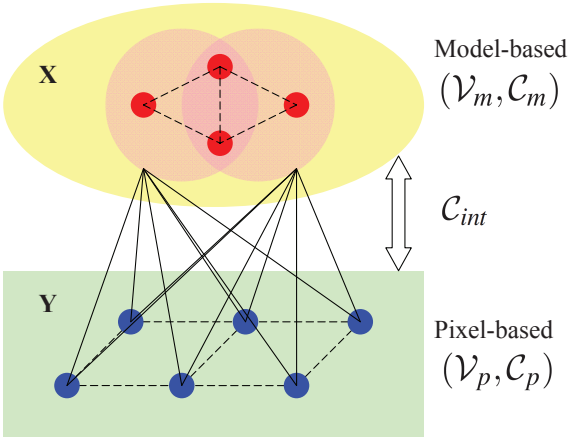


Figure 5.5: MRF graphical model for coupling the model space and the labeling space.

model G consists of:

- The model-based sub-graph $G_m = \{\mathcal{V}_m, \mathcal{C}_m\}$ consists of a set $\mathcal{V}_m = \{1, \cdots, n\}$ of *model* nodes (associated with n points in shape model) and a set of cliques $\mathcal{C}_m$ used in model-based segmentation independently.
- The pixel-based sub-graph $G_p = \{\mathcal{V}_p, \mathcal{C}_p\}$ consists of a set $\mathcal{V}_p = \{1, \cdots, k\}$ of *pixel* nodes (associated with k pixels in the image) and a set of cliques $\mathcal{C}_p$ introduced by pixel-based segmentation.
- The two sub-graphs are connected by a set of cliques $\mathcal{C}_{int}$, where both model nodes and pixel nodes are included in each clique.

We illustrate the proposed graph structure in Fig.5.5: (1) The yellow upper part represents the model-based sub-graph G_m, while the red nodes represent the model nodes $\mathcal{V}_m$ and the pink circles represent the local interactions $\mathcal{C}_m$ of the model. (2) The green lower part represents the pixel-based graph G_p, the blue nodes represent the pixel nodes $\mathcal{V}_p$ and the dashed lines represent the local dependencies $\mathcal{C}_p$ of the pixel nodes. (3) Last but not least, the solid lines represent the interactions $\mathbf{C}_{int}$ between the model nodes and the pixel nodes connecting the two separated sub-graphs. To sum up, we can express the whole graph model $G = \{\mathcal{V}_m \cup \mathcal{V}_p, \mathcal{C}_m \cup \mathcal{C}_p \cup \mathcal{C}_{int}\}$ with two types of nodes and three types of cliques. The definitions of each item will be given later.

Concerning a model node $i \in \mathcal{V}_m$, let X_i denote the latent random variable which indicates the coordinates of the associated control point. The variable X_i can take a particular configuration $\mathbf{x}_i$ from its candidate space $\mathcal{U}_i$. Theoretically, the variable X_i can take any

position in the image, but in practice, the candidate space is considered as a small subset of all pixel positions. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i\in\mathcal{V}_m}$ denote a model configuration consisting of all the model node variables over the image labeling space $\mathcal{U} = \prod_{i\in\mathcal{V}_m} \mathcal{U}_i$.

Similarly, concerning a pixel node $i \in \mathcal{V}_p$, let Y_i denote the latent random variable which indicates the label of the associated pixel. The variable Y_i can take a particular value y_i from the label space L (same for each pixel node). We define the pixel label space $L = \{0, \cdots m\}$, where m is the number of triangle parts produced by the clique set $\mathcal{A}$ as defined in the last section. The non-zero value $y_i \in \{1, \cdots, m\}$ indicates a particular part of the object, while zero value $y_i = 0$ indicates the background. Thus each pixel in the image can be assigned to either a part of the object or background. Let $\mathbf{Y} = \{y_i\}_{i\in\mathcal{V}_p}$ denote an image labeling configuration consisting of all the pixel node variables over the label space $\mathcal{L} = L^k$.

Now given an image $\mathbf{I}$, the segmentation problem is formulated as the estimation of optimal model configuration $\mathbf{X}$ over model space $\mathcal{U}$ and optimal labeling configuration $\mathbf{Y}$ over labeling space $\mathcal{L}$ simultaneously.

$$(\mathbf{X}, \mathbf{Y})^{\text{opt}} = \arg \min_{\mathbf{X}\in\mathcal{U}, \mathbf{Y}\in\mathcal{L}} E(\mathbf{X}, \mathbf{Y}, \mathbf{I}) \tag{5.12}$$

where the MRF energy $E(\mathbf{X}, \mathbf{Y}, \mathbf{I})$ consists of the model-based energy $E^{(1)}$, the pixel-based energy $E^{(2)}$ and the interaction-based energy $E^{(3)}$:

$$E(\mathbf{X}, \mathbf{Y}, \mathbf{I}) = E^{(1)}(\mathbf{X}, \mathbf{I}) + E^{(2)}(\mathbf{Y}, \mathbf{I}) + E^{(3)}(\mathbf{X}, \mathbf{Y}) \tag{5.13}$$

where the definition of each energy term are given as follows respectively.

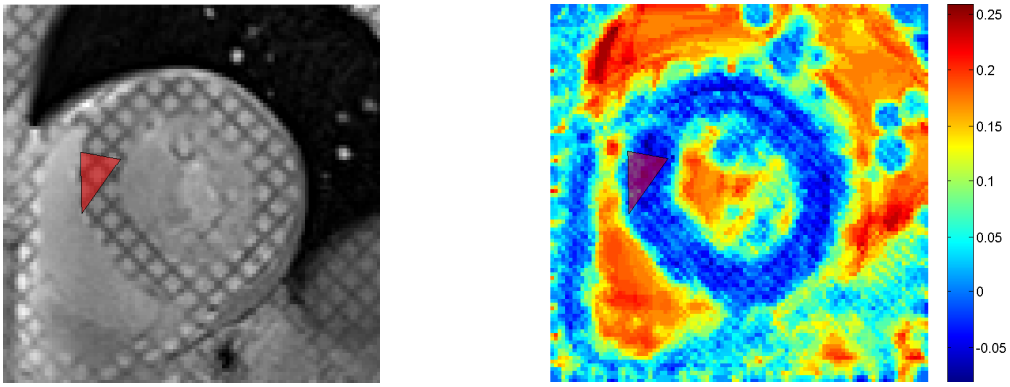
5.4.1 Model-based Energy

The model-based segmentation seeks for the optimal model parameters in order to make a compromise between the observed image and the shape prior constraints. According to Eq.(5.2), this energy is composed of a data term and a prior term.

$$-\log p(\mathbf{X}, \mathbf{I}) = -\log p(\mathbf{I}|\mathbf{X}) - \log p(\mathbf{X}) \tag{5.14}$$

The data term $-\log p(\mathbf{I}|\mathbf{X})$ encodes the image likelihood given a model configuration, while the prior term $-\log p(\mathbf{X})$ encodes spatial constraints of a model configuration with respect to the shape prior manifold.

We define the data term using the region-based criterion which captures the homogeneity properties of the different populations (*i.e.* object and background). As we have



(a) Test image (b) Likelihood map

Figure 5.6: Model-based data potential of a regional triplet.

described in Eq.(4.30) in Chapter 4, the region-based energy can be written as:

$$-\log p(\mathbf{I}|\mathbf{X}) = \sum_{i \in \Omega_{\text{obj}}(\mathbf{X})} -\log \frac{p_{\text{obj}}(\mathbf{I}_i)}{p_{\text{bck}}(\mathbf{I}_i)} \tag{5.15}$$

where $\Omega_{\text{obj}}(\mathbf{X})$ is the object region determined by the model configuration $\mathbf{X}$, and p_{obj}, p_{bck} are the appearance distribution models of object and background learned from a training set. This regional energy encourages the model to be located at the positions where its inner region covers as many object pixels as possible. Based on the model triangulation, the object region is decomposed into a set of triangle parts, thus the regional term can be factorized into higher order potentials $\Phi^{(1)}$ on data triplet set $\mathcal{A}$ which was introduced in Section 5.3.

$$-\log p(\mathbf{I}|\mathbf{X}) = \sum_{a \in \mathcal{A}} \Phi^{(1)}(\mathbf{x}_a), \;\; \Phi^{(1)}(\mathbf{x}_a) = \sum_{i \in \Omega(\mathbf{x}_a)} -\log \frac{p_{\text{obj}}(\mathbf{I}_i)}{p_{\text{bck}}(\mathbf{I}_i)} \tag{5.16}$$

where data potential $\Phi^{(1)}$ encodes the image likelihood over the triangle area $\Omega(\mathbf{x}_a)$. For a regional triplet a, the configuration of triplet $\mathbf{x}_a$ determines a triangle area $\Omega(\mathbf{x}_a)$ in the image domain (shown by red triangle in Fig.5.6). The data potential $\Phi^{(1)}(\mathbf{x}_a)$ is the integral of the pixel likelihood function (show in Fig.5.6 (b) where blue/red represents the smallest/largest value) over the triangle region $\Omega(\mathbf{x}_a)$. It can be computed efficiently using Divergence theorem which transforms the region integral into line integrals as we described in Chapter 4.

The prior term constrains the model configuration to remain in the allowable shape domain. It is formulated by the prior probability $p(\mathbf{X})$ defined in Eq.(5.11), and it is

factorized into potentials $\Psi^{(1)}$ defined on prior clique set $\mathcal{C}_m$.

$$-\log p(\mathbf{X}) = \sum_{k \in \mathcal{C}_m} \Psi^{(1)}(\mathbf{x}_k),\ \Psi^{(1)}(\mathbf{x}_k) = -\log p_k(\mathbf{x}_k) \tag{5.17}$$

where prior clique set $\mathcal{C}_m$ includes triplet cliques $\mathcal{A}$ which represent the object parts, fourth-order cliques $\mathcal{B}$ and fifth-order cliques $\mathcal{C}$ which represent the pairs of the parts. The distribution probabilities p_k are learned from training, where if $k \in \mathcal{A}, p_k = p_a$; if $k \in \mathcal{B}, p_k = p_b$ and if $k \in \mathcal{C}, p_k = p_c$ according to Eq.(5.11).

To sum up the two terms, we formulate the model-based energy $E^{(1)}(\mathbf{X}, \mathbf{I})$ as follows:

$$E^{(1)}(\mathbf{X}, \mathbf{I}) = \lambda_1 \cdot \sum_{a \in \mathcal{A}} \Phi^{(1)}(\mathbf{x}_a) + \lambda_2 \cdot \sum_{k \in \mathcal{C}_m} \Psi^{(1)}(\mathbf{x}_k) \tag{5.18}$$

where λ_1, λ_2 are the weights of data term and prior term respectively. The data clique set $\mathcal{A}$ is used to define the data term, and the prior clique set $\mathcal{C}_m$ is used to define the prior term. We also denote $\mathcal{C}_m$ as the model-based interactions, since $\mathcal{A} \subset \mathcal{C}_m$ is a subset of it.

5.4.2 Pixel-based Energy

Pixel-based segmentation assigns a label variable y_i for each image pixel i to be either part of the object or background. [Boykov 2001b] and [Shotton 2006] are some popular examples of pixel-based segmentation. According to Eq.(5.4), the energy over the pixel assignments consists of a data term and a prior term.

$$-\log p(\mathbf{Y}, \mathbf{I}) = -\log p(\mathbf{I}|\mathbf{Y}) - \log p(\mathbf{Y}) \tag{5.19}$$

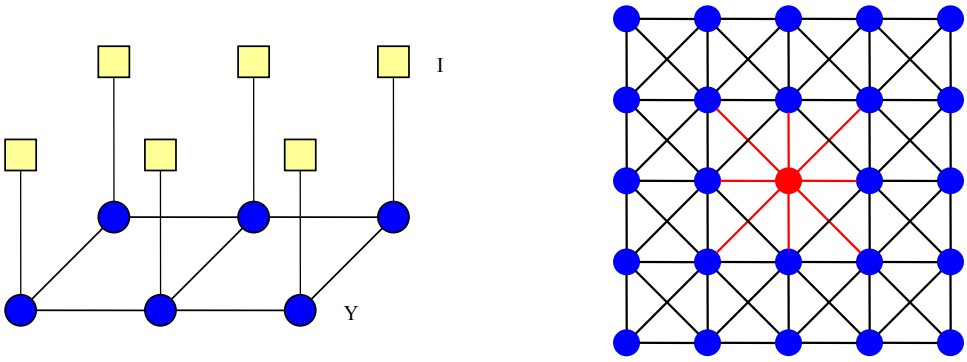
The data term $-\log p(\mathbf{I}|\mathbf{Y})$ encodes the image likelihood given a full assignment to all pixels. Assuming the label variables are independent, it can be computed as the sum of individual penalties for assigning pixel i to object or background:

$$-\log p(\mathbf{I}|\mathbf{Y}) = -\log \prod_{i \in \mathcal{V}_p} p(\mathbf{I}_i|y_i) = \sum_{i \in \mathcal{V}_p} \Phi^{(2)}(y_i) \tag{5.20}$$

where the unary likelihood potential is the emission model which is given by:

$$\Phi^{(2)}(y_i) = \begin{cases} -\log p_{\text{bck}}(\mathbf{I}_i) & \text{if } y_i = 0 \\ -\log p_{\text{obj}}(\mathbf{I}_i) & \text{otherwise} \end{cases} \tag{5.21}$$

where label $y_i = 0$ assigns the pixel i as background, otherwise non-zero value assigns the pixel i as object. As shown in Fig.5.7 (a), the label variable y_i (blue nodes) is only



(a) (b)

Figure 5.7: Pixel-based segmentation. (a) Graph. (b) Prior pairs using 8-connected neighborhood.

dependent on the corresponding pixel data I_i (yellow square), *i.e.* intensity, color or feature vector. We denote p_{bck} and p_{obj} as the appearance distribution model of object and background respectively. Given an observed pixel, it measures how likely it belongs to object/background. As we mentioned in Chapter 4, the appearance model can be represented as mixtures of Gaussians (GMM), or it can be represented as boosted classifiers whose output predicts whether and how likely the pixel is object. Both methods can be learned from a training set.

The prior term encourages the consistency of the pixel labels within a neighborhood system (*e.g.* 8-connected) which is defined by a pairwise clique set $\mathcal{C}_p$.

$$-\log p(\mathbf{Y}) = -\log \prod_{(i,j)\in\mathcal{C}_p} p(y_i, y_j) = \sum_{(i,j)\in\mathcal{C}_p} \Psi^{(2)}(y_i, y_j) \tag{5.22}$$

The clique set $\mathcal{C}_p$ consists of the pairs (edges in Fig.5.7 (b)) of the pixel nodes (nodes in Fig.5.7). Given a pixel node (red node in (b)), the 8-connected neighborhood consists the pairs (red edges) to the nearest 8 neighbors. The prior potential takes the form of an Ising model:

$$\Psi^{(2)}(y_i, y_j) = \begin{cases} 0 & \text{if } y_i = y_j \\ \gamma & \text{otherwise} \end{cases} \tag{5.23}$$

The potential constraints the neighboring pixel i and pixel j to have the same label, where γ is a penalizing parameter. Alternatively, the pairwise potential can take the form of a contrast sensitive Potts model, encouraging neighboring pixels with similar appearance to have the same label. It can be achieved by simply replacing the constant parameter γ by the function $\gamma(i, j)$ which measures the difference in the appearance between the two

neighboring pixels:

$$\gamma(i,j) = \exp\big(-\frac{\|\mathbf{I}_i - \mathbf{I}_j\|^2}{2\sigma^2}\big) \cdot \frac{1}{dist(i,j)} \tag{5.24}$$

where $\|\cdot\|^2$ is the appearance difference between the two pixels, σ^2 is the mean such distance across all neighboring pixels in the image, and $dist(i,j)$ is the spatial distance of the two pixels. $\gamma(i,j)$ is large when pixels i,j are similar and close to zero when two pixels are different in appearance, while $\gamma(i,j)$ also decreases as a function of distance between pixel i and j. Now the pixel-based energy $E^{(2)}(\mathbf{Y},\mathbf{I})$ can be formulated as:

$$E^{(2)}(\mathbf{Y},\mathbf{I}) = \lambda_3 \cdot \sum_{i\in\mathcal{V}_p} \Phi^{(2)}(y_i) + \lambda_4 \cdot \sum_{(i,j)\in\mathcal{C}_p} \Psi^{(2)}(y_i,y_j) \tag{5.25}$$

where λ_3, λ_4 are the weights of the unary potentials and the pairwise potentials respectively. The cliques $\mathcal{C}_p$ over the pixel nodes $\mathcal{V}_p$ is defined by a neighborhood (*i.e.* 8-connected) system.

5.4.3 Interaction-based Energy

The interaction energy is the key of propagating information between shape model and pixel labels in both ways, and thus producing segmentation which outperforms the independent methods. The consistency between model space and labeling space can be interpreted as follows. Given a shape model instance $\mathbf{X}$,

- Pixels far from the model boundary have less uncertainty of the labeling, *i.e.* if the pixel is inside the boundary, it should be labeled as object, otherwise it should be labeled as background.
- Pixels close to the model boundary have more uncertainty of the labeling, *i.e.* whether the pixel should be labeled as object or background, it depends on the data.

It is because that since the model (top-down cue) is an approximation of object shape, it is adequate to provide the location of the object, but it misses the details along the actual boundary. For example, let us represent the object shape with an extremely simple model such as a bounding box (often used for object detection). We are quite sure about the class of the pixels far from the box (inside is object, and outside is background), but we are less sure about whether the pixels close to the box edges are belong to object or background. Based on this observation, we can define the confidence of the pixel labeling being object or background as a signed distance $dist(i,\mathbf{X})$ from the pixel i to the model boundary.

$$dist(i,\mathbf{X}) = \begin{cases} -\min_{\mathbf{x}_a\mathbf{x}_b\in B(\mathbf{X})} d(i,\mathbf{x}_a\mathbf{x}_b) & \text{if } i \text{ is inside} \\ \min_{\mathbf{x}_a\mathbf{x}_b\in B(\mathbf{X})} d(i,\mathbf{x}_a\mathbf{x}_b) & \text{if } i \text{ is outside} \end{cases} \tag{5.26}$$

where $\mathbf{x}_a\mathbf{x}_b$ is a line segment specified by two points $\mathbf{x}_a, \mathbf{x}_b$ on the model boundary $B(\mathbf{X})$, $d(i, \mathbf{x}_a\mathbf{x}_b)$ denotes the distance between the pixel i and the line segment $\mathbf{x}_a\mathbf{x}_b$. The distance $dist(i, \mathbf{X})$ is negative when the pixel i is inside the model boundary, otherwise it is positive.

To combine model estimation and pixel labeling, [Kumar 2005] defines an energy term using the signed distance map:

$$E(\mathbf{X}, \mathbf{Y}) = \sum_{i \in \mathcal{V}_p} \Phi(y_i|\mathbf{X}) \tag{5.27}$$

where $\Phi(y_i|\mathbf{X})$ is a function of the pixel's distance $dist(i, \mathbf{X})$ and the pixel label y_i. Due to the fact that the signed distance map can be calculated only when all the states of the shape variables $\mathbf{X}$ are given, the optimization the model-based and the pixel-based segmentation is performed in an interleaved way : (1) Given an estimate of $\mathbf{Y}$, they sample the model $\mathbf{X}$; (2) Given the distribution of $\mathbf{X}$, they optimize $\mathbf{Y}$. However, this minimization procedure does not provide any guarantee on the optimality properties of the obtained solution, and it does not fully exploit both top-down and bottom-up information. In this context, we prefer to optimize both model variables and pixel labels in a single shot framework, while using the information propagated from each other. The problem is the distance map (which links the model space and pixel space) depends on the global shape model. If we can transform the interactions between the global model and single pixel label into the interactions between the local shape and single pixel, the joint optimization problem in both model and pixel space can be solved in a single shot framework.

We introduce a new interaction-based energy that encourages the consistency between the model-based segmentation and the pixel-based segmentation. Instead of connecting each pixel label with the entire model variables, we consider the interaction between each pixel label with each model part which is defined by the model triangulation. Thus the model-pixel interaction energy is defined by:

$$E^{(3)}(\mathbf{X}, \mathbf{Y}) = \sum_{(i,a) \in \mathcal{C}_{int}} \Phi^{(3)}(y_i, \mathbf{x}_a) \tag{5.28}$$

The interaction clique set $\mathcal{C}_{int} = \{(i, a)|i \in \mathcal{V}_p, a \in \mathcal{A}\}$ connects every pixel with every regional triangle (triplet of model points). The third-order potential $\Phi^{(3)}$ penalizes a pixel label conditioned on a regional triplet.

$$\Phi^{(3)}(y_i, \mathbf{x}_a) = \begin{cases} -[i \in \Omega(\mathbf{x}_a)]\, dist(i, \mathbf{x}_a) & \text{if } y_i = l_a \\ -[i \in R(\mathbf{x}_a)]\, dist(i, \mathbf{x}_a) & \text{if } y_i = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.29}$$

- Let $l_a \in \{1, \cdots, m\}$ denote a known label of the triplet a, which means that the triangle $\mathbf{x}_a$ is indexed as l_a in the object, while m is the total number of the object parts.

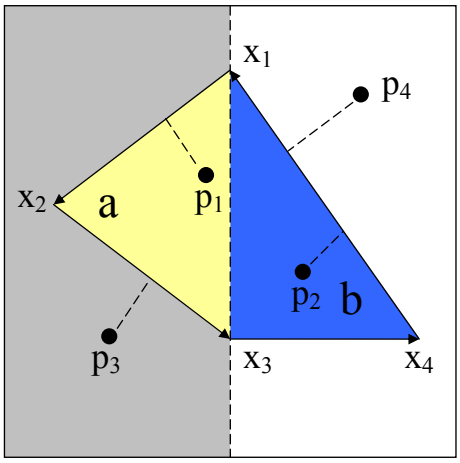


Figure 5.8: Interaction between a pixel label and a triplet.

- Let $dist(i, \mathbf{x}_a)$ denote the distance of pixel i related the triangle $\mathbf{x}_a$. Particularly, the absolute distance is defined as the minimum distance of the pixel i to the model edges of the triangle, while the model edges are the segments on the object model boundary.

- Let $R(\mathbf{x}_a)$ denote a subset of the region outside of the triangle $\mathbf{x}_a$. Assuming the triangle is oriented in counter-clockwise, the point falling into region $R(\mathbf{x}_a)$ should satisfy that it is on the right side of the model edges as well as on the left side of the non-model edges.

This potential expresses the relation between a pixel i and a triplet a: (1) If the pixel i is inside the triangle region $\Omega(\mathbf{x}_a)$, then assigning the pixel label y_i to the triplet label l_a is encouraged by the distance $dist(i, \mathbf{x}_a)$. (2) If the pixel i is located in $R(\mathbf{x}_a)$ (the background region close to the triangle $\mathbf{x}_a$), then the pixel label assignment to the background label $y_i = 0$ is encouraged by the distance $dist(i, \mathbf{x}_a)$.

We show an example composed of two parts a, b (shown in yellow and blue respectively) in Fig.5.8, and we define $l_a = 1, l_b = 2$. Considering a pixel and the triplet part a, (1) If the pixel (*e.g.* point p_1) is inside of the triangle $\Omega(\mathbf{x}_a)$ (yellow region), the cost of assigning the pixel label to triplet part a (*i.e.* $y_i = 1$) is the negative of the distance (*e.g.* $dist(i, \mathbf{x}_a) = min\{d(i, \mathbf{x}_1\mathbf{x}_2), d(i, \mathbf{x}_2\mathbf{x}_3)\}$). (2) If the pixel (*e.g.* point p_3) is inside the region $R(\mathbf{x}_a)$ (gray region), the cost of assigning the pixel label to background (*i.e.* $y_i = 0$) is the negative distance $dist(i, \mathbf{x}_a)$. Considering all pairs of pixel and triplet, the interaction potentials are obtained in the table below, where all the pixels can be divided into four cases such as p_1, p_2, p_3, p_4. In fact for any pixel i, let $\Phi(y_i, \mathbf{X}) = \Phi(y_i, \mathbf{x}_a) + \Phi(y_i, \mathbf{x}_b)$

denote the sum of the potentials of this pixel and all triplets, which has the analogical interpretation as $\Phi(y_i, \mathbf{X})$ used in Eq.(5.27). The minimized $\Phi(y_i, \mathbf{X})$ over the pixel label y_i is the approximate distance of the pixel to the model, while the optimal pixel label indicates the class (positive values represent object, zero value represents background).

		$y_i = 0$	$y_i = 1$	$y_i = 2$
$i = p_1$	$\mathbf{x}_a$	0	$-dist(i, \mathbf{x}_a)$	0
	$\mathbf{x}_b$	0	0	0
$i = p_2$	$\mathbf{x}_a$	0	0	0
	$\mathbf{x}_b$	0	0	$-dist(i, \mathbf{x}_b)$
$i = p_3$	$\mathbf{x}_a$	$-dist(i, \mathbf{x}_a)$	0	0
	$\mathbf{x}_b$	0	0	0
$i = p_4$	$\mathbf{x}_a$	0	0	0
	$\mathbf{x}_b$	$-dist(i, \mathbf{x}_b)$	0	0

Table 5.1: The potentials of pixel label and triplet.

So far, all energy terms with respect to model-based, pixel-based and model-pixel module are defined in Eq.(5.18, 5.25, 5.28). Hence, the proposed higher order MRF energy (5.13)) can be written as follows:

$$\begin{aligned} E(\mathbf{X}, \mathbf{Y}, \mathbf{I}) =& \lambda_1 \sum_{a \in \mathcal{A}} \Phi^{(1)}(\mathbf{x}_a) + \lambda_2 \sum_{c \in \mathcal{C}_m} \Psi^{(1)}(\mathbf{x}_c) \\ &+ \lambda_3 \sum_{i \in \mathcal{V}_p} \Phi^{(2)}(y_i) + \lambda_4 \sum_{(i,j) \in \mathcal{C}_p} \Psi^{(2)}(y_i, y_j) \\ &+ \sum_{(i,a) \in \mathcal{C}_{int}} \Phi^{(3)}(y_i, \mathbf{x}_a) \end{aligned} \tag{5.30}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the weights for the different energy terms.

To search for an optimal model configuration, an iterative strategy is used. Given an initial position, a set of displacements of the model node is chosen as its label space; it is adapted with a coarse-to-fine setting during the iterations. For higher-order MAP-MRF inference, some efficient state of the art methods such as the TRW-S algorithms [Kolmogorov 2006] and Dual decomposition [Komodakis 2011b] are available. In our case, we project our higher order MRF into a pairwise MRF and employ the TRW-S [Kolmogorov 2006] inference algorithms to optimize the above MRF energy of $\mathbf{X}$ and $\mathbf{Y}$. It is achieved by reducing a triplet (3 control points) to a model node in the graphical model, while adding the constraints that the same point (involved in related triplets) should coincide. In this manner, all the triplet-related energies become unary terms; the

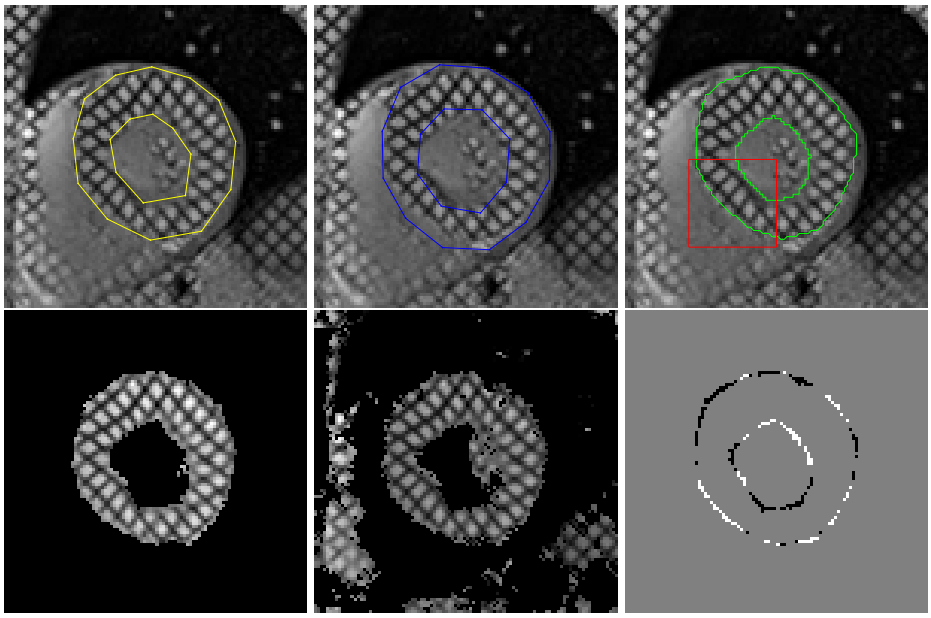
other higher order energies become pairwise terms, while the energy function definitions remain the same.

5.5 Experimental Validation

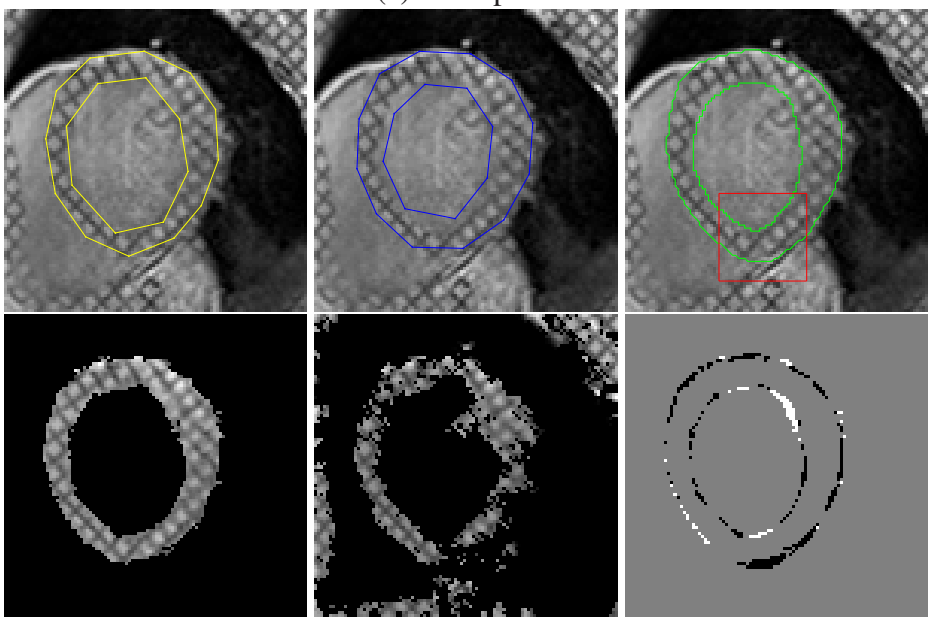
A dataset of 40 2D tagged cardiac Magnetic Resonance (MR) images is used to validate our method. The segmentation ground truth of the dataset is provided by experts, and it is employed for both training and validation. With regard to the shape model, 20 control points and 20 regional triplets are defined manually. For each original image, a preliminary step of object detection is performed to extract a sub-image with 100×100 pixels which contains the object instance. This object detection can be obtained from a top-down segmentation, or we manually choose the location of the sub-image. Then we apply the model-pixel combined segmentation on the sub-image. In order to deal with image appearance properties, we consider Gabor features to represent the texture pattern of the tagged MR images. Using a training set, Adaboost algorithm is used to learn to the classifier of object/background in order to model the image appearance model. We performed a leave-one-out cross validation on the whole dataset. The experiments were run on a 2.8GHz, 12GB Ram computer and our segmentation took a couple of seconds per image.

Some final visual results of two test images are presented in Fig.5.9. The first column is our results in both model space $\mathbf{X}$ and label space $\mathbf{Y}$. In the upper sub-figure, the yellow contours represent the model localization results, while in the lower sub-figure the pixels labeled as object are shown in gray level as the original intensities and the pixels labeled as background are shown in black. The second column shows the results of independent model-based module (using only energy $E^{(1)}$) and pixel-based module (using only energy $E^{(2)}$) respectively. Similarly, the upper sub-figure shows the model results with blue contours and the lower sub-figure represents the pixel-based segmentation results. The third column compares our results with the ground truth. In the upper sub-figure, the ground truth of the object contours is shown as green contours. The lower sub-figure shows the difference image between our labeling result and the ground truth, where the gray pixels are correct labeled, the white/black pixels are wrongly labeled as object/background. Fig.5.10 zooms in the area bounded by red box shown in the third column. The yellow contours in the left images represent the model results of joint model-pixel method, and the blue contours in the middles images represent the model results of independent model-based module, and the right images show the ground truth mask in red.

From the above results, we can see that the model-pixel combined method provides better segmentation performance than only pixel-based or model-based method. The only pixel-based method is sensitive to the complicated background and noise. The only model-

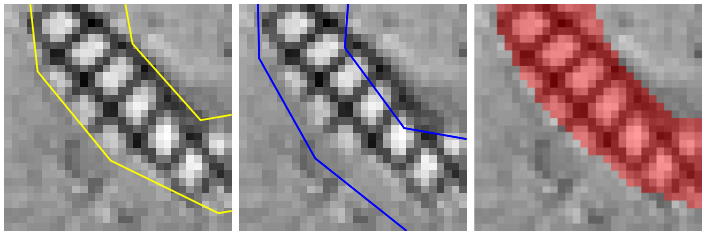


(a) Example 1

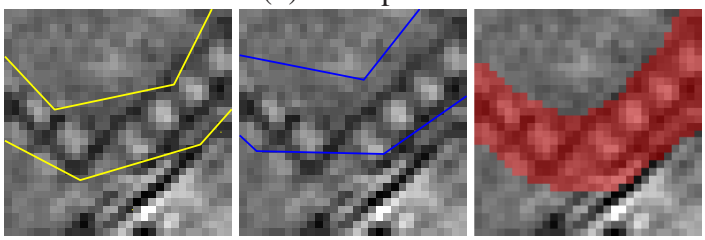


(b) Example 2

Figure 5.9: Segmentation results of 2 test images. The columns from left to right are our results, only model/pixel-based results, ground truth/comparison.



(1) Example 1



(2) Example 2

Figure 5.10: Zoom effects of 2 test images. The columns from left to right are the model results of the combined method, the independent model-based results, ground truth.

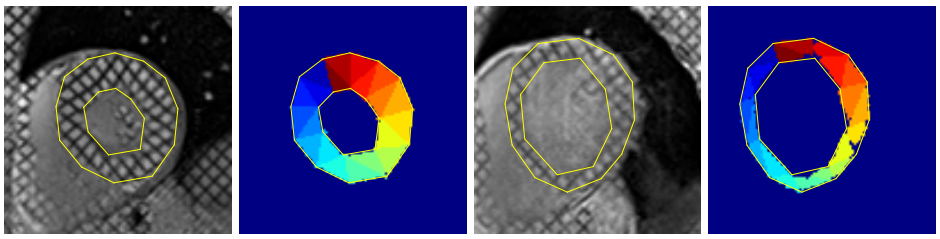


Figure 5.11: Both model localization and pixel labeling results.

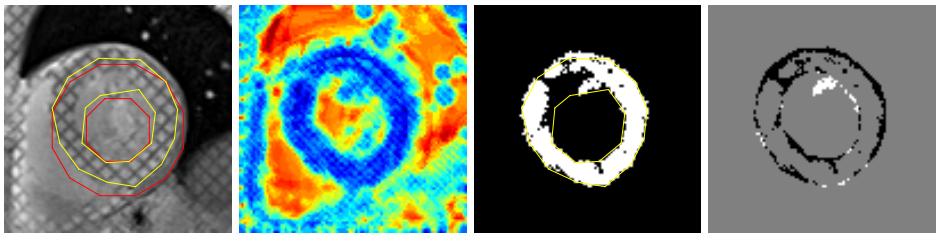


Figure 5.12: An intermediate iteration. From left to right: original image, likelihood in color map, labeling, difference map between current result and ground truth.

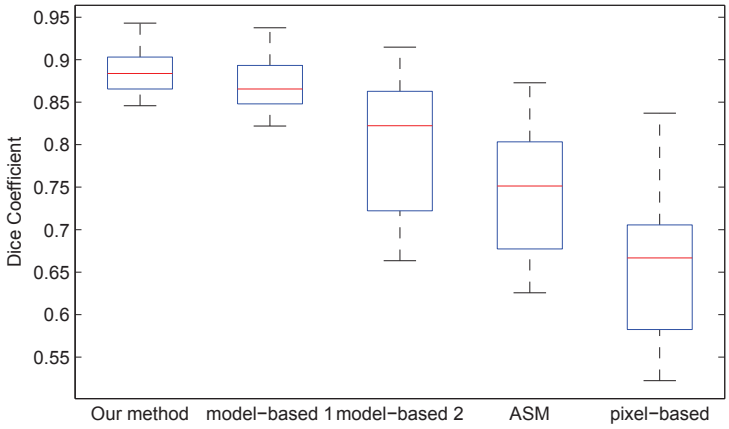


Figure 5.13: Comparisons on dice coefficients.

based results are globally correct, but do not give accurate segmentation around the boundary locally (*e.g.* see the zoom in effects in Fig.5.10). Our integrated framework can overcome this defect, showing flexibility of having local deviations as well as producing pixel-wise labeling result at the same time. Furthermore, our method also deals well with the varying scales of the object (*e.g.* the scale of the inner contour shows large variability) due to the pose-invariant shape prior.

We also mention that our pixel-based model using multi-class labels. Fig.5.11 shows the image labeling results in a color map, where each color represents one label for one triangle part of the object. The multi-label segmentation can provide more information of the local parts of the object.

Fig.5.12 shows an intermediate iteration of a test image. Although the current model localization at this iteration (yellow contour) is not close to ground truth, the model-pixel combined labeling result (the third image) benefits from both model and pixel labeling information and produces better performance than independent segmentation. This also illustrates that the hybrid method is robust to initialization (red contour).

For both quantitative evaluation and comparison purposes, we present the boxes from left to right in Fig.5.13, representing the Dice coefficient distributions obtained respectively by (1) our hybrid method, (2) model-based method 1 (using only model module), (3) model-based method 2 [Besbes 2009], (4) standard Active Shape Model (ASM) [Cootes 1995] and (5) pixel-base method (using only pixel module). Noted that a higher Dice coefficient implies a better segmentation result, Fig. 5.13 highlights the better performance of our method compared with the previous methods. The following table shows the statistical values of the dice coefficient distributions of each method. One can note that,

although the performance of only model-based model is competitive on statistic figures, the hybrid method outputs much more delicate visual performance.

Method	mean	std
our method	0.8882	0.0263
model-based 1	0.8709	0.0298
model-based 2	0.7966	0.0761
ASM	0.7439	0.0710
pixel-based	0.6560	0.0781

Table 5.2: The statistical parameters of dice coefficient distributions.

5.6 Conclusion

In this chapter we propose a novel approach to address jointly model/image-based segmentation using a higher order graphical model. The proposed formulation can easily encode regional support, meanwhile being able to account for shape variability unseen during training. Furthermore, it produces states of the art results in particular when exact boundary delineation is of interest through the combined model-pixel graph. To the best of our knowledge, this is the first method that recovers a consistent solution between the model and the image space in a single shot optimization framework, while being pose-invariant.

Chapter 6

Conclusion

In this thesis, we propose novel knowledge-based object segmentation approaches which incorporate shape priors within higher-order Markov Random Fields. It is motivated by the observation that using high-level information, *i.e.* prior knowledge regarding the geometric properties of the class of objects, can make the segmentation robust to the disturbance in low-level information, *i.e.* noise, non-discriminative visual support and occlusions. In this thesis, we address the knowledge-based segmentation task by solving two major concerns: (1) How to build a statistical shape model? (2) How to incorporate the prior knowledge in the segmentation framework?

6.1 Contributions

We represent the shape model as a point-based graphical model, where each node in the graph corresponds to a control point on the shape boundary, while each clique in the graph corresponds to the dependencies of a subset of the control points. In particular, choosing the clique size as three *i.e.* each clique consists of three different nodes, the local spatial constraint of the three related control points is modeled by the statistics on the angle measurement which inherits invariance to global transformations (*i.e.* translation, rotation and scale). The shape manifold is constructed through the L_1 sparse higher-order graph, accumulating the local constraints. The sparse graph consists of a subset of cliques from all possible second-order cliques, and it is learned through MRF training using dual decomposition. The pose-invariant shape prior through sparse higher-order graph can be easily encoded in a higher-order Markov Random Field.

In order to incorporate the prior knowledge in the segmentation, we propose a model-based segmentation method. It is formulated as estimating the object boundary model in the observed image, combining the prior knowledge as well as the image support. We

address the segmentation as a maximum a posteriori (MAP) estimation, since the probabilistic framework has the advantage to include a statistical prior model over the solution space. In particular, the model estimation is formulated as minimizing an energy function over the model parameters (*i.e.* the positions of the control points) solutions, and the regional statistics, boundary support as well as prior knowledge are encoded through a global formulation. We embed this framework in a Markov Random Filed, since efficient discrete MRF optimization algorithms have been developed and they can provide optimal or sub optimality guarantees. The shape prior is expressed by the second-order MRF potentials where each potential encodes the local statistical prior. The boundary support and regional statistics are integrated by the pairwise or second-order potentials on the model boundary cliques (pairwise cliques in 2D cases, and second-order cliques in 3D cases). The use of Divergence theorem provides an exact calculation of regional statistics acting on the image or a derived feature space. The considered framework is optimized using dual decomposition and used towards 2D and 3D object segmentation with promising result.

Furthermore, we propose a novel framework for joint model-pixel segmentation, in order to integrate both top-down and bottom-up approaches in a unified framework towards a more refined segmentation. The proposed framework simultaneously solves the problem in both model space and image space. The graphical model consists of both model nodes (positions of control points) and pixel nodes (labels of pixels), and model-interaction, pixel-interaction, and model-pixel interaction. In particular, a model decomposition associates the model parts with pixels labeling which aims to create consistency between the model and the image space. The proposed objective function aims to: (i) assign labels to image pixels in order to maximize the image likelihood, (ii) deform a point-based model in order to maximize the geometric likelihood of the model as well as the model-to-image likelihood, (iii) impose consistency between the two label spaces. The resulting higher order graphical model formulation is solved by using a state of the art message passing algorithm. The promising results on a challenging clinical setting demonstrate the potentials of our method.

To sum up, the main contributions of this thesis are the following:

- We propose a pose-invariant statistical shape model. It can capture linear and non-linear shape variations of a class of objects. The local model has much greater flexibility than global models, and it is able to account for shape variability unseen during training. It can be learned from a small training set. A sparse graph structure achieved from Markov Random Field learning has boosting efficiency while preserving its ability to represent the variations. It does not need aligning the shapes in a common coordinate frame.

- We propose a model-based segmentation using higher-order Markov Random Field.

We develop a global approach to jointly encode the regional statistics, boundary support, as well as prior knowledges within a probabilistic framework. The pose-invariant priors are encoded by second-order MRF potentials. The regional statics is exactly factorized into pairwise or second-order terms using Divergence theorem. The proposed segmentation method is robust to noise, partial object occlusions and initializations. It is efficient and does not suffer from bad local minima issues using developed MRF optimization algorithms.

- We propose a novel approach to address jointly model/image-based segmentation using a higher order graphical model. It produces states of the art results through the combined model-pixel graph, in particular when exact boundary delineation is of interest. To the best of our knowledge, this is the first method that recovers a consistent solution between the model and the image space in a single shot optimization framework, while being pose-invariant.

6.2 Future Work

Regarding our shape model, we capture a sparse graph structure using MRF learning using dual decomposition. In particular, we associate each triplet clique in the complete graph with a weight parameter, and we learn the weight parameters by using MRF learning. Then we use a threshold to choose a number of cliques who have the weight value larger than the threshold, and these selected clique compose the sparse graph. However, the threshold is manually defined and it can influence the strength of resulting shape model. In some cases, the sparsity is not guaranteed by the L_1 norm. More investigation and analysis have to be studied on the sparse graph construction.

Spatio-temporal shape modeling is another natural extension to account for the temporal nature of the object in segmentation and tracking (3D+time). It is of great interest when studying the dynamic anatomical structures in many applications especially in medical image analysis. The spatio-temporal shape can be modeled in a global way by principle component analysis (PCA) as in active shape models, but we are more interested in extending the shape model using the graph representation where temporal connectivities will be introduced so that it can have local flexibility.

Moreover, although our shape model can naturally deal with partial occlusion, but it is not designed for objects with overlapping parts. We believe that a part-based representation is more suitable for variations in articulated objects and it can account for self-occlusions.

In our segmentation framework, the exact region statistics are encoded as the image support. However, to computer the regional terms in 3D cases involves complex com-

putation and it is slow, since all the pixels inside of the model boundary have to been considered. Integrating anatomical landmark extraction in the segmentation can improve the accuracy and efficiency. The landmark extraction based on geometric information is known to be robust to the initial conditions, and it also can reduce the candidate space for each control point, thus boosting the segmentation speed. Efficient/intelligent sampling of the search space is another important direction that could burden computational complexity, while providing more accurate solutions. The use of marginals as suggested in [Glocker 2008] is under consideration.

To extend our model-pixel segmentation method to 3D cases is another future research direction. The current method is now limited in 2D cases because the shape triangulation which associates the model parts and pixel labels can be only applied in 2D. Finding a way to produce consistence between the model space and the labeling space in 3D can largely exploit the bottom-up and top-down approaches at the same time, which is especially profitable in medical images where 3D images are widely used.

Last but not least, the objective function which we formulate for knowledge-based segmentation consists of many different terms *i.e.* boundary terms, regional terms, prior terms, and the weight parameters of these terms which control the contribution of each modular are usually manually adjusted. It is not controllable to find the best parameters when the number of the terms are increasing. MRF learning could be a natural path towards learning these parameters from a training set and it could greatly enhance performance of the method.

Publications of the Author

International Conferences

- Bo Xiang, Chaohui Wang, J-F Deux, Alain Rahmouni and Nikos Paragios. *Tagged cardiac MR image segmentation using boundary & regional-support and graph-based deformable priors*. In IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI), pages 1706–1711. IEEE, 2011. **Oral presentation.**

- Bo Xiang, Chaohui Wang, J-F Deux, Alain Rahmouni and Nikos Paragios. *3D cardiac segmentation with pose-invariant higher-order MRFs*. In IEEE International Symposium on Biomedical Imaging (ISBI), pages 1425–1428, 2012. **Oral presentation.**

- Bo Xiang, Nikos Komodakis and Nikos Paragios. *Pose invariant deformable shape priors using L1 higher order sparse graphs*. In International Symposium on Visual Computing (ISVC). Springer, 2013. **Oral presentation.**

- Bo Xiang, J-F Deux, Alain Rahmouni and Nikos Paragios. *Joint model-pixel segmentation with pose-invariant deformable graph-priors*. In Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, 2013.

International Journal Submission

- Nikos Komodakis, Bo Xiang and Nikos Paragios. *A Framework for Efficient Structured Max-Margin Learning of High-Order MRF Models*. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Bibliography

[Alahari 2010] Karteek Alahari, Pushmeet Kohli and Philip HS Torr. *Dynamic hybrid algorithms for MAP inference in discrete MRFs*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 10, pages 1846–1857, 2010. 30

[Ali 2007] Asem M Ali, Aly A Farag and Ayman S El-Baz. *Graph cuts framework for kidney segmentation with prior shape constraints*. In Medical Image Computing and Computer-Assisted Intervention, pages 384–392. Springer, 2007. 24, 25

[Alpert 2007] Sharon Alpert, Meirav Galun, Ronen Basri and Achi Brandt. *Image segmentation by probabilistic bottom-up aggregation and cue integration*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2007. 19

[Andreopoulos 2008] Alexander Andreopoulos and John K Tsotsos. *Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI*. Medical Image Analysis, vol. 12, no. 3, pages 335–357, 2008. 17

[Ayed 2009] Ismail Ben Ayed, Kumaradevan Punithakumar, Shuo Li, Ali Islam and Jaron Chong. *Left ventricle segmentation via graph cut distribution matching*. In Medical Image Computing and Computer-Assisted Intervention, pages 901–909. Springer, 2009. 24

[Behiels 1999] Gert Behiels, Dirk Vandermeulen, Frederik Maes, Paul Suetens and Piet Dewaele. *Active shape model-based segmentation of digital X-ray images*. In Medical Image Computing and Computer-Assisted Intervention, pages 128–137. Springer, 1999. 15

[Besbes 2009] Ahmed Besbes, Nikos Komodakis, Georg Langs and Nikos Paragios. *Shape priors and discrete mrfs for knowledge-based segmentation*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1295–1302. IEEE, 2009. 41, 70, 96, 125

[Bishop 2006] Christopher M Bishop and Nasser M Nasrabadi. Pattern recognition and machine learning, volume 1. springer New York, 2006. viii, 82, 94

[Blake 2004] Andrew Blake, Carsten Rother, Matthew Brown, Patrick Perez and Philip Torr. *Interactive image segmentation using an adaptive GMMRF model*. In European Conference on Computer Vision, pages 428–441. Springer, 2004. 29

[Borenstein 2002] Eran Borenstein and Shimon Ullman. *Class-specific, top-down segmentation*. In European Conference on Computer Vision (ECCV), pages 109–122. Springer, 2002. 105

[Borenstein 2004] Eran Borenstein and Shimon Ullman. *Learning to segment*. In European Conference on Computer Vision (ECCV), pages 315–328. Springer, 2004. 105

[Borenstein 2008] Eran Borenstein and Shimon Ullman. *Combined top-down/bottom-up segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 12, pages 2109–2125, 2008. ix, 105

[Boros 2006] Endre Boros, Peter L Hammer and Gabriel Tavares. *Preprocessing of unconstrained quadratic binary optimization*. 2006. 30

[Boykov 2001a] Y. Boykov, O. Veksler and R. Zabih. *Fast approximate energy minimization via graph cuts*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pages 1222–1239, 2001. 29, 30

[Boykov 2001b] Yuri Y Boykov and M-P Jolly. *Interactive graph cuts for optimal boundary & region segmentation of objects in ND images*. In IEEE International Conference on Computer Vision (ICCV), volume 1, pages 105–112. IEEE, 2001. vii, 19, 20, 29, 116

[Boykov 2003] Yuri Boykov and Vladimir Kolmogorov. *Computing geodesics and minimal surfaces via graph cuts*. In IEEE International Conference on Computer Vision, pages 26–33. IEEE, 2003. 28

[Boykov 2004] Yuri Boykov and Vladimir Kolmogorov. *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pages 1124–1137, 2004. 20, 29

[Boykov 2006] Yuri Boykov and Gareth Funka-Lea. *Graph cuts and efficient ND image segmentation*. International Journal of Computer Vision, vol. 70, no. 2, pages 109–131, 2006. 7, 8, 21, 28, 29, 106

[Bray 2006] Matthieu Bray, Pushmeet Kohli and Philip HS Torr. *Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts*. In European Conference on Computer Vision (ECCV), pages 642–655. Springer, 2006. 104

[Caselles 1993] Vicent Caselles, Francine Catté, Tomeu Coll and Françoise Dibos. *A geometric model for active contours*. Numerische mathematik, vol. 66, no. 1, pages 1–31, 1993. 11

[Caselles 1997] V. Caselles, R. Kimmel and G. Sapiro. *Geodesic active contours*. International Journal of Computer Vision, vol. 22, no. 1, pages 61–79, 1997. 11

[Chan 2001] Tony F Chan and Luminita A Vese. *Active contours without edges*. IEEE Transactions on Image Processing, vol. 10, no. 2, pages 266–277, 2001. 7, 12, 19

[Chan 2005] Tony Chan and Wei Zhu. *Level set based shape prior segmentation*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 2, pages 1164–1170. IEEE, 2005. 24

[Chen 2002] Yunmei Chen, Hemant D Tagare, Sheshadri Thiruvenkadam, Feng Huang, David Wilson, Kaundinya S Gopinath, Richard W Briggs and Edward A Geiser. *Using prior shapes in geometric active contours in a variational framework*. International Journal of Computer Vision, vol. 50, no. 3, pages 315–328, 2002. 69

[Cohen 1991] Laurent D Cohen. *On active contour models and balloons*. CVGIP: Image understanding, vol. 53, no. 2, pages 211–218, 1991. 10

[Cohen 1993] Laurent D Cohen and Isaac Cohen. *Finite-element methods for active contour models and balloons for 2-D and 3-D images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 11, pages 1131–1147, 1993. 10

[Comaniciu 2002] Dorin Comaniciu and Peter Meer. *Mean shift: A robust approach toward feature space analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pages 603–619, 2002. 7

[Cootes 1992] Tim F Cootes, David H Cooper, Christopher J Taylor and Jim Graham. *Trainable method of parametric shape description*. Image and Vision Computing, vol. 10, no. 5, pages 289–294, 1992. 40

[Cootes 1995] Timothy F Cootes, Christopher J Taylor, David H Cooper, Jim Graham*et al.* *Active shape models - their training and application*. Computer Vision and Image Understanding, vol. 61, no. 1, pages 38–59, 1995. 7, 13, 38, 69, 97, 125

[Cootes 1996] Timothy F Cootes and Christopher J Taylor. *Data driven refinement of active shape model search*. In British Machine Vison Conference, pages 383–392, 1996. 40

[Cootes 1998] Timothy F Cootes, Gareth J Edwards and Christopher J Taylor. *Active appearance models*. In European Conference on Computer Vision (ECCV), pages 484–498. Springer, 1998. 15

[Cootes 1999a] Timothy F Cootes, C Beeston, Gareth J Edwards and Christopher J Taylor. *A unified framework for atlas matching using active appearance models*. In Information Processing in Medical Imaging, pages 322–333. Springer, 1999. vii, 16

[Cootes 1999b] Timothy F Cootes and Christopher J Taylor. *A mixture model for representing shape variation*. Image and Vision Computing, vol. 17, no. 8, pages 567–573, 1999. 38, 40

[Cootes 2001] Timothy F. Cootes, Gareth J. Edwards and Christopher J. Taylor. *Active appearance models*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pages 681–685, 2001. 7, 15, 38

[Couprie 2009] Camille Couprie, Leo Grady, Laurent Najman and Hugues Talbot. *Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest*. In IEEE 12th International Conference on Computer Vision, pages 731–738. IEEE, 2009. 22

[Cremers 2003] Daniel Cremers, Timo Kohlberger and Christoph Schnörr. *Shape statistics in kernel space for variational image segmentation*. Pattern Recognition, vol. 36, no. 9, pages 1929–1943, 2003. 40

[Cremers 2006a] D. Cremers, S.J. Osher and S. Soatto. *Kernel density estimation and intrinsic alignment for shape priors in level set segmentation*. International Journal of Computer Vision, vol. 69, no. 3, pages 335–351, 2006. 38, 69

[Cremers 2006b] Daniel Cremers and Leo Grady. *Statistical priors for efficient combinatorial optimization via graph cuts*. In European Conference on Computer Vision (ECCV), volume 3, pages 263–274. Springer, 2006. 45

[Cremers 2007] D. Cremers, M. Rousson and R. Deriche. *A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape*. International Journal of Computer Vision, vol. 72, no. 2, pages 195–215, 2007. 8, 38, 39

[Cui 2008] Jingyu Cui, Qiong Yang, Fang Wen, Qiying Wu, Changshui Zhang, Luc Van Gool and Xiaoou Tang. *Transductive object cutout*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2008. 21

[Das 2009] Piali Das, Olga Veksler, Vyacheslav Zavadsky and Yuri Boykov. *Semiautomatic segmentation with compact shape prior*. Image and Vision Computing, vol. 27, no. 1, pages 206–219, 2009. 23

[Davatzikos 2003] Christos Davatzikos, Xiaodong Tao and Dinggang Shen. *Hierarchical active shape models, using the wavelet transform*. Medical Imaging, IEEE Transactions on, vol. 22, no. 3, pages 414–423, 2003. 39, 40

[De Berg 2000] Mark De Berg, Marc Van Kreveld, Mark Overmars and Otfried Cheong Schwarzkopf. Computational geometry. Springer, 2000. 109

[De Bruijne 2003] Marleen De Bruijne, Bram Van Ginneken, Wiro J Niessen, Marco Loog and Max A Viergever. *Model-based segmentation of abdominal aortic aneurysms in CTA images*. In Medical Imaging, pages 1560–1571. International Society for Optics and Photonics, 2003. 15

[De Bruijne 2004] Marleen De Bruijne and Mads Nielsen. *Shape particle filtering for image segmentation*. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2004, pages 168–175. Springer, 2004. 15

[De Bruijne 2005] Marleen De Bruijne and Mads Nielsen. *Multi-object segmentation using shape particles*. In Information Processing in Medical Imaging, pages 762–773. Springer, 2005. 15

[Delong 2012] Andrew Delong, Anton Osokin, Hossam N Isack and Yuri Boykov. *Fast approximate energy minimization with label costs*. International Journal of Computer Vision, vol. 96, no. 1, pages 1–27, 2012. 28

[Donner 2006] Rene Donner, Michael Reiter, Georg Langs, Philipp Peloschek and Horst Bischof. *Fast active appearance model search using canonical correlation analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 10, pages 1690–1694, 2006. 17

[Donner 2010] René Donner, Georg Langs, Branislav Mičušik and Horst Bischof. *Generalized sparse MRF appearance models*. Image and Vision Computing, vol. 28, no. 6, pages 1031–1038, 2010. 26

[Dryden 1998] I.L Dryden and K.V Mardia. Statistical shape analysis. Wiley & Sons, 1998. 38

[Eskildsen 2012] Simon F Eskildsen, Pierrick Coupé, Vladimir Fonov, José V Manjón, Kelvin K Leung, Nicolas Guizard, Shafik N Wassef, Lasse Riis Østergaard and D Louis Collins. *BEaST: Brain extraction based on nonlocal segmentation technique*. NeuroImage, vol. 59, no. 3, pages 2362–2373, 2012. viii, 104

[Etyngier 2007] Patrick Etyngier, Florent Segonne and Renaud Keriven. *Shape priors using manifold learning techniques*. In IEEE International Conference on Computer Vision (ICCV), pages 1–8. IEEE, 2007. 38, 40

[Felzenszwalb 2005] P.F. Felzenszwalb. *Representation and detection of deformable shapes*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 2, pages 208–220, 2005. 25, 28

[Ford 1962] DR Ford and Delbert Ray Fulkerson. Flows in networks. Princeton university press, 1962. 29

[Freedman 2005] D. Freedman and T. Zhang. *Interactive graph cut based segmentation with shape priors*. In Computer Vision and Pattern Recognition, volume 1, pages 755–762. IEEE, 2005. 23, 25, 69

[Freund 1995] Yoav Freund and Robert E Schapire. *A desicion-theoretic generalization of on-line learning and an application to boosting*. In Computational learning theory, pages 23–37. Springer, 1995. 83

[Freund 1996] Yoav Freund, Robert E Schapire*et al.* *Experiments with a new boosting algorithm*. In International Conference on Machine Learning, volume 96, pages 148–156, 1996. 83

[Freund 2000] Yoav Freund and Robert E Schapire. *Additive logistic regression: A statistical view of boosting*. The Annals of Statistics, vol. 28, no. 2, pages 391–393, 2000. 83, 84, 85

[Funka-Lea 2006] Gareth Funka-Lea, Yuri Boykov, Charles Florin, M-P Jolly, Romain Moreau-Gobard, Rana Ramaraj and Daniel Rinck. *Automatic heart isolation for*

CT coronary visualization using graph-cuts. In IEEE International Symposium on Biomedical Imaging: Nano to Macro, pages 614–617. IEEE, 2006. 23

[Geman 1984] Stuart Geman and Donald Geman. *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, no. 6, pages 721–741, 1984. 27, 28

[Glocker 2008] Ben Glocker, Nikos Paragios, Nikos Komodakis, Georgios Tziritas and Nassir Navab. *Optical flow estimation with uncertainties through dynamic MRFs*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2008. 130

[Glocker 2009] Ben Glocker, Nikos Komodakis, Nassir Navab, Georgios Tziritas and Nikos Paragios. *Dense registration with deformation priors*. In Information processing in medical imaging, pages 540–551. Springer, 2009. 70

[Glocker 2010] Ben Glocker, T Hauke Heibel, Nassir Navab, Pushmeet Kohli and Carsten Rother. *Triangleflow: Optical flow with triangulation-based higher-order likelihoods*. In European Conference on Computer Vision, pages 272–285. Springer, 2010. 28

[Glocker 2011] Ben Glocker, Aristeidis Sotiras, Nikos Komodakis and Nikos Paragios. *Deformable Medical Image Registration: Setting the State of the Art with Discrete Methods**. Annual review of biomedical engineering, vol. 13, pages 219–244, 2011. 70

[Goldberg 1988] Andrew V Goldberg and Robert E Tarjan. *A new approach to the maximum-flow problem*. Journal of the ACM (JACM), vol. 35, no. 4, pages 921–940, 1988. 29

[Gower 1975] John C Gower. *Generalized procrustes analysis*. Psychometrika, vol. 40, no. 1, pages 33–51, 1975. 13

[Grady 2005] Leo Grady. *Multilabel random walker image segmentation using prior models*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 763–770. IEEE, 2005. 22

[Grady 2006] Leo Grady. *Random walks for image segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 11, pages 1768–1783, 2006. 22, 70, 97

[Grady 2008] Leo Grady and Ali Kemal Sinop. *Fast approximate random walker segmentation using eigenvector precomputation*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2008. 22

[Greig 1989] DM Greig, BT Porteous and Allan H Seheult. *Exact maximum a posteriori estimation for binary images*. Journal of the Royal Statistical Society. Series B (Methodological), pages 271–279, 1989. 28, 29

[Gross 2005] Ralph Gross, Simon Baker, Iain Matthews and Takeo Kanade. *Face recognition across pose and illumination*. In Handbook of Face Recognition, pages 193–216. Springer, 2005. 17

[Gross 2006] Ralph Gross, Iain Matthews and Simon Baker. *Active appearance models with occlusion*. Image and Vision Computing, vol. 24, no. 6, pages 593–604, 2006. 17

[Hammer 1984] Peter L Hammer, Pierre Hansen and Bruno Simeone. *Roof duality, complementation and persistency in quadratic 0–1 optimization*. Mathematical Programming, vol. 28, no. 2, pages 121–155, 1984. 30

[Han 2007] Ju Han and Kai-Kuang Ma. *Rotation-invariant and scale-invariant Gabor features for texture image retrieval*. Image and Vision Computing, vol. 25, no. 9, pages 1474–1481, 2007. 80

[Heimann 2009] Tobias Heimann, Hans-Peter Meinzer*et al.* *Statistical shape models for 3D medical image segmentation: A review*. Medical image analysis, vol. 13, no. 4, page 543, 2009. vii, 8, 14, 39

[Heitz 2009] Geremy Heitz, Gal Elidan, Benjamin Packer and Daphne Koller. *Shape-based object localization for descriptive classification*. International journal of computer vision, vol. 84, no. 1, pages 40–62, 2009. 106

[Hyvärinen 2000] Aapo Hyvärinen and Erkki Oja. *Independent component analysis: algorithms and applications*. Neural networks, vol. 13, no. 4, pages 411–430, 2000. 39

[Jain 1998] Anil K Jain, Yu Zhong and Marie-Pierre Dubuisson-Jolly. *Deformable template models: A review*. Signal processing, vol. 71, no. 2, pages 109–129, 1998. 8

[Jehan-Besson 2003] Stéphanie Jehan-Besson, Michel Barlaud and Gilles Aubert. *DREAM2S: Deformable regions driven by an eulerian accurate minimization method for image and video segmentation*. International Journal of Computer Vision, vol. 53, no. 1, pages 45–70, 2003. 12

[Jiao 2003] Feng Jiao, Stan Li, Heung-Yeung Shum and Dale Schuurmans. *Face alignment using statistical models and wavelet features*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages I–321. IEEE, 2003. 15

[Jolliffe 2002] Ian T Jolliffe. Principal component analysis. Springer, 2002. 38

[Juan 2006] Olivier Juan and Yuri Boykov. *Active graph cuts*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 1023–1029. IEEE, 2006. 30

[Kass 1988] M. Kass, A. Witkin and D. Terzopoulos. *Snakes: Active contour models*. International Journal of Computer Vision, vol. 1, no. 4, pages 321–331, 1988. 8, 69

[Kelemen 1999] András Kelemen, Gábor Székely and Guido Gerig. *Elastic model-based segmentation of 3-D neuroradiological data sets*. Medical Imaging, IEEE Transactions on, vol. 18, no. 10, pages 828–839, 1999. 39

[Kendall 1984] David G Kendall. *Shape manifolds, procrustean metrics, and complex projective spaces*. Bulletin of the London Mathematical Society, vol. 16, no. 2, pages 81–121, 1984. 44

[Kichenassamy 1995] Satyanad Kichenassamy, Arun Kumar, Peter Olver, Allen Tannenbaum and Anthony Yezzi. *Gradient flows and geometric active contour models*. In International Conference on Computer Vision, pages 810–815. IEEE, 1995. 11

[Kohli 2005] Pushmeet Kohli and Philip HS Torr. *Efficiently solving dynamic markov random fields using graph cuts*. In IEEE International Conference on Computer Vision, volume 2, pages 922–929. IEEE, 2005. 30

[Kohli 2009a] Pushmeet Kohli, M Pawan Kumar and Philip HS Torr. *P^3 & Beyond: Move Making Algorithms for Solving Higher Order Functions*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 9, pages 1645–1656, 2009. 28

[Kohli 2009b] Pushmeet Kohli, Philip HS Torr*et al.* *Robust higher order potentials for enforcing label consistency*. International Journal of Computer Vision, vol. 82, no. 3, pages 302–324, 2009. 28

[Kolmogorov 2004] Vladimir Kolmogorov and Ramin Zabin. *What energy functions can be minimized via graph cuts?* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 2, pages 147–159, 2004. 29

[Kolmogorov 2005] V. Kolmogorov and Y. Boykov. *What metrics can be approximated by geo-cuts, or global optimization of length/area and flux*. In ICCV, volume 1, pages 564–571. IEEE, 2005. vii, 21

[Kolmogorov 2006] V. Kolmogorov. *Convergent tree-reweighted message passing for energy minimization*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 10, pages 1568–1583, 2006. 31, 33, 106, 121

[Kolmogorov 2007] Vladimir Kolmogorov and Carsten Rother. *Minimizing nonsubmodular functions with graph cuts-a review*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 7, pages 1274–1279, 2007. 30

[Komodakis] Nikos Komodakis, Bo Xiang and Nikos Paragios. *A Framework for Efficient Structured Max-Margin Learning of High-Order MRF Models*. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[Komodakis 2007a] Nikos Komodakis, Nikos Paragios and Georgios Tziritas. *MRF optimization via dual decomposition: Message-passing revisited*. In IEEE International Conference on Computer Vision (ICCV), pages 1–8. IEEE, 2007. vii, 31, 34, 51, 54, 55, 94

[Komodakis 2007b] Nikos Komodakis and Georgios Tziritas. *Approximate labeling via graph cuts based on linear programming*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 8, pages 1436–1453, 2007. 32, 33

[Komodakis 2008a] Nikos Komodakis and Nikos Paragios. *Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles*. In European Conference on Computer Vision, pages 806–820. Springer, 2008. 31

[Komodakis 2008b] Nikos Komodakis, Georgios Tziritas and Nikos Paragios. *Performance vs computational efficiency for optimizing single and dynamic mrfs: Setting the state of the art with primal-dual strategies*. Computer Vision and Image Understanding, vol. 112, no. 1, pages 14–29, 2008. 30

[Komodakis 2011a] Nikos Komodakis. *Efficient training for pairwise or higher order CRFs via dual decomposition*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1841–1848. IEEE, 2011. 52

[Komodakis 2011b] Nikos Komodakis, Nikos Paragios and Georgios Tziritas. *MRF energy minimization and beyond via dual decomposition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 3, pages 531–552, 2011. 31, 34, 51, 54, 121

[Kumar 2003] Sanjiv Kumar and Martial Hebert. *Discriminative random fields: A discriminative framework for contextual interaction in classification*. In IEEE International Conference on Computer Vision, pages 1150–1157. IEEE, 2003. 28, 29

[Kumar 2004] M Pawan Kumar, Philip HS Torr and Andrew Zisserman. *Learning Layered Pictorial Structures from Video.* In ICVGIP, pages 158–164, 2004. 104

[Kumar 2005] M Pawan Kumar, PHS Ton and Andrew Zisserman. *Obj cut*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 18–25. IEEE, 2005. 104, 119

[Kumar 2006] Sanjiv Kumar and Martial Hebert. *Discriminative random fields*. International Journal of Computer Vision, vol. 68, no. 2, pages 179–201, 2006. 29

[Kwon 2008] Dongjin Kwon, Kyong Joon Lee, Il Dong Yun and Sang Uk Lee. *Nonrigid image registration using dynamic higher-order mrf model*. In European Conference on Computer Vision, pages 373–386. Springer, 2008. 28

[Lafferty 2001] John Lafferty, Andrew McCallum and Fernando CN Pereira. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. In International Conference on Machine Learning (ICML), 2001. 28

[Langs 2006] Georg Langs, Philipp Peloschek, Rene Donner, Michael Reiter and Horst Bischof. *Active feature models*. In International Conference on Pattern Recognition, volume 1, pages 417–420. IEEE, 2006. 15

[Latecki 1999] Longin Jan Latecki and Rolf Lakämper. *Convexity rule for shape decomposition based on discrete contour evolution*. Computer Vision and Image Understanding, vol. 73, no. 3, pages 441–454, 1999. 109

[Leclerc 1989] Yvan G Leclerc. *Constructing simple stable descriptions for image partitioning*. International journal of computer vision, vol. 3, no. 1, pages 73–102, 1989. 12

[Leventon 2000] M.E. Leventon, W.E.L. Grimson and O. Faugeras. *Statistical shape influence in geodesic active contours*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 316–323. IEEE, 2000. 39

[Levin 2006] Anat Levin and Yair Weiss. *Learning to combine bottom-up and top-down segmentation*. In European Conference on Computer Vision (ECCV), pages 581–594. Springer, 2006. 105

[Li 2004] Shuyu Li, Litao Zhu and Tianzi Jiang. *Active shape model segmentation using local edge structures and AdaBoost*. In Medical Imaging and Augmented Reality, pages 121–128. Springer, 2004. 15

[Li 2005] Yuanzhong Li and Wataru Ito. *Shape parameter optimization for adaboosted active shape model*. In IEEE International Conference on Computer Vision, volume 1, pages 251–258. IEEE, 2005. 15

[Malladi 1995] R. Malladi, J.A. Sethian and B.C. Vemuri. *Shape modeling with front propagation: A level set approach*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 2, pages 158–175, 1995. 11

[Manjunath 1996] Bangalore S Manjunath and Wei-Ying Ma. *Texture features for browsing and retrieval of image data*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 8, pages 837–842, 1996. 78

[Matthews 2004] Iain Matthews and Simon Baker. *Active appearance models revisited*. International Journal of Computer Vision, vol. 60, no. 2, pages 135–164, 2004. 17

[McInemey 1999] T McInemey and Demetri Terzopoulos. *Topology adaptive deformable surfaces for medical image volume segmentation*. IEEE Transactions on Medical Imaging, vol. 18, no. 10, pages 840–850, 1999. 11

[McInerney 1995] Tim McInerney and Demetri Terzopoulos. *A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis*. Computerized Medical Imaging and Graphics, vol. 19, no. 1, pages 69–83, 1995. vii, 10

[McInerney 1996] Tim McInerney and Demetri Terzopoulos. *Deformable models in medical image analysis: a survey*. Medical Image Analysis, vol. 1, page 2, 1996. 8

[Metaxas 1993] Dimitris Metaxas and Demetri Terzopoulos. *Shape and nonrigid motion estimation through physics-based synthesis*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 6, pages 580–591, 1993. 38

[Montagnat 2001] Johan Montagnat, Hervé Delingette and Nicholas Ayache. *A review of deformable surfaces: topology, geometry and deformation*. Image and vision computing, vol. 19, no. 14, pages 1023–1040, 2001. 8, 11

[Mumford 1989] David Mumford and Jayant Shah. *Optimal approximations by piecewise smooth functions and associated variational problems*. Communications on pure and applied mathematics, vol. 42, no. 5, pages 577–685, 1989. 7, 12, 19

[Nain 2007] Delphine Nain, Steven Haker, Aaron Bobick and Allen Tannenbaum. *Multiscale 3-d shape representation and segmentation using spherical wavelets*. IEEE Transactions on Medical Imaging, vol. 26, no. 4, pages 598–618, 2007. 39

[Nowozin 2009] Sebastian Nowozin and Christoph H Lampert. *Global connectivity potentials for random field models*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 818–825. IEEE, 2009. 28

[Osher 1988] Stanley Osher and James A Sethian. *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations*. Journal of Computational Physics, vol. 79, no. 1, pages 12–49, 1988. 11

[Osher 2003] Stanley Osher and Nikos Paragios. Geometric level set methods in imaging, vision, and graphics. Springer, 2003. 11

[Packer 2010] Ben Packer, Stephen Gould and Daphne Koller. *A unified contour-pixel model for figure-ground segmentation*. In European Conference on Computer Vision (ECCV), pages 338–351. Springer, 2010. 106

[Paragios 2002] N. Paragios and R. Deriche. *Geodesic active regions and level set methods for supervised texture segmentation*. International Journal of Computer Vision, vol. 46, no. 3, pages 223–247, 2002. 7, 12

[Pizer 1999] Stephen M. Pizer, Daniel S. Fritsch, Paul A. Yushkevich, Valen E. Johnson and Edward L. Chaney. *Segmentation, registration, and measurement of shape variation via image object shape*. IEEE Transactions on Medical Imaging, vol. 18, no. 10, pages 851–865, 1999. 38

[Pohl 2006] Kilian M Pohl, John Fisher, Martha Shenton, Robert W McCarley, W Eric L Grimson, Ron Kikinis and William M Wells. *Logarithm odds maps for shape representation*. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), pages 955–963. Springer, 2006. 39

[Rohlfing 2005] Torsten Rohlfing, Robert Brandt, Randolf Menzel, Daniel B Russakoff and Calvin R Maurer Jr. *Quo vadis, atlas-based segmentation?* In Handbook of Biomedical Image Analysis, pages 435–486. Springer, 2005. 8

[Roth 2005] Stefan Roth and Michael J Black. *Fields of experts: A framework for learning image priors*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 2, pages 860–867. IEEE, 2005. 28

[Roth 2009] Stefan Roth and Michael J Black. *Fields of experts*. International Journal of Computer Vision, vol. 82, no. 2, pages 205–229, 2009. 28

[Rother 2004] Carsten Rother, Vladimir Kolmogorov and Andrew Blake. *Grabcut: Interactive foreground extraction using iterated graph cuts*. In ACM Transactions on Graphics (TOG), volume 23, pages 309–314. ACM, 2004. 20, 28, 29

[Rother 2007] Carsten Rother, Vladimir Kolmogorov, Victor Lempitsky and Martin Szummer. *Optimizing binary MRFs via extended roof duality*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2007. 30

[Rousson 2002] M. Rousson and N. Paragios. *Shape priors for level set representations*. pages 416–418. Springer, 2002. 38, 39

[Rousson 2008] Mikael Rousson and Nikos Paragios. *Prior knowledge, level set representations & visual grouping*. International Journal of Computer Vision, vol. 76, no. 3, pages 231–243, 2008. 69

[Schapire 1999] Robert E Schapire and Yoram Singer. *Improved boosting algorithms using confidence-rated predictions*. Machine learning, vol. 37, no. 3, pages 297–336, 1999. 83

[Schoenemann 2007] T. Schoenemann and D. Cremers. *Globally optimal image segmentation with an elastic shape prior*. In ICCV, pages 1–6. IEEE, 2007. 24

[Schölkopf 1998] Bernhard Schölkopf, Alexander Smola and Klaus-Robert Müller. *Nonlinear component analysis as a kernel eigenvalue problem*. Neural computation, vol. 10, no. 5, pages 1299–1319, 1998. 40

[Schraudolph 2010] Nic Schraudolph. *Polynomial-time exact inference in np-hard binary MRFs via reweighted perfect matching*. In International Conference on Artificial Intelligence and Statistics, pages 717–724, 2010. 31

[Seghers 2008] Dieter Seghers, Jeroen Hermans, Dirk Loeckx, Frederik Maes, Dirk Vandermeulen and Paul Suetens. *Model-based segmentation using graph representations*. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), pages 393–400. Springer, 2008. 25, 41, 45, 70

[Sethian 1999] James Albert Sethian. Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, volume 3. Cambridge University Press, 1999. 11

[Sharon 2006] Eitan Sharon, Meirav Galun, Dahlia Sharon, Ronen Basri and Achi Brandt. *Hierarchy and adaptivity in segmenting visual scenes*. Nature, vol. 442, no. 7104, pages 810–813, 2006. 19

[Shi 2000] Jianbo Shi and Jitendra Malik. *Normalized cuts and image segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pages 888–905, 2000. 7, 18

[Shlezinger 1976] MI Shlezinger. *Syntactic analysis of two-dimensional visual signals in the presence of noise*. Cybernetics and Systems Analysis, vol. 12, no. 4, pages 612–628, 1976. 30

[Shotton 2006] Jamie Shotton, John Winn, Carsten Rother and Antonio Criminisi. *Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation*. In European Conference on Computer Vision (ECCV), pages 1–15. Springer, 2006. 116

[Siddiqi 1998] Kaleem Siddiqi, Yves Bérubé Lauziere, Allen Tannenbaum and Steven W Zucker. *Area and length minimizing flows for shape segmentation*. IEEE Transactions on Image Processing, vol. 7, no. 3, pages 433–443, 1998. vii, 12

[Sigal 2006] Leonid Sigal and Michael J Black. *Measure locally, reason globally: Occlusion-sensitive articulated pose estimation*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 2, pages 2041–2048. IEEE, 2006. 28

[Singaraju 2009] Dheeraj Singaraju, Leo Grady and René Vidal. *P-brush: Continuous valued MRFs with normed pairwise distributions for image segmentation*. In

IEEE Conference on Computer Vision and Pattern Recognition, pages 1303–1310. IEEE, 2009. 22

[Sinop 2007] Ali Kemal Sinop and Leo Grady. *A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm*. In IEEE International Conference on Computer Vision, pages 1–8. IEEE, 2007. 22

[Sjostrand 2007] K Sjostrand, Egill Rostrup, Charlotte Ryberg, Rasmus Larsen, Colin Studholme, Hansjoerg Baezner, Jose Ferro, Franz Fazekas, Leonardo Pantoni, Domenico Inzitari*et al.* *Sparse decomposition and modeling of anatomical shape variation*. IEEE Transactions on Medical Imaging, vol. 26, no. 12, pages 1625–1635, 2007. 39

[Slabaugh 2005] Greg Slabaugh and Gozde Unal. *Graph cuts segmentation using an elliptical shape prior*. In IEEE International Conference on Image Processing, volume 2, pages II–1222. IEEE, 2005. 22, 24

[Sontag 2007] David Sontag and Tommi S Jaakkola. *New outer bounds on the marginal polytope*. In Advances in Neural Information Processing Systems, pages 1393–1400, 2007. 31

[Staib 1996] Lawrence H Staib and James S Duncan. *Model-based deformable surface finding for medical images*. IEEE Transactions on Medical Imaging, vol. 15, no. 5, pages 720–731, 1996. 8, 38, 39

[Stegmann 2006] Mikkel B Stegmann, Karl Sjöstrand and Rasmus Larsen. *Sparse modeling of landmark and texture variability using the orthomax criterion*. In Medical Imaging, pages 61441G–61441G. International Society for Optics and Photonics, 2006. 39

[Székely 1996] Gábor Székely, András Kelemen, Christian Brechbühler and Guido Gerig. *Segmentation of 2-D and 3-D objects from MRI volume data using constrained elastic deformations of flexible Fourier contour and surface models*. Medical Image Analysis, vol. 1, no. 1, pages 19–34, 1996. 38

[Taskar 2004] Ben Taskar, Carlos Guestrin and Daphne Koller. *Max-margin Markov networks*. In Advances in Neural Information Processing Systems, volume 16, page 25. MIT Press, 2004. 52

[Tresadern 2009] Philip A Tresadern, Harish Bhaskar, Steve A Adeshina, Christopher J Taylor and Timothy F Cootes. *Combining Local and Global Shape Models for*

Deformable Object Matching. In British Machine Vision Conference, volume 9, pages 451–458, 2009. 15

[Tsai 2001a] Andy Tsai, Anthony Yezzi Jr, William Wells III, Clare Tempany, Dewey Tucker, Ayres Fan, W Eric Grimson and Alan Willsky. *Model-based curve evolution technique for image segmentation*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages I–463. IEEE, 2001. 39

[Tsai 2001b] Andy Tsai, Anthony Yezzi Jr and Alan S Willsky. *Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification*. IEEE Transactions on Image Processing, vol. 10, no. 8, pages 1169–1186, 2001. 12

[Twining 2001] Carole J Twining and Christopher J Taylor. *Kernel principal component analysis and the construction of non-linear active shape models*. In British Machine Vision Conference (BMVC), volume 1, pages 23–32, 2001. 40

[Van Ginneken 2002] Bram Van Ginneken, Alejandro F Frangi, Joes J Staal, Bart M ter Haar Romeny and Max A Viergever. *Active shape model segmentation with optimal features*. IEEE Transactions on medical Imaging, vol. 21, no. 8, pages 924–933, 2002. 15

[Veksler 2008] Olga Veksler. *Star shape prior for graph-cut image segmentation*. In European Conference on Computer Vision, pages 454–467. Springer, 2008. 23

[Vicente 2008] Sara Vicente, Vladimir Kolmogorov and Carsten Rother. *Graph cut based image segmentation with connectivity priors*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2008. 21, 28

[Vu 2008] Nhat Vu and BS Manjunath. *Shape prior segmentation of multiple objects with graph cuts*. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2008. 24

[Wainwright 2005] Martin J Wainwright, Tommi S Jaakkola and Alan S Willsky. *MAP estimation via agreement on trees: message-passing and linear programming*. IEEE Transactions on Information Theory, vol. 51, no. 11, pages 3697–3717, 2005. 31, 33

[Wang 2009] Chaohui Wang, Martin de La Gorce and Nikos Paragios. *Segmentation, ordering and multi-object tracking using graphical models*. In International Conference on Computer Vision (ICCV), pages 747–754. IEEE, 2009. 105

[Wang 2010] Chaohui Wang, Olivier Teboul, Fabrice Michel, Salma Essafi and Nikos Paragios. *3D knowledge-based segmentation using pose-invariant higher-order graphs*. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), pages 189–196. Springer, 2010. 41, 70, 94

[Werner 2007] Tomas Werner. *A linear programming approach to max-sum problem: A review*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 7, pages 1165–1179, 2007. 31

[Xiang 2011] Bo Xiang, Chaohui Wang, J-F Deux, Alain Rahmouni and Nikos Paragios. *Tagged cardiac MR image segmentation using boundary & regional-support and graph-based deformable priors*. In IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI), pages 1706–1711. IEEE, 2011.

[Xiang 2012] Bo Xiang, Chaohui Wang, J-F Deux, Alain Rahmouni and Nikos Paragios. *3D cardiac segmentation with pose-invariant higher-order MRFs*. In IEEE International Symposium on Biomedical Imaging (ISBI), pages 1425–1428, 2012. 97

[Xiang 2013a] Bo Xiang, J-F Deux, Alain Rahmouni and Nikos Paragios. *Joint model-pixel segmentation with pose-invariant deformable graph-priors*. In Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, 2013.

[Xiang 2013b] Bo Xiang, Nikos Komodakis and Nikos Paragios. *Pose invariant deformable shape priors using L1 higher order sparse graphs*. In International Symposium on Visual Computing (ISVC). Springer, 2013.

[Xu 1998] Chenyang Xu and Jerry L Prince. *Snakes, shapes, and gradient vector flow*. IEEE Transactions on Image Processing, vol. 7, no. 3, pages 359–369, 1998. vii, 9, 10

[Yezzi Jr 1997] Anthony Yezzi Jr, Satyanad Kichenassamy, Arun Kumar, Peter Olver and Allen Tannenbaum. *A geometric snake model for segmentation of medical imagery*. IEEE Transactions on Medical Imaging, vol. 16, no. 2, pages 199–209, 1997. 11

[Yu 2007] Peng Yu, P Ellen Grant, Yuan Qi, Xiao Han, Florent Ségonne, Rudolph Pienaar, Evelina Busa, Jenni Pacheco, Nikos Makris, Randy L Buckner*et al. Cortical surface shape analysis based on spherical wavelets*. IEEE Transactions on Medical Imaging, vol. 26, no. 4, pages 582–597, 2007. 39

[Zhan 2003] Yiqiang Zhan and Dinggang Shen. *Automated segmentation of 3D US prostate images using statistical texture-based matching method*. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), pages 688–696. Springer, 2003. 80

[Zhang 2004] Jiayong Zhang, Robert Collins and Yanxi Liu. *Representation and matching of articulated shapes*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 2, pages II–342. IEEE, 2004. 25

[Zheng 2008] Yefeng Zheng, Adrian Barbu, Bogdan Georgescu, Michael Scheuering and Dorin Comaniciu. *Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features*. IEEE Transactions on Medical Imaging, vol. 27, no. 11, pages 1668–1681, 2008. vii, 2

[Zikic 2010] Darko Zikic, Ben Glocker, Oliver Kutter, Martin Groher, Nikos Komodakis, Ali Kamen, Nikos Paragios and Nassir Navab. *Linear intensity-based image registration by Markov random fields and discrete optimization*. Medical image analysis, vol. 14, no. 4, pages 550–562, 2010. 70