Model-Based Organ Segmentation in CT Scans

Jiun-Hung Chen

General Exam/ Thesis Proposal

University of Washington

2009

Program Authorized to Offer Degree: UW Computer Science and Engineering

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Motivation

Organ segmentation in three-dimensional (3D) CT scans is the first step for any further medical analysis such as computer-aided detection, computer-aided diagnosis, radiation treatment, or organ shape analysis. Figure 1.1 shows a typical liver segmentation example on seven CT slices that are uniformly sampled from top to down from a abdominal CT volume. If the CT scan starts from the chest as the example shown in Figure 1.1., the liver is not visible in the first slices. In addition, from Figure 1.1, it is easy to see how the size and the shape of the liver in different slices changes dramatically.

Although manual organ segmentation can be accurate, it is very time-consuming. Therefore, an automatic or semi-automatic organ segmentation method that is efficient and effective is highly desired. However, such a system is hard to design for the following four reasons. First, there are many different shape variations in organs, especially in livers. For example, Figure 1.2 shows a set of 20 livers, some of which have very different shapes. Second, the soft tissues and the neighboring organs around an organ of interest can look very similar to the organ itself in CT scans. Third, some disease symptoms (e.g., tumors) or contrast materials may greatly change the appearance or visibility of an organ in CT scans. Fourth, a patient's posture or movement can influence the rotation and position of the organs of interest in his CT scans.

1.2 Problem Statement

A CT slice or image is a gray tone intensity image that forms one plane of a full 3D CT volume. A corresponding ground truth slice or image is a binary image whose pixels have value 1 in the location where the organ of interest appears in the CT slice and 0 elsewhere. Let $\mathfrak{I}$ and $\mathfrak{L}$ denote the space of all possible CT slices and the space of all possible ground

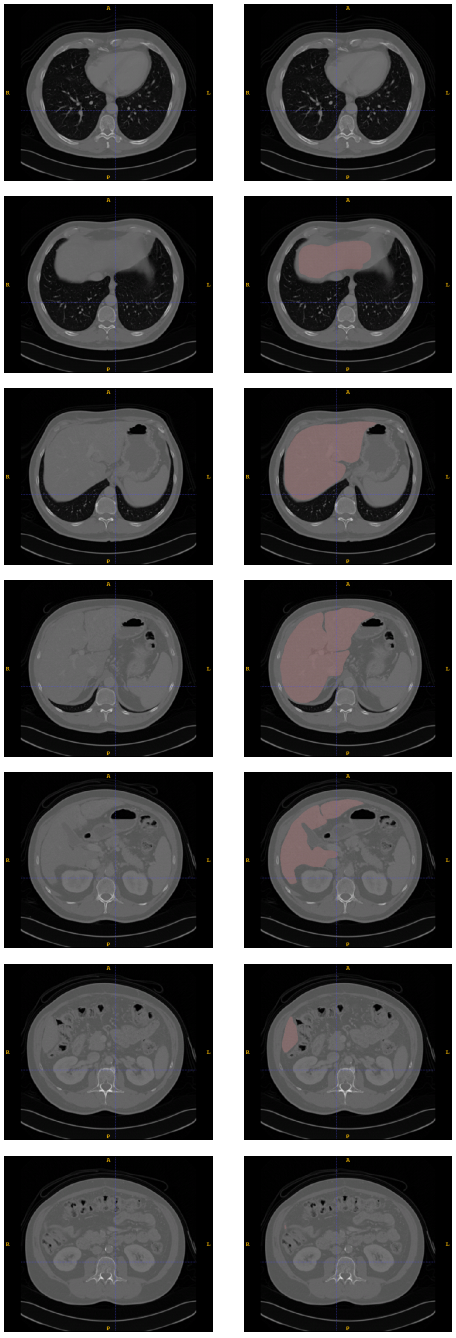


Figure 1.1: CT scans (axial view) and liver segmentation. The first column shows the CT scans and the second column shows the corresponding ground truth liver segmentation.

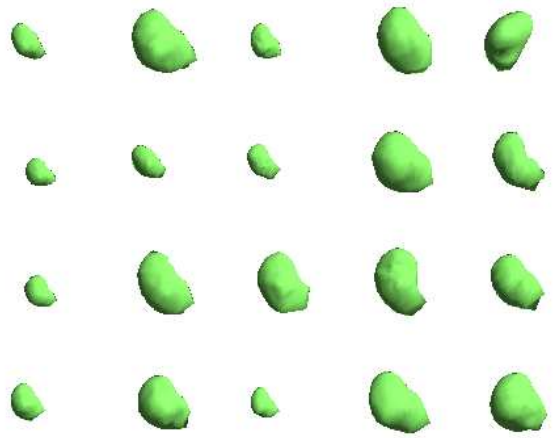


Figure 1.2: A set of 20 3D liver shapes.

truth slices, respectively. Let $\Upsilon_t = \{(\mathfrak{I}_i^t, \mathfrak{L}_i^t) | i = 1, \ldots, \mathfrak{N}\}$ denote a set of training image pairs where each $\mathfrak{I}_i^t \in \mathfrak{I}$ is a CT slice, and $\mathfrak{L}_i^t \in \mathfrak{L}$ is its corresponding ground truth slice. Similarly, let $\Upsilon_s = \{(\mathfrak{I}_i^s, \mathfrak{L}_i^s) | i = 1, \ldots, \mathfrak{K}\}$ denote a set of testing image pairs in which $\mathfrak{I}_i^s \in \mathfrak{I}$ are CT slices from the similar positions of the body as Υ_t and $\mathfrak{L}_i^s \in \mathfrak{L}$ is the correct ground truth for $\mathfrak{I}_i^s$. The problem of organ segmentation is to use the information in Υ_t to learn how to segment CT images in Υ_s. More formally, given the training set Υ_t, construct a classifier $\mathfrak{C} : \mathfrak{I} \rightarrow \mathfrak{L}$ that when applied to Υ_s, minimizes the following error function.

$$\sum_{i=1}^{\mathfrak{K}} \mathfrak{E}(\mathfrak{C}(\mathfrak{I}_i^s), \mathfrak{L}_i^s) \tag{1.1}$$

where $\mathfrak{E}(.,.)$ denotes a dissimilarity measure that measures the differences between estimated segmentations and ground truth segmentations. In this work, we will pose the segmentation problem as an optimization problem that not only minimizes an error function that measures the difference between the predicted label images and ground truth label images over the training set, but also can generalize to unseen testing CT images. The above problem

statement and formulation can be easily extended to deal with a training set of several full 3D CT volumes.

1.3 Paper Outline

The paper is organized as follows. We review the previous work in Chapter 2. In Chapter 3, we present our current method and report the primitive results. The proposal is presented in Chapter 4, and we conclude in Chapter 5.

Chapter 2

PREVIOUS WORK

In this chapter, we review previous work. Our review focuses on the four main methods and their applications in organ segmentations. Active shape models and active contour and level set methods are reviewed in Section 2.1 and in Section 2.2, respectively. Graph cuts are the focus of Section 2.3 and adaptive thresholding, morphology and pixel/voxel classifications are reviewed in Section 2.4.

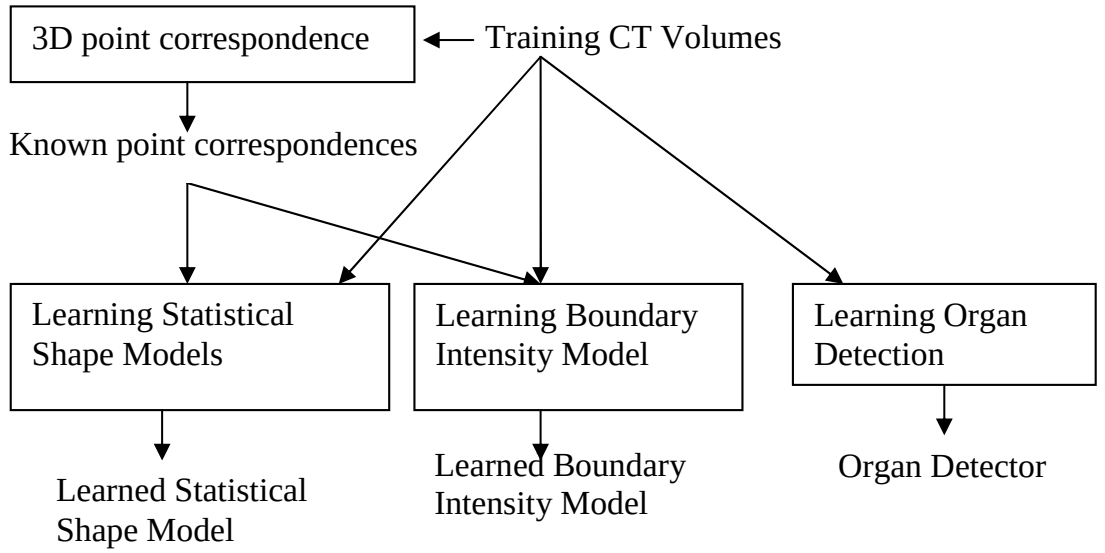
2.1 Active Shape Models

2.1.1 Framework

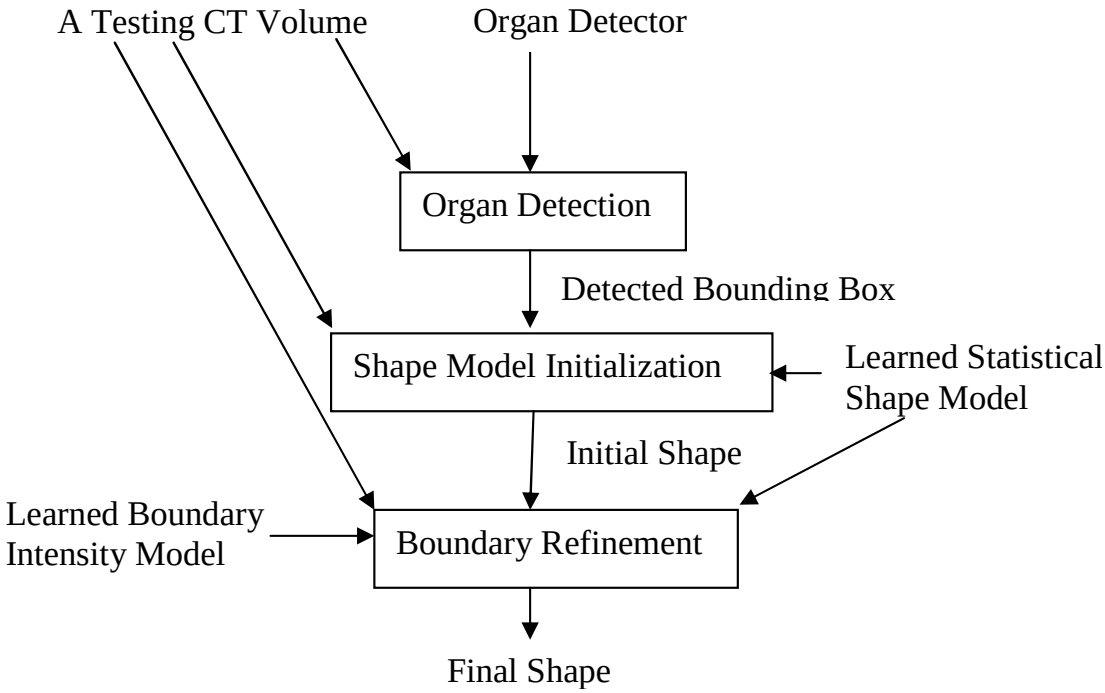
The framework used in this paper employs active shape models [104], which are widely used to model shapes of organs in medical image segmentation. We use Figure 2.1 to describe the key ideas and steps in our approach.

We assume that a shape is represented by a set of vertices in a 3D mesh consisting of vertices and edges, and there is a set of training CT volumes whose ground truth segmentations are known. Figure 2.2 shows a conceptual example of simplified elliptical training shapes. In the training phase, the goal is to learn a statistical shape model to model a particular organ, a boundary intensity model for each vertex point and an organ detector to detect a particular organ. Before learning the first two models, we need to find 3D point correspondences among a set of training shapes. Figure 2.1(a) shows the training process. There are four main steps in the training phase: finding 3D point correspondences, learning a statistical shape model for a particular organ, learning a boundary intensity model for each vertex point and learning an organ detector to detect a particular organ.

- Step 1. Finding 3D point correspondences: Each point on a mesh corresponds to a point on each of the other meshes. This step finds the correspondences for all points



(a) Training phase.



(b) Testing phase.

Figure 2.1: Training and testing phases for active shape models.

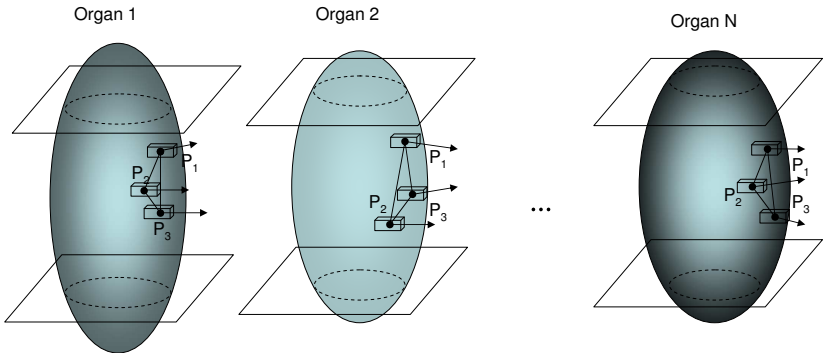


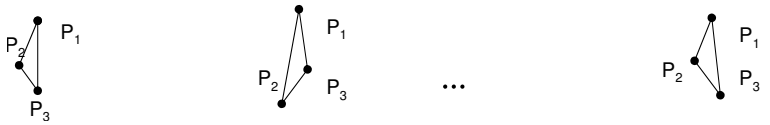
Figure 2.2: A set of training CT volumes. The colors represent the intensity values in the CT volumes. The shape of an organ is represented by a triangular mesh model. A triangular mesh that contains three vertex points, P_1, P_2 and P_3, is drawn in each organ. For each vertex point, an arrow is drawn to represent its normal vector and a 3D box represents its intensity profile. We use different sizes and colors to indicate the variations in sizes and intensities of organs.

on each mesh. Figure 2.3 shows an example of finding 3D point correspondences for simple three-point triangular meshes extracted from Figure 2.2. More detailed explanations and reviews on this step are in Section 2.1.2.

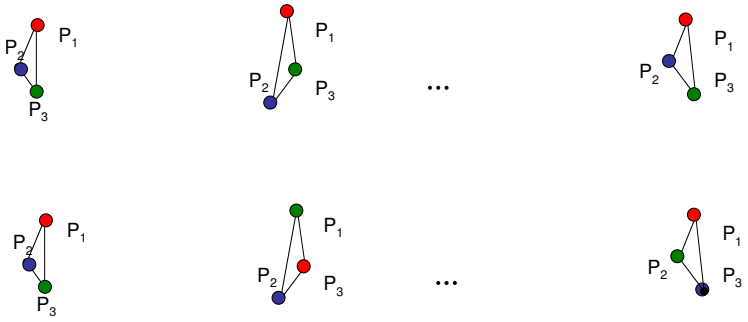
- Step 2. Learning a statistical shape model. A statistical shape model is learned from the training set of shapes whose 3D point correspondences are known. Figure 2.4 illustrates this concept. The learned statistical shape model will be used in the testing phase to guide the segmentation. Different statistical models will be reviewed in Section 2.1.3.

- Step 3. Learning a boundary intensity model for each vertex point. Figure 2.5 illustrates this concept. The goal is to learn a model that decides whether a pixel is on a boundary point. The learned model will be used in the testing phase for the boundary refinement process. Different boundary intensity models will be reviewed in Section 2.1.4.

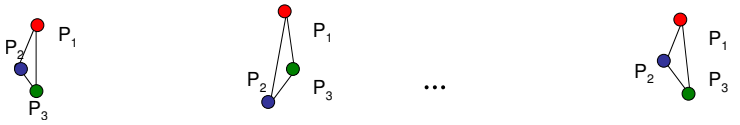
- Step 4. Learning an organ detector. The goal is to learn an algorithm that can find the bounding box including the organ in 3D CT volume as closely as possible. Different organ detection algorithms will be reviewed in Section 2.1.5.



(a) Input shapes whose 3D point correspondences are unknown.



(b) Possible 3D point correspondences.

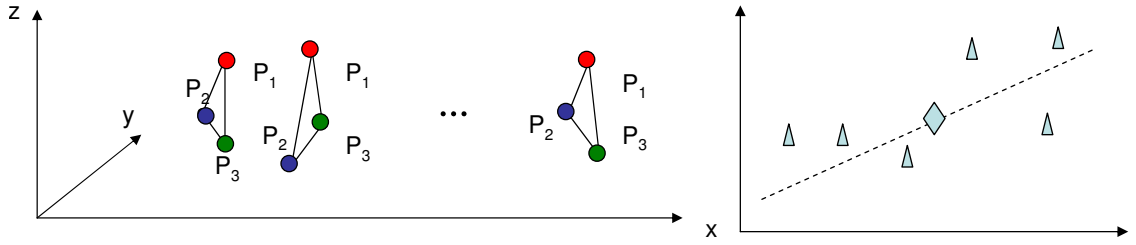


(c) Correct 3D point correspondences.

Figure 2.3: 3D point correspondence. Points in the same color indicate they are corresponding points.

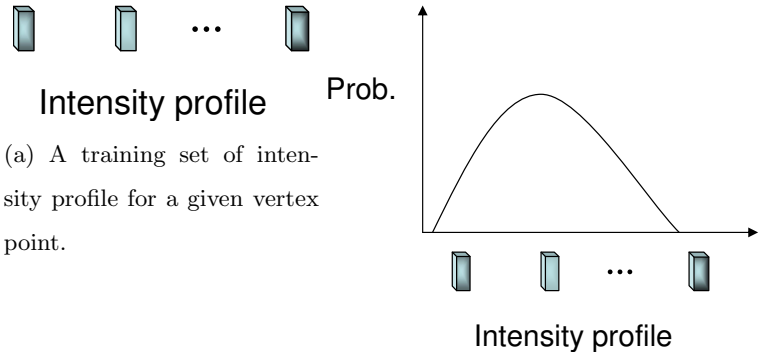
Given a testing CT volume (e.g., Figure 2.6(a)), we want to compute an organ segmentation based on the information learned in the training phase. There are three main steps in the testing phase, organ detection, shape model initialization and boundary refinement, as shown in Figure 2.1(b).

- Step 1. Organ Detection and Shape Model Initialization.
  - Step 1.1 Organ Detection: Use the learned organ detector to detect the organ in



(a) The training shapes whose 3D point correspondences are known. (b) A learned statistical shape model.

Figure 2.4: Learning a statistical shape model. A triangle represents a training shape and the mean shape of all the training shapes is drawn with a diamond shape. Here we assume that we learn a linear model (e.g., the dotted line) to represent the training shapes.



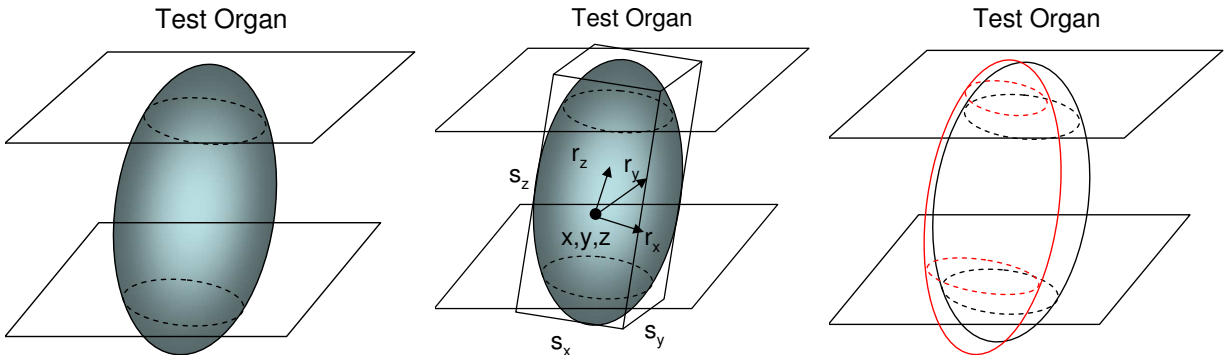
(a) A training set of intensity profile for a given vertex point.

(b) A learned boundary intensity model.

Figure 2.5: Learning a boundary intensity model. A simple histogram model indicates the probability of a point belonging to the boundary given its intensity profile.

the testing CT volumes and return a bounding box that contains the detected organ (see Figure 2.6 for example).

– Step 1.2. Shape Model Initialization. The learned statistical shape model is initialized based on the detected bounding box. The detection results from Step 1 are used to project the 3D model into the 3D CT scan, resulting in an initial boundary.



(a) A testing CT volume.

(b) A bounding box that contains the detected organ.

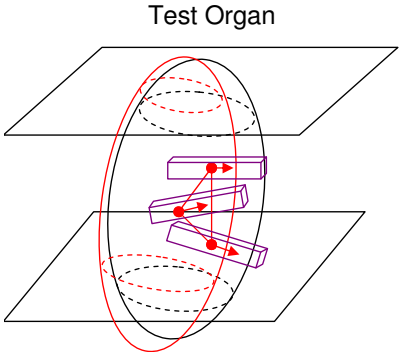
(c) The learned statistical shape model is initialized based on the detected bounding box to obtain the initial shape hypothesis.

Figure 2.6: Organ Detection and Shape Model Initialization. The bounding box that contains the detected organ has 9D parameters, the position (x, y, z), the orientation, (r_x, r_y, r_z) and the scale (s_x, s_y, s_y). The initialized shape model is drawn in red, while the unknown organ model is drawn in black.

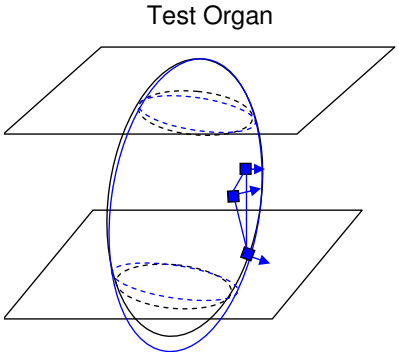
In addition to organ detection, shape model initialization will be reviewed in Section 2.1.5.

- Step 2. Boundary refinement (Figure 2.7). Typically, a two step procedure is iterated. First, the learned boundary intensity model is used to estimate the refinement, which involves estimating a displacement vector for each vertex point in the neighborhood of the current vertex point (Figures 2.7(a)-2.7(b)). Second, principal component subspace projection is performed to find the closest approximation to the addition of the current shape plus the current estimated displacement. (Figures 2.7(c)-2.7(d)). Boundary refinement will be reviewed in Section 2.1.6.

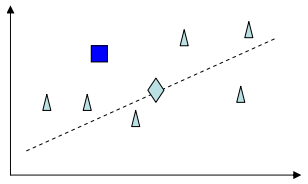
Thus, there are four main subproblems in the training phase for active shape models. First, 3D point correspondences among a set of training shapes represented by a set of vertices are found. Second, after the correspondence problem is solved, a statistical shape



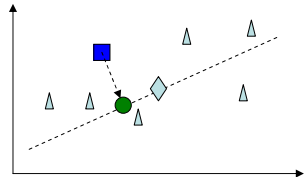
(a) An initial shape.



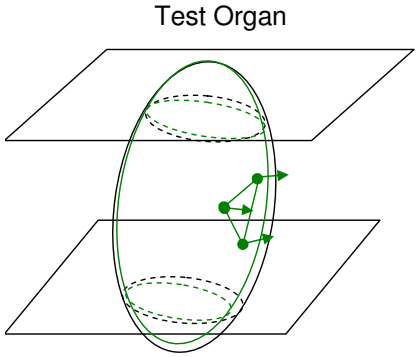
(b) The updated shape results from the addition of the displacement vector to the initial shape.



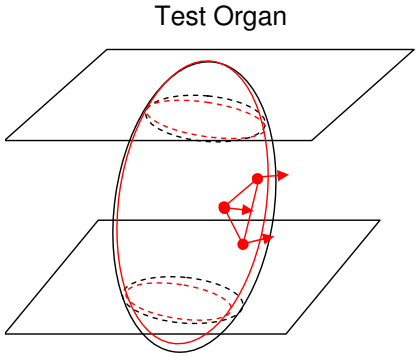
(c) The updated shape projected in the shape space.



(d) Find the closest point on the statistical shape model to the updated shape.



(e) Place the (closest) shape model in the CT volume.



(f) The initial shape is replaced with the (closest) shape and the process (steps a-f) is repeated.

Figure 2.7: Boundary refinement.(a) For a vertex point, we draw its normal vector as a red arrow and the search range as a violet box. The learned boundary intensity model is used to estimate the refinement (e.g., estimating a displacement vector for each vertex point in the search range or the violet box of the current vertex point).(b) After the displacement for each vertex point is estimated, we can get an updated shape. (c) The updated shape is shown in the shape space as a blue square. (d) The closest shape to the updated shape (the blue square) on the statistical shape model (the dotted line) is drawn as the green circle. (e) The closest shape is put in the CT volume. (f) The initial shape is replaced with the (closest) shape and the process (steps a-f) is repeated.

model is learned from the set of training shapes. Third, a statistical intensity model of the boundaries is learned. Fourth, an organ detector is learned. In the testing phase, after the organ is detected, the parameters of the learned model are initialized. Then, the boundary of the mesh is refined until a set of convergence conditions are met. We will discuss each subproblem in details in the following subsections.

2.1.2 3D Point Correspondence Problem

Finding point correspondence is essential for automatically building statistical shape models [35] from a training set of 3D surfaces. Statistical shape models are widely used in model-based image segmentation and tracking [35]. Davies *et al.* [34] assumed the projected coefficients have multivariate Gaussian distributions and derived an objective function for point correspondence problems that uses minimum description length (MDL) to balance the training errors and generalization ability. A recent evaluation study [102] that compares several well known 3D point correspondence methods for modeling purposes shows that the MDL-based approach [34] is the best method.

Assume that we have a training set of N 3D shapes and each shape is represented by M 3D landmarks points. Conventionally, we can represent each such shape by a vector $\mathbf{x}$ whose dimension is $3M \times 1$ and let $\Omega_t = \{\mathbf{x}_i | i = 1, \ldots, N\}$ denote the training set of N 3D shapes. Before we describe the MDL-based approach [34], we review principal component analysis (PCA), which will be used in this approach.

2.1.2.1 Principal Component Analysis

PCA is a common approach used to model the shape variations of a given training set of 3D shapes. Given Ω_t, the total scatter matrix $\mathbf{S}$ is defined as

$$\mathbf{S} = \sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})^t (\mathbf{x}_i - \bar{\mathbf{x}}) \tag{2.1}$$

where $\bar{\mathbf{x}}$ is the mean shape vector

$$\bar{\mathbf{x}} = \frac{\sum_{i=1}^{N} \mathbf{x}_i}{N} \tag{2.2}$$

PCA finds a projection axis $\mathbf{b}$ that maximizes $\mathbf{b^t S b}$. Intuitively, the total scatter of the projected samples is maximized after the projection of the samples onto $\mathbf{b}$. The optimal $\mathscr{K}$ projection axes $\mathbf{b}_k, k = 1, \ldots, \mathscr{K}$ that maximize the above criterion are the eigenvectors of $\mathbf{S}$ corresponding to the largest $\mathscr{K}$ eigenvalues, $\{\lambda_k \mid k = 1, \ldots, \mathscr{K}\}$. The reconstruction $\tilde{\mathbf{x}}$ of shape vector $\mathbf{x}$ can be used to approximate it.

$$\tilde{\mathbf{x}} = \bar{\mathbf{x}} + \sum_{k=1}^{\mathscr{K}} c_k \mathbf{b}_k \tag{2.3}$$

where $c_k = (\mathbf{x} - \bar{\mathbf{x}})^t \mathbf{b}_k$.

2.1.2.2 Correspondence by Minimizing Description Length

Davies *et al.* [34] proposed a MDL-based objective function to quantize the quality of the correspondence. In this paper, we focus on the simplified version of the MDL, proposed by Thodberg [105] as defined below.

$$F = \sum_{k=1}^{N} L_k \text{ with } L_k = \begin{cases} 1 + \log(\lambda_k / \lambda_{cut}), & \text{if } \lambda_k \geq \lambda_{cut} \\ \lambda_k / \lambda_{cut}, & \text{otherwise} \end{cases} \tag{2.4}$$

If 3D point correspondences among a set of shapes are estimated, then F can be used to quantize the quality of the correspondences. The eigenvalues, $\{\lambda_k | k = 1, \ldots, N\}$, used in calculating F are computed by performing PCA for the set of shapes with the estimated correspondences. λ_{cut} is a parameter that represents the expected noise in the training data, and its value is manually determined.

Given the above MDL-based objective function, an efficient method for manipulating correspondences and an optimization algorithm that minimizes the objective function are required in order to find optimal correspondences [35]. Typically, manipulating correspondences is treated as parameterizing and then re-parameterizing the surfaces. A parameterization assigns every point on the surface of the mesh to a unique point on the unit sphere, although parameterizations may not exist for arbitrary surfaces. In this paper, we assume that the 3D shapes are closed two-manifolds of genus 0 (e.g., a sphere). We use a conformal mapping as a parameterization and a reparameterization that modifies the parameterization based on kernels with strictly local effects, as developed in [51].

We assume that the parameterization of the i-th shape is controlled by some parameter vector α_i, for which the individual parameters are given by $\{\alpha_{i,a}|a = 1, \ldots, A\}$. The gradient descent approach is used to minimize F with respect to a parameter vector α_i. The Jacobian matrix for the gradient of the objective function is defined as

$$\frac{\partial F}{\partial \alpha_{i,a}} = \sum_{k=1}^{N} \frac{\partial L_k}{\partial \lambda_k} \frac{\partial \lambda_k}{\partial \alpha_{i,a}} \tag{2.5}$$

It is easy to compute $\frac{\partial L_k}{\partial \lambda_k}$ and so we focus on $\frac{\partial \lambda_k}{\partial \alpha_{i,a}}$ in the following discussions. $\frac{\partial \lambda_k}{\partial \alpha_{i,a}}$ can be computed by using the following chain rule for derivatives.

$$\frac{\partial \lambda_k}{\partial \alpha_{i,a}} = \frac{\partial \lambda_k}{\partial \mathbf{x}_i} \cdot \frac{\partial \mathbf{x}_i}{\partial \alpha_{i,a}} \tag{2.6}$$

While $\frac{\partial \mathbf{x}_i}{\partial \alpha_{i,a}}$ is typically computed by using finite differences, the following analytic form for $\frac{\partial \lambda_k}{\partial \mathbf{x}_i}$ exists.

$$\frac{\partial \lambda_k}{\partial x_i} = 2c_{i,k}\mathbf{b}_k \tag{2.7}$$

where $c_{i,k}$ is the projection coefficient of the i-th shape vector $\mathbf{x}_i$ onto the k-th eigenvector $\mathbf{b}_k$.

Finally, after the optimization is done, the correspondences that minimizes F can be easily extracted from the corresponding parameterizations.

2.1.3 Statistical Shape Models

After the 3D point correspondence problem has been for solved all models of a training set for a particular organ, a statistical shape model of that organ is learned from them. How to model shape variations plays an important role in active shape models [104] that is widely used in model-based medical image segmentation, and PCA is a common approach for this task. Recently, there have been many algorithms that apply kernel tricks for different linear algorithms and result in many corresponding nonlinear algorithms. For example, kernel principal component analysis (KPCA) [92] and kernel independent component analysis [6] are two representative examples of applying kernel tricks to principal component analysis and independent component analysis, respectively. The resultant nonlinear algorithms generally perform better than the original linear algorithms due to nonlinearity. For example,

[88][108] use KPCA in active shape models to model shapes and achieve better performances than PCA.

In contrast with using statistical shape models, Brinkley [19] uses geometric constraint networks as a representation to model biological objects. The assumption behind this representation is that networks of local interacting geometric constraints between structure subparts are able to represent the shape and the variations of the biological object in question. Specifically, in [20], Brinkely used a radial contour model that is an example of a geometric constraint network. In the radial contour model, the contours are controlled by the flexible lengths of radials from the centroid of the organ to uniformly-spaced points on the contour. The ratio between the lengths of two neighboring radials is used as a constraint for this pair. In other words, given the length of one of the neighboring radial, this ratio constraint constrains the length of the other radial. The ratios are estimated from training examples. The end points of the radials are computed by a relaxation algorithm. The algorithm uses prior knowledge about shapes of the organ in the form of the ratio constraints and edge information in the images to select the end points for each radial. After the end points are computed, the final contour is formed by connecting each end point. A 2D semi-automatic organ segmentation and matching system is implemented based on this approach [20].

2.1.4 Boundary Intensity Models

In addition to statistical shape models, a boundary intensity model for each landmark point is learned and is used for boundary refinement that will be explained in Section 2.1.5. Typically, the intensity profile along the normal vector at a landmark point is modeled. A common approach is to model the intensity profile as a Gaussian distribution [104]. Because the assumption that intensity profiles can be modeled as Gaussian distributions may not hold in real applications, Li and Ito [73] used AdaBoosted histogram classifiers [46] for modeling the intensity profiles and achieved better performance than that by using Gaussian distributions in face alignment. Kainmüller, Lange and Lamecker [57] used a heuristic intensity distribution to model liver boundaries.

2.1.5 Organ Detection and Shape Model Initialization

Given a CT scan, we need to detect the organ of interest and initialize the learned statistical shape model, so that the organ of the interest in the CT scan is aligned with the initialized shape model. There are two main types of organ detection methods, learning-based methods and heuristics-based methods. While learning-based methods use different machine learning algorithms to learn how to detect organs, heuristics-based methods use priors or heuristics to detect organs. For example, Ling et al. [74] use marginal space learning with 3D Haar features and steerable features to learn to detect livers. In contrast, Kainmüller, Lange and Lamecker [57] use heuristics about the lower rim of the right lobe of the lung to position the liver shape model below it. Campadelli, Casiraghi and Lombardi [23] detected livers by using heart segmentation information.

Typically, the statistical shape model is initialized to be the mean shape of all the training shapes whose correspondence problems are solved.

2.1.5.1 3D Organ Detection using Marginal Space Learning [124][125]

Learning-based object detectors achieve great performance in many different object detection applications, such as face detection [115] and pedestrian detection[116]. The basic idea of learning-based object detection is to train a classifier that can determines whether an input hypothesis is the hypothesis of interest and assign it a classification score, usually in the range [0 1]. The full parameter space is then quantized into a large number of hypotheses. Each hypothesis is tested with the learned classifier and the hypotheses with the best classification scores are selected as the output. However, as the dimension of parameter space increases, the number of hypotheses increases exponentially. One observation that marginal space learning (MSL) [124][125] or incremental parameter learning (IPL) [75] use to deal with this problem is that in many real applications the posterior distribution is clustered in a small region in the high-dimensional parameter space. For this reason, it is unnecessary to perform the uniform and exhaustive search in the high-dimensional parameter space. In MSL, the dimensionality of the search space is gradually increased. Consider the 3D organ detection problem: to find the bounding box including the organ as closely

as possible in a 3D CT volume. The parameter space is 9D (3D positions, 3D scales and 3D orientations). Position $\mathfrak{T}$ is the center of the bounding box of the organ, scale $\mathfrak{S}$ is its size, and orientation $\mathfrak{R}$ is in terms of the three Euler angles. The 3D organ detection problem is decomposed into three steps: position estimation, position-scale estimation and finally position-scale-orientation estimation. The high-level idea behind the MSL algorithm is given in Algorithms 1-2.

In testing, for every 3D object box candidate in the test CT volume, the position classifier, $\mathfrak{C}_{\mathfrak{T}}$, is first used to determine whether a 3D object box candidate contains an organ of interest. The top $\mathfrak{m}$ candidates are kept and augmented with $\mathfrak{n}$ scales. After this augmentation, we have a set of $\mathfrak{mn}$ 3D object box candidates. Then, the position-scale classifier, $\mathfrak{C}_{\mathfrak{S}}$, is used to discriminant the set of candidates and the best $\mathfrak{m}$ candidates are kept. Finally, the top $\mathfrak{m}$ candidates from the classification results generated by the position-scale classifier are kept and augmented with $\mathfrak{n}$ orientations. After this augmentation, we have a set of $\mathfrak{mn}$ 3D object box candidates. Then, the position-scale-orientation classifier, $\mathfrak{C}_{\mathfrak{R}}$, is used to discriminant the set of candidates and the best $\mathfrak{m}$ candidates are kept as the final output. Using MSL algorithm for testing is given in Algorithm 3.

MSL has used extensively to detect livers [74], heart chambers [124][125], Ileo-Cecal Valve [75] in CT volumes. Recently, Zheng et al. [126] further improved MSL by using better representation of a rotation matrix in which Euler angles are replaced with quaternions and constrained spaces for object position, orientation and scale.

2.1.6 Boundary Refinement

After the detection and initiations are done, the boundaries of the organ of interest in CT scans are refined iteratively as shown below.

- Step 1. For a given vertex point, refine its location by searching for a new point in the neighborhood along the normal direction such that the intensity profile of the new point best fits the boundary intensity model learned for this vertex point.

- Step 2. Find the parameters of the statistical shape model that best fits the whole set of new vertex points obtained in the first step

Algorithm 1 MSL algorithm

Step 1. For training, a set of 3D organs are hand-labeled with their bounding boxes $\{\mathfrak{T}, \mathfrak{S}, \mathfrak{R}\}$.

Step 2. Record a set of $\mathfrak{n}$ position samples $\{\mathfrak{T}_1, \ldots, \mathfrak{T}_\mathfrak{n}\}$ randomly sampled around the true object positions $\{\mathfrak{T}\}$.

Step 3. Set parameters $_\mathfrak{S}$ and $_\mathfrak{R}$ with the mean values $\mathfrak{S}^*$ of $\{\mathfrak{S}\}$ and $\mathfrak{R}^*$ of $\{\mathfrak{R}\}$ as priors.

Step 4. Compute the distances $dist_c((\mathfrak{T}_i, \mathfrak{S}^*, \mathfrak{R}^*), (\mathfrak{T}_t, \mathfrak{S}_t, \mathfrak{R}_t)), i = 1, \ldots, \mathfrak{n}$.

$$dist_c((\mathfrak{T}_i, \mathfrak{S}^*, \mathfrak{R}^*), (\mathfrak{T}_t, \mathfrak{S}_t, \mathfrak{R}_t) = ||\mathfrak{C}_\mathfrak{i} - \mathfrak{C}_\mathfrak{t}|| \tag{2.8}$$

where $\mathfrak{C}_\mathfrak{i}$ and $\mathfrak{C}_t$ are the centers of the sampling box $(\mathfrak{T}_i, \mathfrak{S}^*, \mathfrak{R}^*)$ and the ground truth box, $(\mathfrak{T}_t, \mathfrak{S}_t, \mathfrak{R}_t)$, respectively. $||\mathfrak{C}_\mathfrak{i} - \mathfrak{C}_\mathfrak{t}||$ is the Euclidean distance between the two centers.

Step 5. The box samples $\{(\mathfrak{T}_i, \mathfrak{S}^*, \mathfrak{R}^*), \ldots, (\mathfrak{T}_\mathfrak{n}, \mathfrak{S}^*, \mathfrak{R}^*)\}$ are divided into positive $\mathfrak{d}^+_\mathfrak{T}$ if $dist_c((\mathfrak{T}_i, \mathfrak{S}^*, \mathfrak{R}^*), (\mathfrak{T}_t, \mathfrak{S}_t, \mathfrak{R}_t) < \theta_1$ or into negative $\mathfrak{d}^-_\mathfrak{T}$ if $dist_c((\mathfrak{T}_i, \mathfrak{S}^*, \mathfrak{R}^*), (\mathfrak{T}_t, \mathfrak{S}_t, \mathfrak{R}_t) > \theta_2$

Step 6. Train a classifier $\mathfrak{C}_\mathfrak{T}$ to discriminate $\mathfrak{d}^+_\mathfrak{T}$ from $\mathfrak{d}^-_\mathfrak{T}$ based on extracted image features.

Step 7. Use $\mathfrak{C}_\mathfrak{T}$ to classify training samples.

Step 8. Top $\mathfrak{m}$ candidates are retained as $\{(\mathfrak{T}'_1, \mathfrak{S}^*, \mathfrak{R}^*), \ldots, (\mathfrak{T}'_\mathfrak{m}, \mathfrak{S}^*, \mathfrak{R}^*)\}$

Step 9. Each candidate $(\mathfrak{T}'_i, \mathfrak{S}^*, \mathfrak{R}^*)$ $i = 1, \ldots, \mathfrak{m}$ is augmented with $\mathfrak{n}$ samples from the scale space, resulting in $\{(\mathfrak{T}'_i, \mathfrak{S}_1, \mathfrak{R}^*), \ldots, (\mathfrak{T}'_i, \mathfrak{S}_\mathfrak{n}, \mathfrak{R}^*)\}$

Step 10. Similarly, the above $\mathfrak{mn}$ samples are divided into $\mathfrak{d}^+_\mathfrak{S}$ if $dist_b((\mathfrak{T}'_i, \mathfrak{S}_j, \mathfrak{R}^*), (\mathfrak{T}'_t, \mathfrak{S}_t, \mathfrak{R}_t) < \tau_1$ or into negative $\mathfrak{d}^-_\mathfrak{S}$ if $dist_b((\mathfrak{T}'_i, \mathfrak{S}_j, \mathfrak{R}^*), (\mathfrak{T}'_t, \mathfrak{S}_t, \mathfrak{R}_t) > \tau_2$

$$dist_b(box_1, box_2) = \sum_{i=1}^{8} ||\mathfrak{v}^i_1 - \mathfrak{v}^i_2||/8 \tag{2.9}$$

where $\mathfrak{v}^i_1$ and $\mathfrak{v}^i_2$ are the i-th vertex of box_1 and box_2, respectively. This distance is based on all corners of the box, not just the center

- Step 3. Goto Step 1 until the convergence conditions are met.

Algorithm 2 MSL algorithm (cont.)

Step 11. Train a classifier $\mathfrak{C}_{\mathfrak{S}}$ to discriminate $\mathfrak{d}^{+}_{\mathfrak{S}}$ from $\mathfrak{d}^{-}_{\mathfrak{S}}$ using image features.

Step 12. Use $\mathfrak{C}_{\mathfrak{S}}$ to classify samples in training (e.g., the above $\mathfrak{mn}$ samples).

Step 13. Top $\mathfrak{m}$ candidates from Step 12 are retained as $\{(\mathfrak{T}'_1, \mathfrak{S}'_1, \mathfrak{R}^*), \ldots, (\mathfrak{T}'_{\mathfrak{m}}, \mathfrak{S}'_{\mathfrak{m}}, \mathfrak{R}^*)\}$

Step 14. Each candidate $((\mathfrak{T}'_i, \mathfrak{S}'_i, \mathfrak{R}^*))$ $i = 1, \ldots, \mathfrak{m}$ is augmented with $\mathfrak{n}$ samples $\{(\mathfrak{T}'_i, \mathfrak{S}'_i, \mathfrak{R}_j), \ldots, (\mathfrak{T}'_i, \mathfrak{S}'_i, \mathfrak{R}_{\mathfrak{n}})\}$ that have different orientations sample from a uniform distribution.

Step 15. Similarly, the above $\mathfrak{mn}$ samples are divided into $\mathfrak{d}^{+}_{\mathfrak{R}}$ if $dist_b((\mathfrak{T}'_i, \mathfrak{S}'_i, \mathfrak{R}_j), (\mathfrak{T}'_t, \mathfrak{S}'_t, \mathfrak{R}_t) < \eta_1$ or into negative $\mathfrak{d}^{-}_{\mathfrak{R}}$ if $dist_b((\mathfrak{T}'_i, \mathfrak{S}'_i, \mathfrak{R}_j), (\mathfrak{T}'_t, \mathfrak{S}'_t, \mathfrak{R}_t) > \eta_2$

Step 16. Train a classifier $\mathfrak{C}_{\mathfrak{R}}$ to discriminate $\mathfrak{d}^{+}_{\mathfrak{R}}$ from $\mathfrak{d}^{-}_{\mathfrak{R}}$ using image features.

Algorithm 3 MSL algorithm for testing

Step 1. For every 3D object box candidate in the test CT volume, use $\mathfrak{C}_{\mathfrak{T}}$ to classify whether it contains an organ of interest.

Step 2. Top $\mathfrak{m}$ candidates are retained as $\{(\mathfrak{T}'_1, \mathfrak{S}'_1, \mathfrak{R}'_1), \ldots, (\mathfrak{T}'_{\mathfrak{m}}, \mathfrak{S}'_{\mathfrak{m}}, \mathfrak{R}'_{\mathfrak{m}})\}$

Step 3. Each candidate $(\mathfrak{T}'_i, \mathfrak{S}'_i, \mathfrak{R}'_i)$ $i = 1, \ldots, \mathfrak{m}$ is augmented with $\mathfrak{n}$ samples from the scale space, resulting in $\{(\mathfrak{T}'_i, \mathfrak{S}_1, \mathfrak{R}'_i), \ldots, (\mathfrak{T}'_i, \mathfrak{S}_{\mathfrak{n}}, \mathfrak{R}'_i)\}$

Step 4. Use $\mathfrak{C}_{\mathfrak{S}}$ to classify the above $\mathfrak{mn}$ samples from Step 3.

Step 5. Top $\mathfrak{m}$ candidates from Step 4 are retained as $\{(\mathfrak{T}'_1, \mathfrak{S}'_1, \mathfrak{R}'_1), \ldots, (\mathfrak{T}'_{\mathfrak{m}}, \mathfrak{S}'_{\mathfrak{m}}, \mathfrak{R}'_{\mathfrak{m}})\}$

Step 6. Each candidate $((\mathfrak{T}'_i, \mathfrak{S}'_i, \mathfrak{R}'_i))$ $i = 1, \ldots, \mathfrak{m}$ is augmented with $\mathfrak{n}$ samples $\{(\mathfrak{T}'_i, \mathfrak{S}'_i, \mathfrak{R}_1), \ldots, (\mathfrak{T}'_i, \mathfrak{S}'_i, \mathfrak{R}_{\mathfrak{n}})\}$ that have different orientations sample from a uniform distribution.

Step 7. Use $\mathfrak{C}_{\mathfrak{R}}$ to classify samples the above $\mathfrak{mn}$ samples from Step 6

Step 8. Top $\mathfrak{m}$ candidates in Step 7 are retained as $\{(\mathfrak{T}'_1, \mathfrak{S}'_1, \mathfrak{R}'_1), \ldots, (\mathfrak{T}'_{\mathfrak{m}}, \mathfrak{S}'_{\mathfrak{m}}, \mathfrak{R}'_{\mathfrak{m}})\}$ as the final output.

2.1.7 Applications

Kainmüller, Lange and Lamecker [57] use the statistical shape model built from [72] as their liver model and a heuristic intensity distribution to model liver boundaries. Ling et al. [74]

use hierarchial learning to learn a coarse-to-fine liver model and their method is fast. Okada [82] combined both probabilistic atlas and statistical shape models for liver segmentation. Heinmann [52][51] proposed a method based on a genetic algorithm, a statistical shape model and a deformable model for liver segmentation.

2.1.8 Review for Shape Constrained Automatic Segmentation of the Liver Based on a Heuristic Intensity Model[57]

Kainmüller, Lange and Lamecker [57] described an automatic method for 3D liver segmentation in CT volumes. Their method is a combination of an active shape model and a constrained free-form segmentation. After a standard active shape model based segmentation is finished, they perform a so-called free form segmentation step to further refine the segmentation. The correspondence problem is solved by using a geometric approach that is based on minimizing the distortion of the correspondence mapping between two different surfaces [72]. The statistical shape model built from [72] is used as their liver model, principal component analysis is used to model the variations of the shapes, and a heuristic intensity distribution is used to model liver boundaries. To detect the liver, they first use the information from the DICOM header and detect the lungs which appear as very dark area in the images. The liver model is then translated and rotated, based on the location and orientation of the detected lungs.

Their shape model [72] is built from 43 CT volumes where a 3D triangular mesh model is crated for each liver in a each CT volume. Their overall performance is ranked first in a recent liver segmentation competition. The average symmetric surface distance over 10 testing volumes is 1.1 mm and the system takes 15 minutes to generate the liver segmentation for a CT volume.

2.1.9 Review for Hierarchical, Learning-Based Automatic Liver Segmentation[74]

Ling et al. [74] used hierarchial shape models and a learning-based approach for automatic liver segmentation. They proposed a hierarchial and coarse-to-fine shape model (e.g, a shape pyramid) to model livers. For each layer, principal component analysis is used to

model the shape variations. To model boundary intensity, their idea is to use a boundary classifier to learn a probabilistic boundary response indicating whether a point belongs to a boundary. Because of the heterogeneity of the texture pattern along liver boundaries, they further enhance the above idea by using path dependent boundary classifiers. In other words, a liver surface is further decomposed into five types of patches, liver-lung, liver-heart, liver-kidney, liver-tissue and liver-misc. For each type, a boundary classifier is learned. In testing, the boundary classifier with the maximum boundary response among the five is selected.

For liver detection, marginal space learning is used to find the positions, rotations and scales of the bounding box for the detected liver. After a liver is detected, a common approach is to use the mean shape to initialize the learned shape model. In contrast with the common approach, they use marginal space learning to learn how to initialize the learned shape model (e.g, estimating the first three coefficients for principal component analysis) based on the detection results. After this learning approach is used to initialize the shape model, the learned model is placed in the detected bounding box. Both liver detection and initialization are done on the coarse layer.

For their hierarchial shape model, they propose a hierarchial boundary refinement that can be seen as a coarse-to-fine generalization to the standard boundary refinement. After the learned shape model is initialized on the coarsest layer, the following loop is iterated. The standard boundary refinement is performed for several iterations: patch-dependent boundary classifiers are used to estimate the displacement vector and then principal component subspace projection is performed to find the closest approximation to the addition of the current estimated displacement to the current shape. If the shape model is on the finest layer, the resultant shape model is the final segmentation output. Otherwise, the shape model is upsampled to the next finer layer with thin-plate spline warping and the above loop continues.

Their system is tested on 75 3D CT volumes. After excluding outliers, their mean errors, measured as the average symmetric surface distance, are 1.59 ± 0.50 mm and the median is 1.38mm by using 5-fold cross validation. The system takes 12 seconds to generate the liver segmentation for a CT volume.

2.2 Active Contour (Snakes) and Level Set Methods

The active contour (snake) methodology places an initial contour on the image in the vicinity of the region to be detected and uses an energy function to iteratively modify the contour until it "fits" the region. Let $C : [0, 1] \rightarrow R \times R$ denote an explicit planar parameterized curve. The energy functional minimized in [59] is

$$E(C) = \alpha \int_0^1 E_{int}(C(p))dp + \beta \int_0^1 E_{img}(C(p))dp + \gamma \int_0^1 E_{con}(C(p))dp \tag{2.10}$$

The first term, the internal energy, measures the regularity and smoothness along the curve. While the second term, the image energy, directs the active contour to the desired image properties (e.g., strong gradients), the third term, the external energy, takes user-defined constraints or prior knowledge on the active curve into considerations. α, β and γ are user-defined weights to control the contribution of each energy term. The drawbacks of the snake algorithm include that it can not deal with topological changes, and it requires explicit contour representations. In contrast, level set methods, which use implicit contours, [83][95][32] can remove these drawbacks.

In the implicit contour representation, a contour is described by a signed-distance-function that given a point returns the distance to the contour. Inside the contour, this distance is negative, and outside it is positive. The set of points in which the distance is 0, is called the iso-surface or the zero level set. Let $\phi : R \times R \times \{0 \cup R^+\} \rightarrow R$ denote a time-dependent signed distance function. Let $\phi(x, y, t)$ denote the distance from the contour to the point $(x, y) \in R \times R$ at the time $t \in \{0 \cup R^+\}$. So, a contour $C(t)$ at time t in the two-dimensional plane is represented as the zero level set when it satisfies

$$C(t) = \{(x, y) \in R \times R | \phi(x, y, t) = 0\}. \tag{2.11}$$

Consider the evolution equation for ϕ

$$\frac{\partial \phi}{\partial t} = - | \bigtriangledown \phi | \mathfrak{F} \tag{2.12}$$

where $\bigtriangledown \phi = [\frac{\partial \phi}{\partial y} \ \frac{\partial \phi}{\partial y}]^t$, $|v|$ is the magnitude of a vector v, and $\mathfrak{F}$ is a speed function which can take a variety of parameters and return a real value. Given an initial contour (e.g., $C(t)$ or

$\phi(x, y, t)$ for some known t) and an evolution equation for ϕ, we can estimate $\phi(x, y, t + \delta t)$ at $t + \delta t$ for small δt by, for example, the Euler method.

Xiong et al. [118] used active contours and variational level sets for liver segmentation. Dawant et al. in [36] proposed a level set algorithm with the manual contour initialization and correction for liver segmentation. Mintz, Raicu and Furst [80] used Gabor filtering and active contours for liver segmentation. Chi et al. [31] used rotational template matching and k-means clustering to exclude the irrelevant areas for liver segmentation and then applied gradient vector flow geometric snake for liver segmentation. Furukawa, Shimizu and Kobatake [33] first used rough extraction based on maximum a posterior probability estimation and then a level-set method for more precise liver segmentation. Kompalli et al. [70] used both Markov random fields and active contours for refinement for liver segmentation.

2.3 Graph Cuts

Graph cuts is another common method for image segmentation, made popular by [16][69][99].

2.3.1 Energy Minimization

Given a set of n pixels P and a set of two labels $\mathrm{L} = \{0, 1\}$ where 0 represents the background and 1 represents the foreground, the goal in many computer vision problems is to find a labeling $(\mathrm{l}_1, \ldots, \mathrm{l}_n), \mathrm{l}_i \in \mathrm{L}$ (i.e, a mapping from P to L) which minimizes a standard form of the energy function [69] as shown below .

$$\sum_{p \in \mathrm{P}} D_p(\mathrm{l}_p) + \sum_{p,q \in \mathrm{N}} V_{pq}(\mathrm{l}_p, \mathrm{l}_q) \tag{2.13}$$

where $\mathrm{N} \subset \mathrm{P} \times \mathrm{P}$ is a set of all pairs of two adjacent pixels. $D_p(\mathrm{l}_p)$ is a data term that measures the cost of assigning the label l_p to pixel p, and $V_{pq}(\mathrm{l}_p, \mathrm{l}_q)$ is a smoothness term that measures the cost of assigning the labels l_p and l_q to two adjacent pixels p and q.

Although maximizing an arbitrary energy function like (2.13) over the binary n-cube B^{n} is NP-hard [16], previous work [87][48][15][69][44][62][67] showed what energy functions can be exactly minimized with max flow/min cut techniques and provided details on how to construct a weighted graph with respect to a given energy function. For example, if

the energy functions are submodular (or regular)[69], they can be minimized using max flow/min cut techniques [1].

2.3.2 Applications

Campadelli, Casiraghi and Lombardi [23] detected livers by using heart segmentation information and then used a graph-cut algorithm to segment the livers. Beichel et al. [11] used interactive graph-cuts and segmentation refinement for liver segmentation.

2.4 Adaptive Thresholding, Morphology and Pixel/Voxel Classification

The purpose of adaptive thresholding and pixel/voxel classification is to decide whether a pixel or voxel belongs to an organ of interest based on its or its neighboring features. The commonly used features are intensities or textures and many different types of classifiers [39] can be used for this classification task. After classifying each pixel or voxel, we typically need to use morphology operations [98] such as erosion, dilation, opening and closing to group the pixels or voxels with similar labels to meaningful regions or volumes.

Massoptier and Casciaro [78] used adaptive thresholding to detect livers and refined the segmentations by graph cut. Campadelli, Casiraghi and Lombardi [23] detected livers by using heart segmentation information and then used adaptive thresholding and morphology as an alternative to graph cut to segment livers. Campadelli, Casiraghi and Pratissoli used a fast marching technique for abdominal organs segmentations. Rusko [89] et al. mainly used region-growing with various preprocessing and post-processing steps to segment livers. Ma and Yang [76] used expectation maximization and morphological filters for liver segmentation. Pham et al. [86] segmented livers based on texture information. Rikxoort, Arzhaeva and van Ginneken [110] used voxel classification and atlas matching for liver segmentation. Susomboon, Raicu and Furst [103] proposed a hybrid approach consisted of intensity-based partition, region-based texture classification, connected component analysis and thresholding for liver segmentation. However, their approach is very time-consuming. In contrast with active shape models, Seghers et al. [94] incorporated both local intensity and local shape models for liver segmentation. Schmidt et al. [91] combined pixel classification and

region growing with a semantic knowledge base for liver segmentation. Adaptive thresholding and morphology plays an important role in [60] where Kobashi and Shapiro proposed a knowledge-based approach for organ segmentation from CT images. In the following, we review their work in detail.

2.4.1 Review for Knowledge-based organ identification from CT images [60]

Kobashi and Shapiro [60] proposed a knowledge-based approach for organ segmentation from CT images. Their three main ideas are adaptive thresholding, negative-shape constraints that rapidly rule out infeasible segmentations, and progressive landmarking that extract organs based on the confidences of already found organs.

Here is a summary of the knowledge used in this paper.

- Position in the ordering of gray tone levels amon the organs. For example, bones have the highest gray tones, while the air has the lowest gray tones.

- Relevant gray-tone range. The relevant range is the range within which the threshold that can segment the organ correctly lies.

- Height of gray-tone cliff. Assume that there exists a gray level threshold that can segment a target organ. The height of the "cliff" is the range in which such thresholds fall. In contrast with bones that have very high cliffs, livers and spleens have relatively low cliffs.

- Location relative to some stable landmarks (the aorta and the spine). The coordinate system relative to these landmarks is used.

- Adjacency with other organs. The angle of direction relative to the body-based coordinate system and the distance between two organs of interest are used to represent the relative position.

- Size in terms of the area of the slice for each approximate level of slice. The size is given by the number of pixels in the area of the organ.

- Relations with slices at other levels. The overlap ratio of the regions of interest between adjacent slices is used as a constraint for this relation. The overlap ratio must fall within a certain range or it would be a sign of a wrong segmentation.

- Positive and negative shape constraints. While positive shape constraints describe the shape of acceptable objects, negative ones describe what is unacceptable.

The high-level idea of their algorithm is listed in Algorithm 4.

Algorithm 4 Algorithm in [60]

Step 1. $h \Leftarrow$ the high end of the relevant gray tone range predefined for the organ of interest

Step 2. $l \Leftarrow$ the low end of the relevant gray tone range predefined for the organ of interest

Step 3. $t \Leftarrow h$

while $t > l$ **do**

Step 4.1. Execute the thresholding operation with t.

Step 4.2. Execute the connected components operation on the binary image produced by the previous step.

Step 4.3 Check different properties (area check, location check, positive/negative shape check, overlap check, collision check).

if There are no suitable candidates **then**

Step 4.4. Goto Step 5.

end if

Step 4.5. Select the best candidate from the candidates at t.

Step 4.6 Decrease t by some predefined constant (e.g., 10).

end while

Step 5. Perform "slope check" to select the best threshold. {Check the rate of increase of the area of the candidate region with respect to the reduction of threshold and select the least rate increase.}

Step 6. Execute the predefined morphological operations to get the final result.

Chapter 3

INITIAL RESULTS

3.1 Preliminary Work

In the following three sections, we introduce our preliminary methods [28][29][30] and report the initial results. The chapter is organized as follows. In section 3.1.1, we explain how we solve 3D point correspondence by using minimum description length with 2DPCA [29]. In Section 3.1.2., we explain how to learn statistical shape models by using tensor-based dimension reduction methods [30] after the correspondence problem is solved. We explain how we can use a min s-t cut with side constraints to help in boundary refinement [28] in Section 3.1.3. To highlights these three preliminary methods, Figures 3.1 is generated by replacing the corresponding three components in Figure 2.1 with new ones. In Section 3.2, we will describe some current work on organ detection and boundary refinement and show some very initial results and ideas.

3.1.1 3D Point Correspondence by Minimum Description Length with 2DPCA

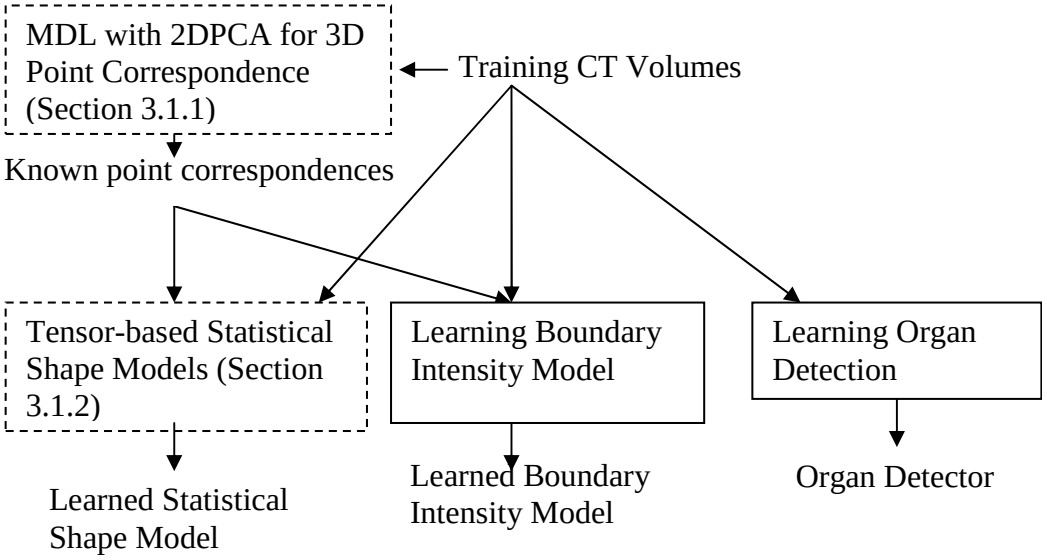
3.1.1.1 2DPCA

In contrast with PCA, two-dimensional principal component analysis [120] (2DPCA) is based on 2D matrices rather than 1D vectors. In other words, in contrast with conventionally using a vector representation to represent a shape, 2DPCA represents a shape by a two-dimensional matrix representation. The idea behind [120] is to project a $3 \times M$ shape matrix $\mathbf{X}$ onto an $M \times 1$ vector $\mathbf{b}$ by the linear transformation.

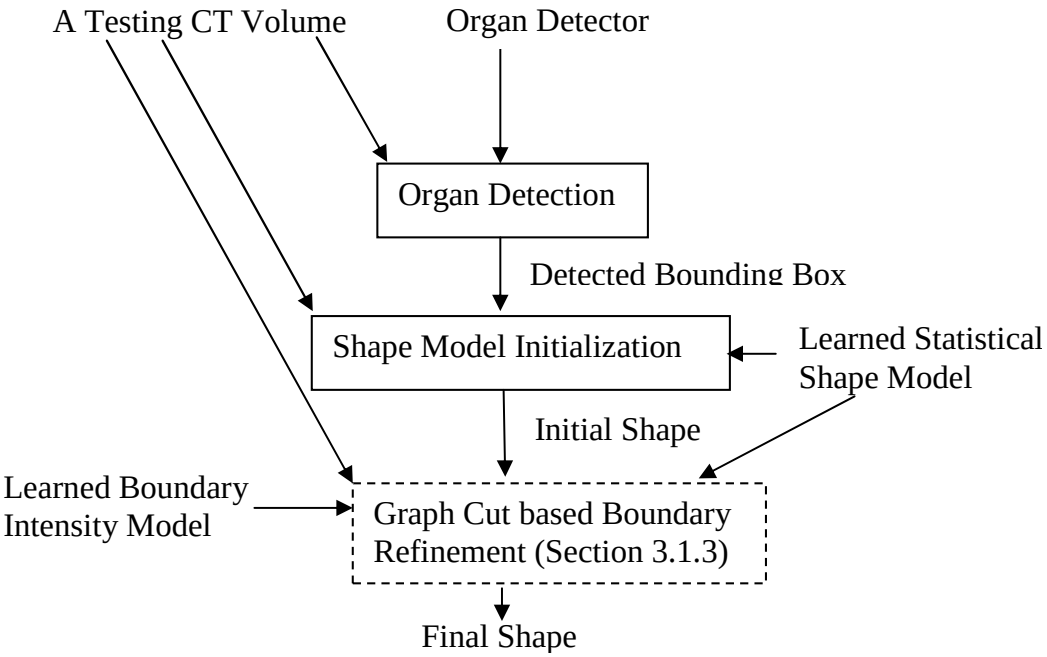
$$\mathbf{c} = \mathbf{X}\mathbf{b} \tag{3.1}$$

The 2DPCA scatter matrix $\mathbf{G}$ is defined as

$$\mathbf{G} = \sum_{i=1}^{N} (\mathbf{X}_i - \bar{\mathbf{X}})^t (\mathbf{X}_i - \bar{\mathbf{X}}) \tag{3.2}$$



(a) Training phase.



(b) Testing phase.

Figure 3.1: Training and testing phases for active shape models with our preliminary methods. These preliminary methods are shown with dash lines.

where

$$\bar{\mathbf{X}} = \frac{\sum_{i=1}^{N} \mathbf{X}_i}{N} \tag{3.3}$$

Similar to PCA, the goal of 2DPCA is to find a projection axis $\mathbf{b}$ that maximizes $\mathbf{b}^t\mathbf{G}\mathbf{b}$. The optimal $\mathscr{K}$ projection axes $\mathbf{b}_k, k = 1, \ldots, \mathscr{K}$ that maximize the above criterion are the eigenvectors of $\mathbf{G}$ corresponding to the largest $\mathscr{K}$ eigenvalues. For a shape matrix $\mathbf{X}$, its reconstruction $\tilde{\mathbf{X}}$, defined below, can be used to approximate it.

$$\tilde{\mathbf{X}} = \bar{\mathbf{X}} + \sum_{k=1}^{\mathscr{K}} \mathbf{c}_l {\mathbf{b}_l}^t \tag{3.4}$$

where $\mathbf{c}_k = (\mathbf{X} - \bar{\mathbf{X}})\mathbf{b}_k$.

3.1.1.2 Minimum Description Length with 2DPCA

We extend the MDL-based objective function, (2.4), to 2DPCA. In other words, instead of using the eigenvalues computed by PCA for finding the best correspondence for a set of 3D shapes, we propose to use those computed by 2DPCA. We propose a gradient descent approach to minimize the objective function based on the ideas in Section 2.1.1.2 to compute the Jacobian matrix for the gradient of the objective function. In order to reuse (2.5) and (2.6), (2.7) must be extended to 2DPCA. Let the eigen-decomposition of $\mathbf{G}$ be defined by

$$\mathbf{D} = \mathbf{U}^T\mathbf{G}\mathbf{U} \tag{3.5}$$

where $\mathbf{D}$ is the diagonal matrix composed of the eigenvalues of $\mathbf{G}$, and $\mathbf{U}$ is a matrix composed of eigenvectors of $\mathbf{G}$. The extension of (2.7) to 2DPCA is defined by

$$\frac{\partial \mathbf{D}}{\partial \mathbf{X}_i(j,k)} = \mathbf{U}^t(\sum_{z=1}^{N}((\delta_{z==i}\mathbf{J}_{jk} - \frac{1}{N}\mathbf{J}_{jk})^t(\mathbf{X}_z - \bar{\mathbf{X}}) + (\mathbf{X}_z - \bar{\mathbf{X}})^t(\delta_{z==i}\mathbf{J}_{jk} - \frac{1}{N}\mathbf{J}_{jk})))\mathbf{U} \tag{3.6}$$

where $\mathbf{J}_{jk}$ is a binary matrix all of whose elements are zeros, except the element at position (j,k), which is one. $\delta_{z==i}$ is defined to be one if z=i. Otherwise, $\delta_{z==i}$ is defined to be zero.

3.1.1.3 Experimental Results for 3D Point Correspondence

Our dataset consists of 3D triangular mesh models of 20 livers, 17 left kidneys, 15 right kidneys, and 18 spleens as shown in Figure 3.2. All 3D meshes are constructed from CT

scans of different patients[1]. The 3D point correspondence problems among different 3D mesh models of the organs are solved by both the state-of-the-art MDL-based approach [34] and the proposed approach. The software library [50][51] that implements the MDL-based approach as described in Section 2.1.1.2 is used for its implementation and is also further modified to implement the proposed approach. All the mesh models of the same organ have the same number of vertices (2563) and the same number of faces (5120), and all vertices are used as landmarks to represent the shapes.

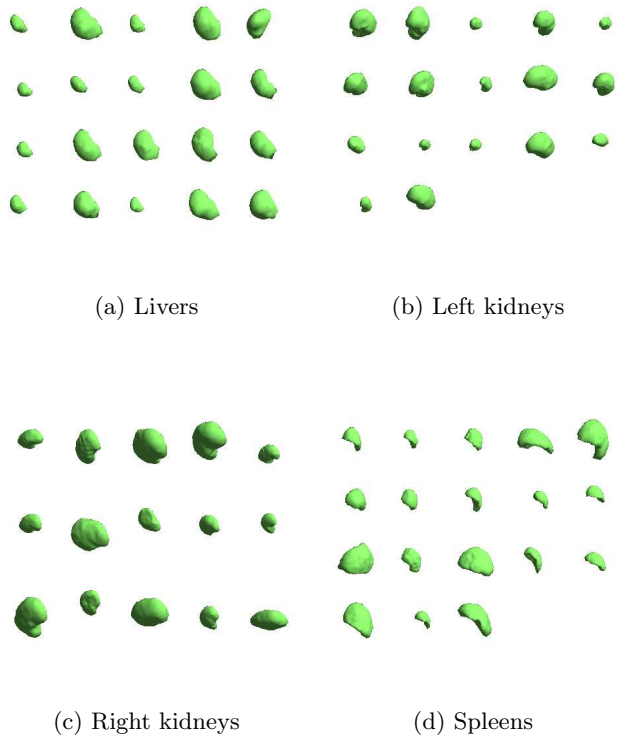
We follow a standard procedure extensively used in [34][35][102][51] to compare different point correspondence methods when the ground truth correspondences among different shapes are not available. Given the correspondences computed by different methods, the standard procedure is to measure the difference between an unknown shape and its reconstruction. Leave-one-out cross validation is used to determine how accurately an algorithm will be able to predict data that it was not trained on. The Euclidean distance (i.e, the sum of the distances between all pairs of corresponding landmarks) and Hausdorff distance are used to measure the shape difference between two shapes.

Even though different correspondence methods use different criteria [35][102] to find the correspondences, the standard procedure [34][35][102][51] uses PCA to model the variations of the shapes whose correspondences are estimated. A potential problem here is that using PCA is biased to the MDL objective function, because the sum of eigenvalues is a good indicator for estimating the reconstruction errors, and the computation of the MDL objective function [34] involves the eigenvalues of PCA. To avoid this potential problem and to be more fair, the original MDL method uses PCA to model shapes, while our extension uses 2DPCA to model shapes.

Figures 3.3 and 3.4 show the average leave-one-out reconstruction errors for different organs. From these two figures, it is clear that the combination of the proposed MDL for 2DPCA and reconstruction using 2DPCA is better than the combination of MDL for PCA and reconstruction using PCA in all test data sets. In addition, as the number of eigenvectors in use increases, the reconstruction error decreases, due to the use of more

[1]We constructs the shape of an organ from manual segmentation of CT scans of a patient by using marching cubes in ITK-SNAP.

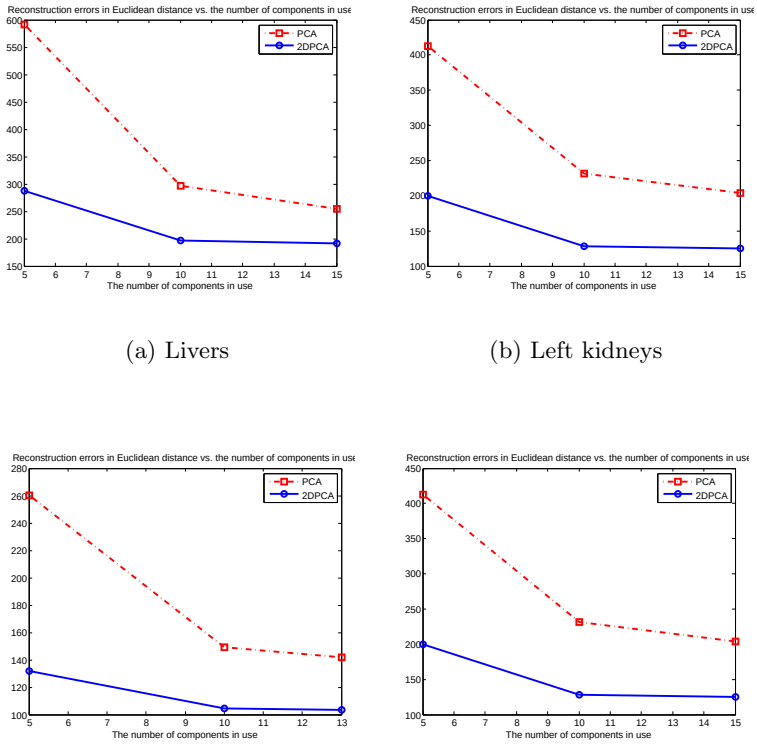
variables to encode the shape variations. However, as the number of eigenvectors in use increases, the differences between the reconstruction errors of these two methods decreases (Figure 3.3), while as the numbers of eigenvectors in use increases, the difference between the reconstruction errors of these two methods can increase accordingly (Figure 3.4). One possible explanation is that the MDL objective functions and the error measures being minimized are all related, to some degree, to Euclidean distances, and so their performances may potentially converge to similar reconstruction errors in terms of Euclidean distance if there is infinite training data. However, their performances may not converge to similar reconstruction errors in terms of Hausdorff distance.



(a) Livers (b) Left kidneys

(c) Right kidneys (d) Spleens

Figure 3.2: The 3D triangular meshes of different organs we use in the experiments.



(a) Livers (b) Left kidneys

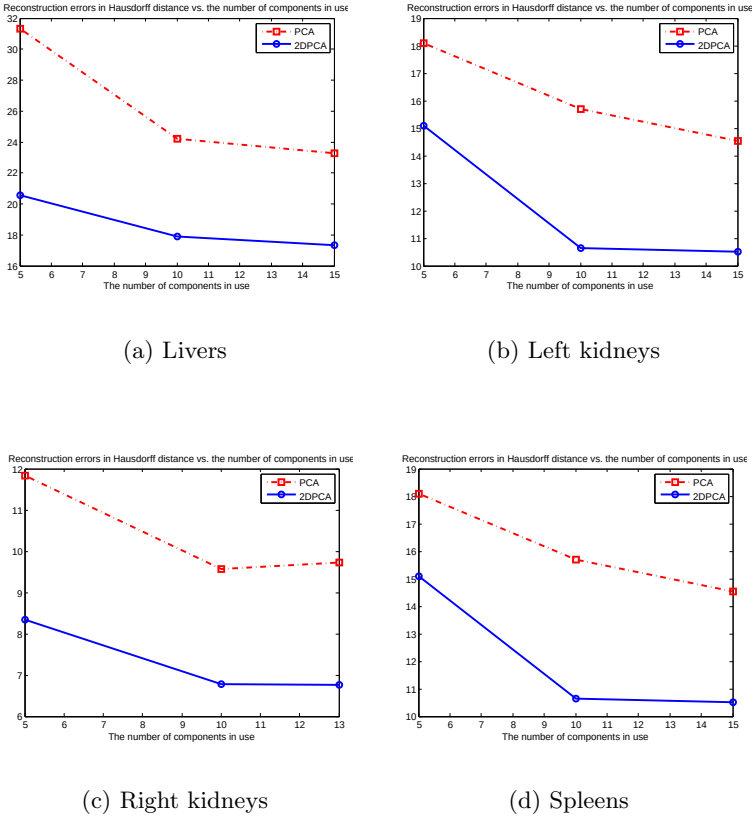
(c) Right kidneys (d) Spleens

Figure 3.3: Generalization ability comparisons in terms of Euclidean distance in different datasets.

3.1.1.4 Summary

We generalize the MDL-based objective function to 2DPCA and propose a gradient descent approach to minimize the objective function. From our experimental results on different sets of 3D shapes of different organs, the combination of the proposed MDL for 2DPCA and reconstruction using 2DPCA is significantly better than the combination of MDL for PCA and reconstruction using PCA.

Instead of using the reconstruction errors, which depend on the particular reconstruction methods, to compare different correspondence methods, we plan to use some datasets whose ground truth correspondences are known to directly compare the proposed method with



(a) Livers (b) Left kidneys

(c) Right kidneys (d) Spleens

Figure 3.4: Generalization ability comparisons in terms of Hausdorff distance in different datasets.

other existing methods. In addition, many different methods [88][108][109] for modeling shapes have been shown to have better potential than PCA. How to generalize MDL-base objective functions to these different approaches is an important direction we will pursue. Furthermore, we would like to relax the assumptions Davies *et al.* made in [34]. One such assumption is that the projected coefficients in different eigenvectors are independent. This assumption may not hold in 2DPCA or in the general case. This work [29] will appear in the proceeding of International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'09).

3.1.2 Tensor-Based Dimension Reduction Methods: An Empirical Comparison on Statistical Shape Models of Organs

After the correspondence problem is solved, we want to learn statistical shape models from the set of shapes whose 3D point correspondences are known. Modeling shape variations is a significant step in active shape models [104] that is widely used in model-based medical image segmentation. A standard method for this step is principal component analysis (PCA). Unlike PCA that uses vector-based representations, varied tensor-based dimension-reduction methods have been recently proposed and achieved better performances than PCA in face recognition [120][111][117]. In contrast with conventionally using a vector representation, tensor-based dimension-reduction methods can represent a shape by a two-dimensional matrix directly or can represent the whole training set of shapes as a tensor [64]. For example, 2DPCA [120] constructs the image covariance matrix directly by using the original image matrices without transforming them into 1D vectors and uses its eigenvectors as principal components. The parallel factor model (Parafac) [49][64] and the Tucker decomposition [107][64] are two major tensor decomposition methods that decompose a tensor into components.

We have not seen any work that has used tensor-based dimension-reduction methods in medical image analysis except [53] that compared 2DPCA [120] with PCA on a normal/abnormal left ventricle shape classification task. In contrast with [53], which mainly focuses on classification, our work requires accurate 3D reconstructions of 3D organs whose shape can vary significantly.

We propose to model shape variations with tensor-based dimension-reduction methods. We will report on our empirical comparison of four reconstruction methods, including PCA, 2DPCA, Parafac and the Tucker decomposition, on several different organs such as livers, spleens and kidneys in Section 3.1.2.2.

3.1.2.1 Tensor-Based Dimension Reduction Methods

A tensor is a generalization of vectors and matrices [64]. The order (or mode) of a tensor is the number of dimensions. We use a third-order tensor $\mathbb{X} \in R^{3\times M\times N}$ to represent the

whole training set of N shapes. The first mode represents the x,y,z coordinates of a point, the second mode represents a point id and the third mode represents a patient id. Although we focus on using third-order tensors in this paper, it is easy to extend our concepts to higher-order tensors. For example, if the above training set changed at regular intervals of time, then a fourth-order tensor in which the fourth mode represents time can be used.

3.1.2.1.1 Parallel Factor Model Parafac [49][64] factorizes a tensor into a weighted sum of component rank-one tensors [64]. In other words, given a tensor $\mathbb{X} \in R^{I \times J \times K}$, Parafac decomposes it as

$$\mathbb{X} \approx \sum_{l=1}^{L} \lambda_l \mathbf{a}_l \circ \mathbf{b}_l \circ \mathbf{c}_l \tag{3.7}$$

where $\mathbf{a}_l \in R^I$, $\mathbf{b}_l \in R^J$, $\mathbf{c}_l \in R^K$, $\lambda_l \in R^1$ for $l = 1, \ldots, L$ and $\circ$ represents the vector outer product [64]. $\mathbf{a}_l, \mathbf{b}_l, \mathbf{c}_l$ are the component rank-one tensors for $l = 1, \ldots, L$ and their roles in tensors or multilinear algebra are conceptually similar to that a basis plays in linear algebra. λ_l is the weight for a order-3 tensor generated by $\mathbf{a}_l \circ \mathbf{b}_l \circ \mathbf{c}_l$ for $l = 1, \ldots, L$. The alternating least squares (ALS) method [49][64] is commonly used to find the Parafac decomposition.

After the decomposition is computed, for a test shape, different methods [112][113] can be used to find the associated coefficient vectors and to compute the reconstruction that approximates it. In this paper, we follow the linear projection method in [112]. Given a shape matrix $\mathbf{X}$, we calculate its reconstruction $\tilde{\mathbf{X}} = \sum_{l=1}^{L} \mathfrak{c}_l \lambda_l \mathbf{a}_l \circ \mathbf{b}_l$ to approximate it by solving the following equation.

$$\min_{\tilde{\mathbf{X}}} ||\mathbf{X} - \tilde{\mathbf{X}}|| \tag{3.8}$$

where $||\mathbf{X}||$ is the Frobenius norm of $\mathbf{X}$. $\mathfrak{c}_l$ is the weight for a order-2 tensor generated by $\lambda_l \mathbf{a}_l \circ \mathbf{b}_l$.

3.1.2.1.2 Tucker Decomposition In contrast with Parafac, which decomposes a tensor into rank-one tensors, the Tucker decomposition is a form of higher-order principal component analysis that decomposes a tensor into a core tensor multiplied by a matrix

along each mode [64]. Given a tensor $\mathbb{X} \in R^{I \times J \times K}$, the Tucker decomposition is given by

$$\mathbb{X} \approx \mathbb{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \tag{3.9}$$

$$= \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r \tag{3.10}$$

where $\mathbb{G} \in R^{P \times Q \times R}$ is called the core tensor, $\mathbf{A} \in R^{I \times P}$, $\mathbf{B} \in R^{J \times Q}$,$\mathbf{C} \in R^{K \times R}$, $\mathbf{a}_p \in R^{I \times 1}$ is the p-th column in $\mathbf{A}$, $\mathbf{b}_q \in R^{J \times 1}$ is the q-th column in $\mathbf{B}$, $\mathbf{c}_r \in R^{K \times 1}$ is the r-th column in $\mathbf{C}$ and $\times_n$ is the n-mode matrix product operator for multiplying a tensor by a matrix in mode n [64]. g_{pqr}, used as shorthand for $\mathbb{G}(p, q, r)$, is a weight for combining $\mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r$. ALS can be used to find the Tucker decomposition.

Let $\mathbf{V}_{(3)}$ be the matrix formed by mode-n matricizing[2] [64] the tensor $\mathbb{G} \times_1 \mathbf{A} \times_2 \mathbf{B}$ with respect to the third mode. Based on the above linear projection idea [112], given a shape vector $\mathbf{x}$, we calculate its reconstruction $\tilde{\mathbf{x}} = \sum_{l=1}^{L} c_l \mathbf{v}_l$ where $\mathbf{v}_l$ is the l-th column of $\mathbf{V}_{(3)}$ to approximate it by solving the following equation.

$$\min_{\tilde{\mathbf{x}}} ||\mathbf{x} - \tilde{\mathbf{x}}|| \tag{3.11}$$

3.1.2.2 Empirical Comparisons of PCA, 2DPCA, Parafac and Tucker Decomposition on Statistical Shape Models of Organs

We use the same dataset as in Section 3.1.1.3 and use [50] to find correspondences among shape models. The tensors are constructed in a manner similar to that described in [112][113]. The task is to measure the difference between an unknown shape model and its reconstruction. We use leave-one-out cross validation to determine how accurately an algorithm will be able to predict data that it was not trained on. Two different commonly used metrics, Euclidean distances (i.e, the sum of the distances between all pairs of corresponding landmarks) and Hausdorff distance, are used to measure the shape difference between two shapes. We use the tensor library [7] for tensor decomposition. In addition, we further investigate how the numbers of components, L, affect the reconstruction errors.

[2]Mode-n matricizing is a way to convert a tensor to a matrix representation.

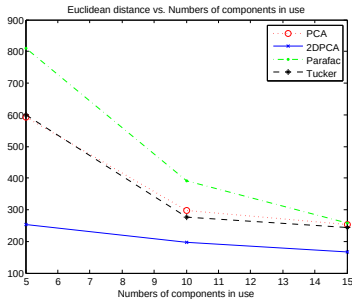
The experimental results are shown in Figures 3.5 and 3.6. From these figures, we can see that 2DPCA performs best among the four methods, the Tucker decomposition performs slightly better than PCA, and the Parafac decomposition is the worst. In fact, the performance differences between 2DPCA and other methods are statistically significant. In addition, as the numbers of components increase, the reconstruction errors of all methods and the differences among different methods decrease. While 2DPCA and PCA had similar computation time in our experiments, the Parafac method and the Tucker decomposition were an order of magnitude slower.

We attribute the poor performance of the Parafac method to the limited expressiveness of rank-one tensors and vector outer products. As explained in [120], because the dimension of the image scatter matrix is much smaller than that of the total scatter matrix, and we deal with small sample size problems where the number of training examples is much smaller than the dimensions of a shape, 2DPCA can capture more accurate covariance information than PCA. Another possible reason is that the spatial dependencies between x, y and z are better preserved in 2DPCA. Although the Tucker decomposition is a form of higher-order principal component analysis [64], its marginal improvement over PCA may be ascribed to the use of the linear projection method to compute the reconstruction, as the potential power of the tensor may be lost in this step.

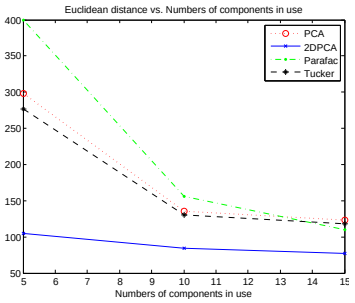
3.1.2.3 Summary

We proposed to use different tensor-based dimension-reduction methods to model shape variations. From our empirical comparisons of the reconstruction errors, 2DPCA is the best among the four compared methods and the performance differences between 2DPCA and the other methods are statistically significant.

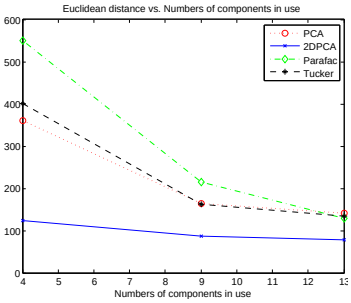
We are currently applying learned models from these tensor-based dimension-reduction methods to model-based organ segmentations. We will also investigate different ways [97][113][117][64] to further improve 2DPCA and tensor decompositions. For example, the grid-sampling strategy [97] is proposed to further improve the performance of 2DPCA. In addition, in contrast to using the linear projection method, a multilinear method that can



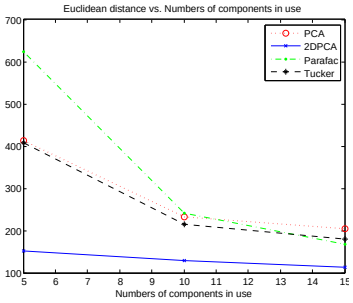
(a) Livers (Euclidean distance)



(b) Left kidneys (Euclidean distance)



(c) Right kidneys (Euclidean distance)



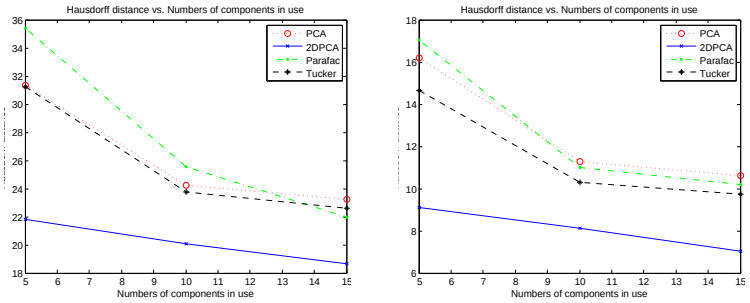
(d) Spleens (Euclidean distance)

Figure 3.5: Reconstruction errors in terms of Euclidean distance for different organs.

simultaneously infer the coefficient vectors in different modes [113] can be used to find better reconstructions. This work [30] will appear in the proceeding of International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'09).

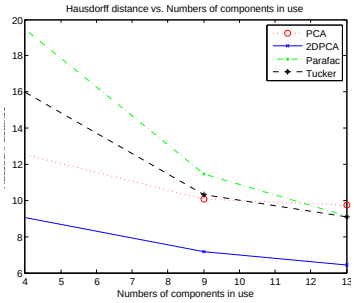
3.1.3 Graph Cuts based Boundary Refinement and Extension

After the organ of interest is detected and the statistical shape is initialized, the boundary refinement process starts. The basic boundary refinement procedure in Section 2.1.5 can be easily solved by using iterative EM-like graph cut algorithms [45][119][4][100][77][128][122].

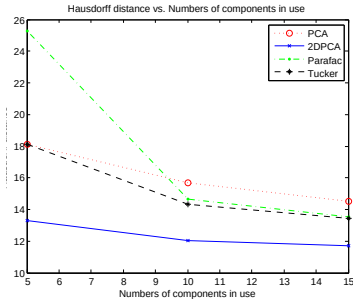


(a) Livers (Hausdorff distance)

(b) Left kidneys (Hausdorff distance)



(c) Right kidneys (Hausdorff distance)



(d) Spleens (Hausdorff distance)

Figure 3.6: Reconstruction errors in terms of Hausdorff distance for different organs.

In fact, we can view using statistical shape models as adding constraints into conventional graph-cut algorithms [16][69][121]. Different constraints such as shape constraints [45][123][4][100] or partial labeling constraints [17] have been incorporated into graph cut algorithms. There are two classes of constraints: hard constraints that must be satisfied by any solution and soft constraints that may or may not be satisfied. For example, Zeng et al. [123] considered hard constraints and proved that topology cut, a min cut algorithm for topology preserving segmentation, is an NP-hard problem, and therefore they provides an approximation algorithm for it. Boykov and Jolly [17] considered other hard constraints

such as the ways that some pixels must be foreground pixels or background pixels. In contrast with hard constraints, soft constraints typically change the capacities of the edges in graphs based on shape priors [45][4][100], which can then be exactly solved with min s-t cut algorithms. The shape priors can be a single template [45][100] or a shape model learned from training examples [4][128][77].

In contrast, we focus on adding hard constraints to graph cut algorithms for medical image segmentation as a further extension. In fact, in many computer vision applications, we are interested in adding hard constraints to help to find better solutions. For example, we may know the upper bound of the size of an organ in a CT volume. We explain how we can use min s-t cut with side constraints to further help boundary refinement [28] below.

3.1.3.1 Min s-t Cut Problem with Side Constraints

Formally, we consider a constrained min s-t problem defined as

$$\min\ \mathbf{c}^t\mathbf{l}\ s.t.\ \begin{bmatrix}\mathbf{A}\\ \mathbf{W}\end{bmatrix}\mathbf{l} \leq \begin{bmatrix}\mathbf{b}\\ \mathbf{d}\end{bmatrix},\ \mathbf{l} \in \mathrm{B}^{\mathrm{n}} \tag{3.12}$$

where the elements of $\mathbf{l}$ are decision variables, l_i, and $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$ are from a given min s-t cut problem [1][114]. Intuitively speaking, the elements of $\mathbf{l}$ are binary labels on nodes (i.e, 0 for background and 1 for foreground) and edges in the capacitated graph, $\mathbf{A}$ and $\mathbf{b}$ reflect the relationships among labels in the graph, and $\mathbf{c}$ gives the capacities of edges. $\mathbf{W}$ and $\mathbf{d}$ are additional hard linear constraints that reflect prior knowledge, and we will refer to them as side constraints. For example, we can represent the upper bound constraint as $\sum_{i\in \mathrm{V}-\{s,t\}} \mathrm{l}_i \leq u$ where u is some positive constant, V is the vertex set of the given capacitated graph, s and t are the source and sink nodes, respectively. The reason why $\sum_{i\in \mathrm{V}-\{s,t\}} \mathrm{l}_i \leq u$ represents this constraint is that l_i for $i \in \mathrm{V}$ is 0 or 1 and $\sum_{i\in \mathrm{V}-\{s,t\}} \mathrm{l}_i$ is the size of the foreground object. Similarly, we can represent the lower bound constraint as $-\sum_{i\in \mathrm{V}-\{s,t\}} \mathrm{l}_i \leq -l$ where l is some positive constant. Note that we only consider linear hard constraints in (3.12); nonlinear hard constraints can be translated into linear constraints by introducing new variables called slack variables and additional linear constraints.

We encode size constraints as side constraints. We try two different methods for formu-

lating size constraints: 1) global bounds and 2) local parts. In the first case, we provide global upper and lower bounds on the contours of the object to be segmented derived from training data on which the true contours are drawn. In the second case, by using the slice above a given slice, we provide upper and lower bounds for each row of the object to be segmented.

Because (3.12) is 0-1 integer programming, which is generally NP-hard [81], we use linear programming relaxation and a simple rounding technique (i.e., if a fractional number is close to 1, we round it to 1 otherwise we round it to 0) as a heuristic to get approximate solutions; this is a standard approximation approach to solve NP hard problems [114].

3.1.3.2 Experiments on Graph Cuts with size constraints and without size constraints

In our experiments, we used the graph-cut-based active-contour algorithm in [119] for foreground/background segmentation and modified their program so that (3.12) can be solved. The metrics we use to evaluate our segmentation results were inspired by common precision and recall measures in information retrieval and are similar to the segmentation accuracy measure used in the PASCAL visual object class challenge 2007 [41]. The measures we used to evaluate the foreground/background segmentation are $\frac{\mathrm{I}}{\mathrm{S}}$ and $\frac{\mathrm{I}}{\mathrm{T}}$ where S is the area of the estimated foreground segment, T is the area of the ground truth foreground segment and I is the area of the intersection of S and T. We will call $\frac{\mathrm{I}}{\mathrm{S}}$ the precision and $\frac{\mathrm{I}}{\mathrm{T}}$ the recall of each segmentation result, since $\frac{\mathrm{I}}{\mathrm{S}}$ tells us how much of the segmented region actually belongs to the organ and $\frac{\mathrm{I}}{\mathrm{T}}$ tells us how much of the actual organ the segmented region covers. Intuitively, the higher these two measures are, the better the segmentation performance is. For the medical images in our experiments, the ground truth segmentations have been performed by the author with the supervision of a medical expert. The goals of our experiments are to show how graph cuts with additional constraints perform compared with graph cuts without any constraints, how different size constraints perform and how constraints and parameters (i.e., upper or lower bounds) can be estimated from training images.

For the global bounds technique, the upper and lower bounds on size are estimated

Table 3.1: Quantitative comparisons.

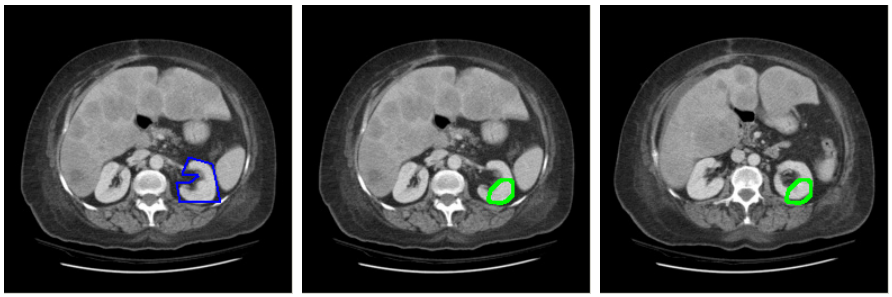
	w.o	w.o	w	w
	precision	recall	precision	recall
Kidney (local parts)	0.9011(0.076)	0.3130(0.0003)	0.9951(0.000)	0.9110(0.003)
Kidney (global bounds)	0.9011(0.076)	0.3130(0.0003)	0.9816(0.0002)	0.9399(0.0005)
Spleen (local parts)	0.9147(0.0023)	0.8248(0.0621)	0.8348(0.0015)	0.9266(0.0038)
Spleen (global bounds)	0.9147(0.0023)	0.8248(0.0621)	0.8783(0.0117)	0.9014(0.0024)

from multiple training images and are entered manually for each organ. The segmentations performed on each slice use the same upper and lower bounds. For the local parts technique, a tracking approach in which the segmentation from the previous image is used as training data for the current image is employed. In this approach, each row of the designated organ in the training image provides separate upper and lower bound size constraints, which taken together can be thought of as a shape constraint.

Quantitative comparisons of both techniques can be found in Table 3.1, which gives the means and variances of precision and recall with and without constraints. The first two rows of Table 3.1 give the results for tracking a kidney over 6 image slices, while the second two rows give the results for tracking a spleen over 10 image slices. For the kidney experiments, size constraints (both global bounds and local parts) significantly improved both precision and recall. For the spleen experiments, recall was significantly improved, but precision went down at the same time. Performance was better for global bounds constraints than local parts constraints in all cases. Our explanation for this observation is that although local part constraints are more expressive than global bounds constraints, they may be too restrictive. In particular, putting a constraint on each row of an organ does not allow its shape to change from slice to slice.

Figure 3.7 illustrates the use of constraints to improve segmentation results. The figure shows how the graph cut algorithm with constraints tracks the contour of a left kidney

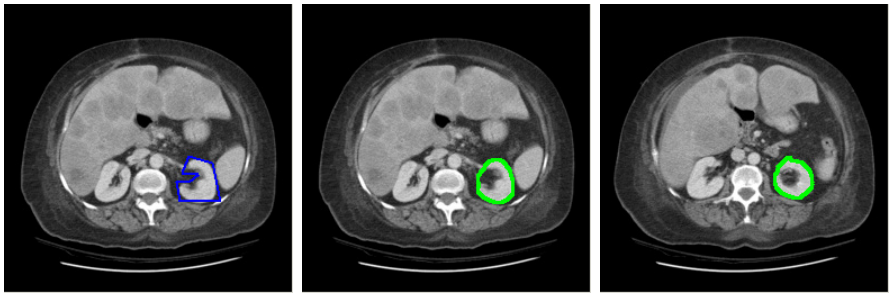
more accurately than that without constraints. Because there are no strong edges around the ground truth shapes of the kidneys in the testing images, the graph cut algorithm with no constraints failed. In contrast, the graph cut algorithm with local part constraints accurately tracks the contour.



(a) Initial contour (without constraints).

(b) Result in the first testing image (without constraints).

(c) Result in the last testing image (without constraints).



(d) Initial contour (with constraints).

(e) Result in the first testing image (with constraints).

(f) Result in the final testing image (with constraints).

Figure 3.7: A comparison between graph cut without any constraints and that with multiple part size constraints on a contour tracking example. The first row shows results without constraints on testing images and the second row shows results with size constraints on testing images. Images (a) and (d) show the same initial contour in the first testing image.

From these results, we can see that size constraints can be used to significantly improve the performance of foreground/background segmentation and that shape constraints are

useful when the shape of the organ being tracked is relatively constant from slice to slice. In addition, our approximation algorithm produced good approximation results in these experiments. This work [28] was published in the proceeding of the International Conference on Pattern Recognition (ICPR'08).

3.2 Current Work

In this section, we describe some current work on organ detection and boundary refinement and show some very initial results and ideas.

3.2.1 An Boosting Based Organ Detection Approach

Currently, we treat organ detection as a binary classification problem. We divide each CT slice into 32x32 nonoverlapping images blocks and learn a classifier to determine whether an image block belongs to an organ of interest. We use both global features and local features. The global features used are the gray-tone histogram of the image slice and its slice index, which indicates its position in the full CT scan. The local features used include the position of a block, the mean and variance of its intensity values, and its intensity histogram. The SVM linear classifier is used as a weak classifier, and Adaboosting is used to combine these weak classifiers.

We have run a set of very preliminary experiments for detection of livers, kidneys, and spleens. The CT images from which 20 livers, 17 left kidneys, 15 right kidneys, and 18 spleens were built in our previous experiments were used for these experiments (Figure 3.2). There are 3024 to 4159 slices for each organ, depending on its visibility in a slice. The organs are partitioned into two disjoint and roughly equal-size sets where one is used as a training set and the other is used as a testing set. For each organ, a strong classifier that results from combining 20,000 weak classifiers with Adaboosting is used in the experiments. Table 3.2 shows the confusion matrices for each organ detection, and Figures 3.8 to 3.11 show some examples of the different organ detections.

A noticeable behavior is that many false positive and false negative examples occurred near the boundaries of organs. One possible explanation for this behavior is that a training image block is seen as a positive example even if the block only overlaps the ground truth

Table 3.2: Training and testing confusion matrices for different organ detection.

(a) Livers (training)

	Positive (Predicted)	Negative (Predicted)
Positive (Actual)	42147(96.23%)	1653(3.77%)
Negative (Actual)	19597(4.57%)	409691(95.43%)

(b) Livers (testing)

	Positive (Predicted)	Negative (Predicted)
Positive (Actual)	42711(91.23%)	4107(8.77%)
Negative (Actual)	35804(6.57%)	509004(93.43%)

(c) Left kidneys (training)

	Positive (Predicted)	Negative (Predicted)
Positive (Actual)	5499(96.96%)	171(3.04%)
Negative (Actual)	24482(6.58%)	347498(93.42%)

(d) Left kidneys (testing)

	Positive (Predicted)	Negative (Predicted)
Positive (Actual)	4905(73.83%)	1739(26.17%)
Negative (Actual)	31211(6.43%)	454433(93.57%)

(e) Right kidneys (training)

	Positive (Predicted)	Negative (Predicted)
Positive (Actual)	4317(98.61%)	61(1.39%)
Negative (Actual)	13688(5.00%)	260206(95.00%)

(f) Right kidneys (testing)

	Positive (Predicted)	Negative (Predicted)
Positive (Actual)	4937(80.33%)	1209(19.67%)
Negative (Actual)	42617(8.70%)	447109(91.30%)

(g) Spleens (training)

	Positive (Predicted)	Negative (Predicted)
Positive (Actual)	10760(96.75%)	362(3.25%)
Negative (Actual)	17154(3.98%)	413580(96.02%)

(h) Spleens (testing)

	Positive (Predicted)	Negative (Predicted)
Positive (Actual)	6010(75.36%)	1965(24.64%)
Negative (Actual)	23489(4.85%)	460824(95.15%)

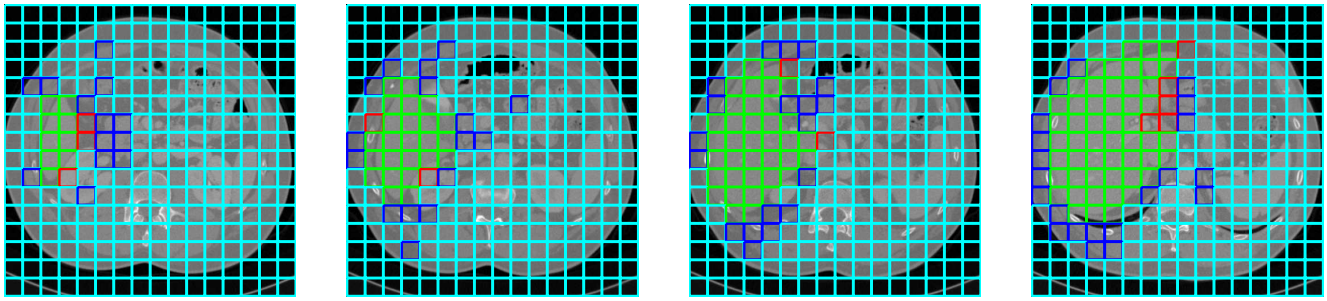
segmentation with a few pixels. Instead of treating the detection problem as a binary classification problem (i.e., a block is either positive or negative), a possible remedy is treating the detection problem as a regression problem in which the label of a block is proportional its overlap area with the ground truth segmentation. We are investigating how to combine these initial detection results to get a final detection result.

3.2.2 Boundary Intensity Model and Shape Model Initialization

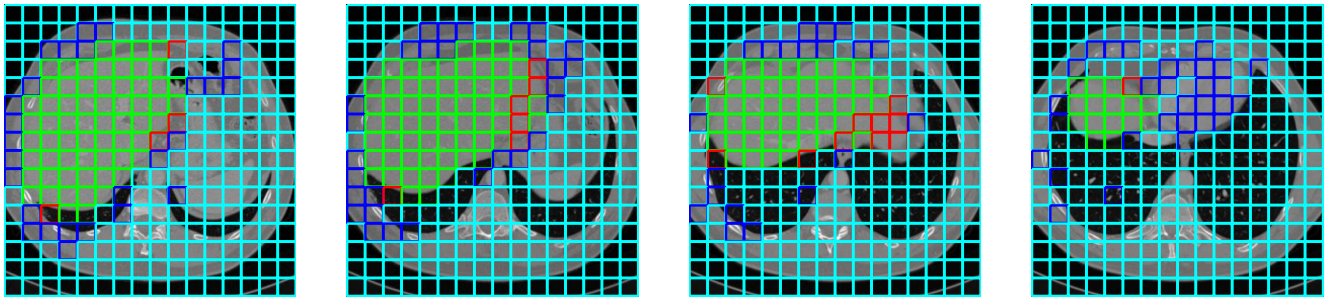
Currently, we use the standard approaches for these two tasks. The intensity profile is modeled as a Gaussian distribution [104], and the shape model is initialized to be the mean shape of all the training shapes whose correspondence problems are solved. Figure 3.12 shows some comparisons between the ground truth shapes and the initial shapes.

3.2.3 Summary

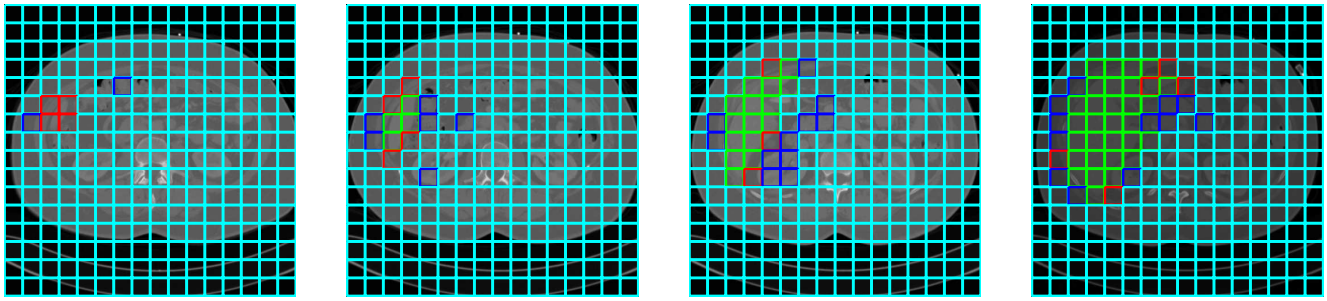
We are currently working to finish the these methods. The first thing we plan to do is to build a system that embodies our whole framework by combining all our preliminary and current methods together. In the next chapter, we propose several directions to further improve our framework.



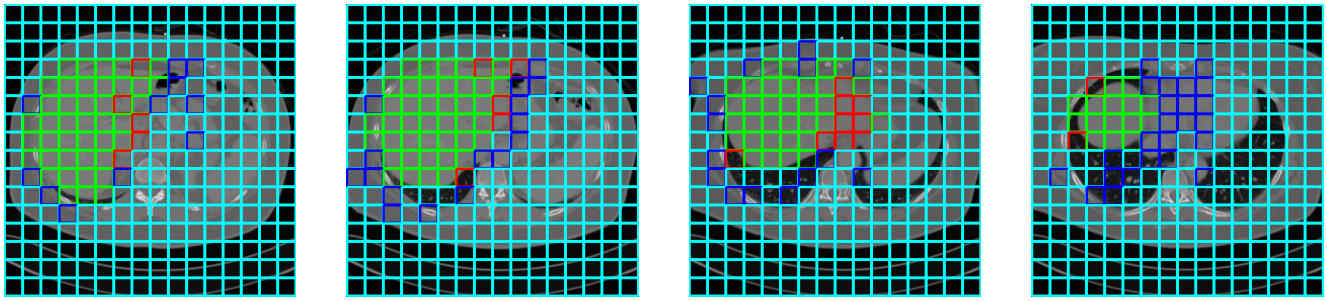
(a) Subject 1, Image 1 (b) Subject 1, Image 2 (c) Subject 1, Image 3 (d) Subject 1, Image 4



(e) Subject 1, Image 5 (f) Subject 1, Image 6 (g) Subject 1, Image 7 (h) Subject 1, Image 8

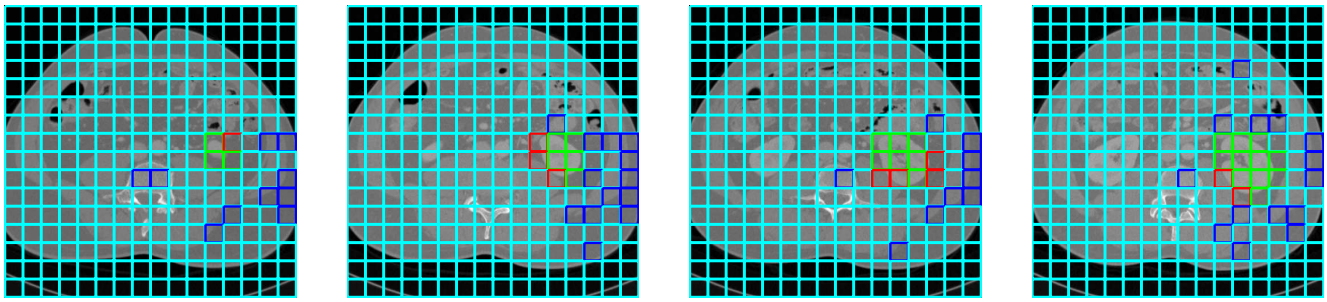


(i) Subject 2, Image 1 (j) Subject 2, Image 2 (k) Subject 2, Image 3 (l) Subject 2, Image 4

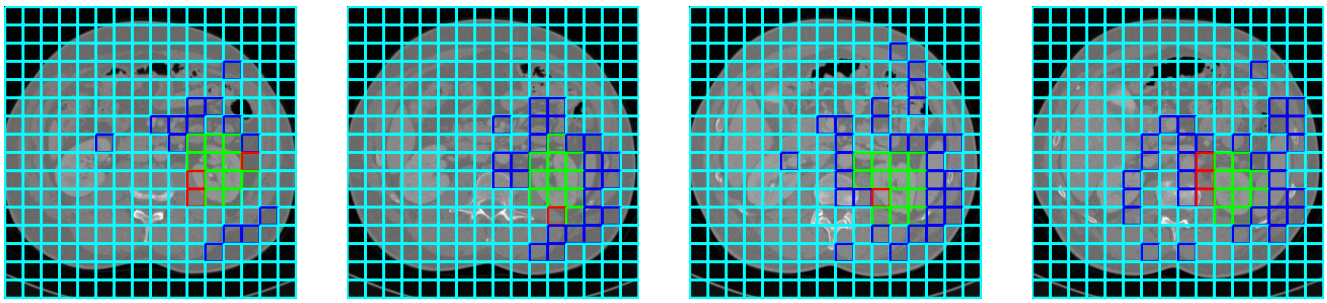


(m) Subject 2, Image 5 (n) Subject 2, Image 6 (o) Subject 1, Image 7 (p) Subject 1, Image 8

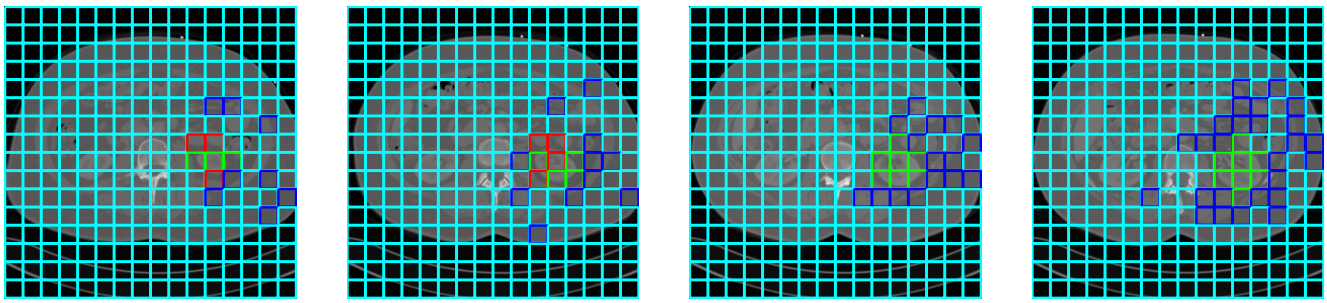
Figure 3.8: Some sample results of liver detection for two test subjects. The true-positive examples (green), false-positive examples (blue), true-negative examples (cyan) and false-negative examples (red) are shown.



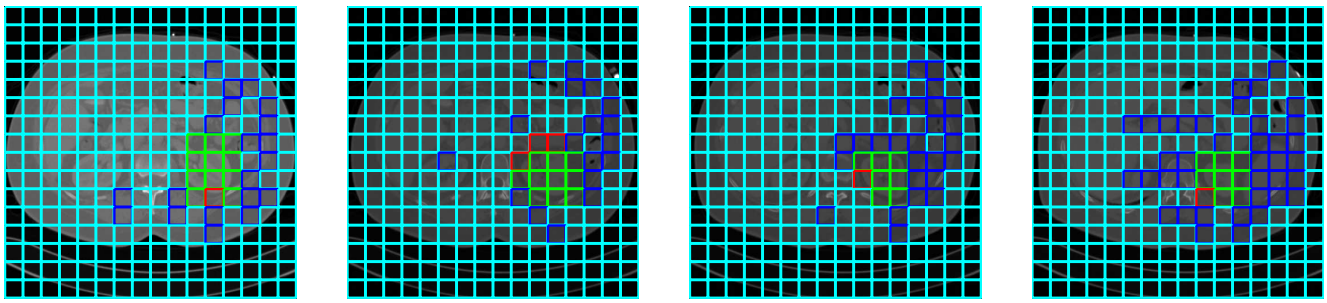
(a) Subject 1, Image 1 (b) Subject 1, Image 2 (c) Subject 1, Image 3 (d) Subject 1, Image 4



(e) Subject 1, Image 5 (f) Subject 1, Image 6 (g) Subject 1, Image 7 (h) Subject 1, Image 8

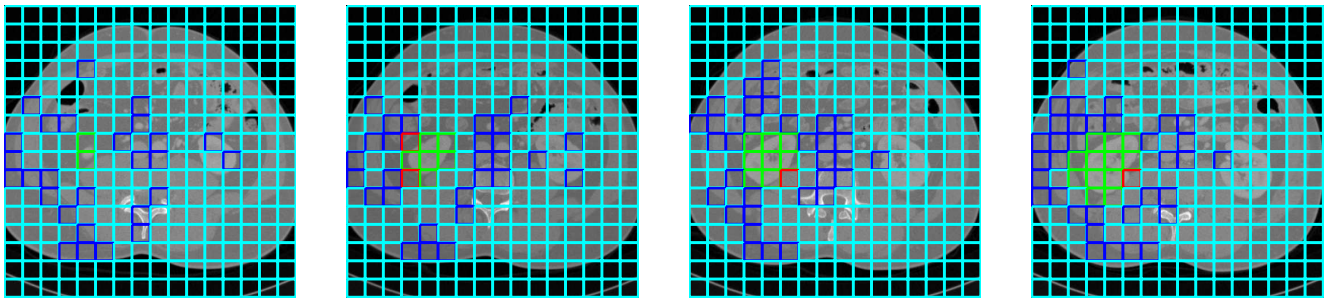


(i) Subject 2, Image 1 (j) Subject 2, Image 2 (k) Subject 2, Image 3 (l) Subject 2, Image 4

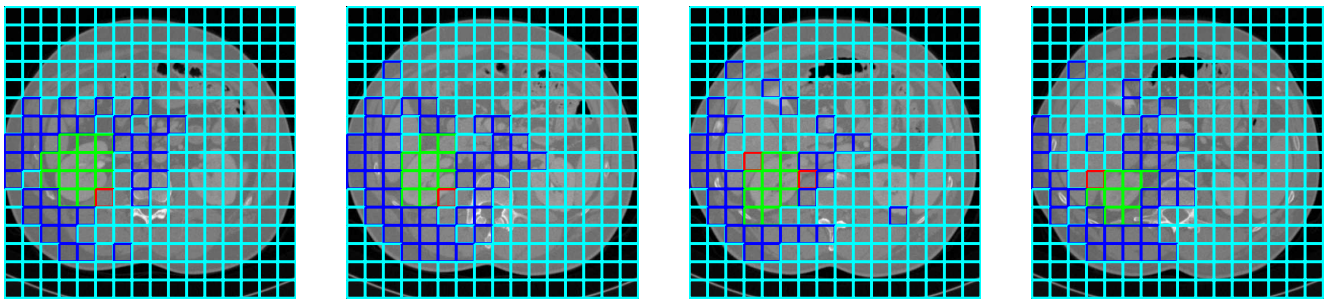


(m) Subject 2, Image 5 (n) Subject 2, Image 6 (o) Subject 2, Image 7 (p) Subject 2, Image 8

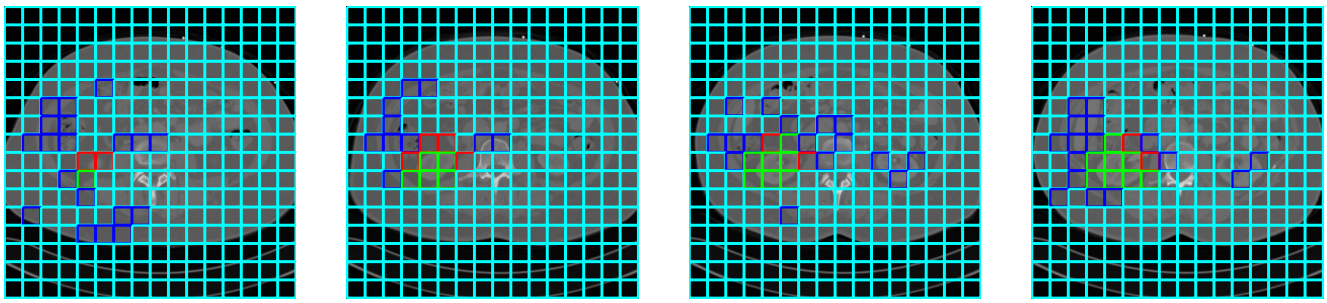
Figure 3.9: Some sample results of left kidney detection for two test subjects. The true-positive examples (green), false-positive examples (blue), true-negative examples (cyan) and false-negative examples (red) are shown.



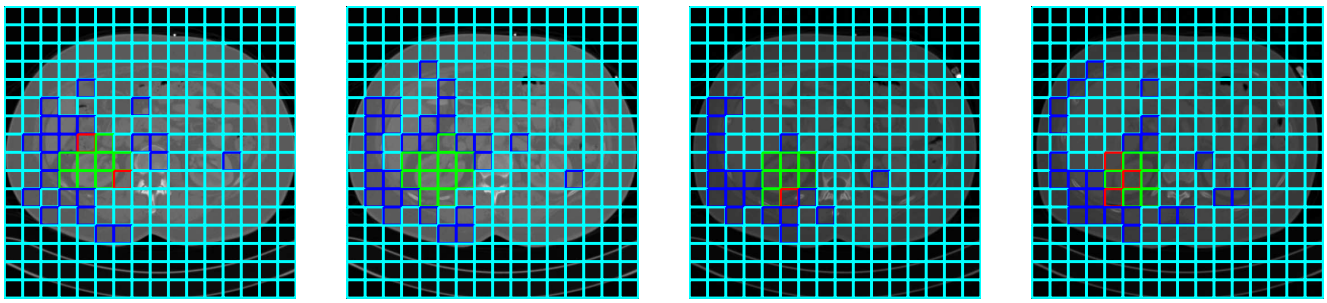
(a) Subject 1, Image 1 (b) Subject 1, Image 2 (c) Subject 1, Image 3 (d) Subject 1, Image 4



(e) Subject 1, Image 5 (f) Subject 1, Image 6 (g) Subject 1, Image 7 (h) Subject 1, Image 8

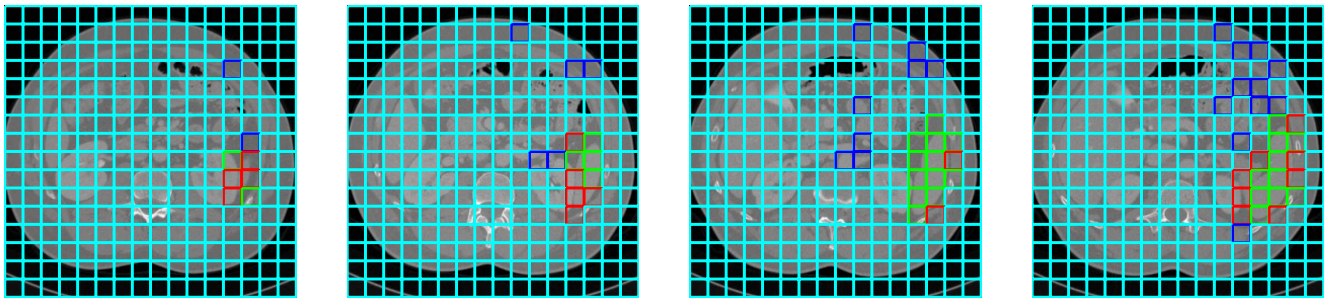


(i) Subject 2, Image 1 (j) Subject 2, Image 2 (k) Subject 2, Image 3 (l) Subject 2, Image 4

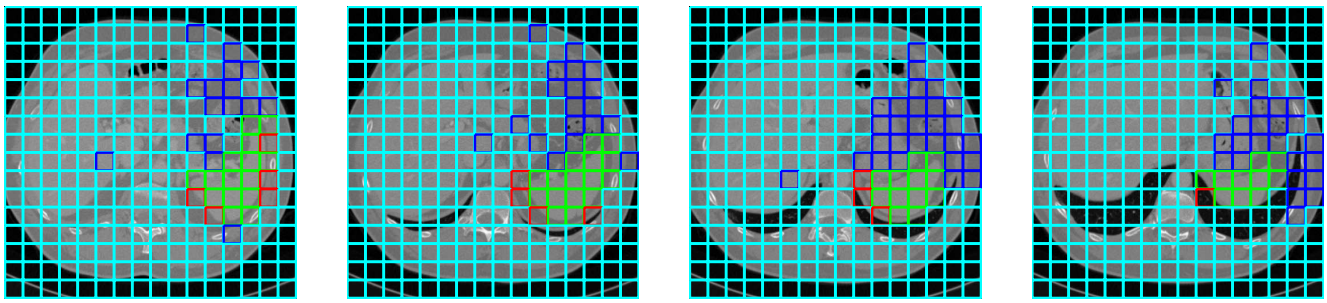


(m) Subject 2, Image 5 (n) Subject 2, Image 6 (o) Subject 2, Image 7 (p) Subject 2, Image 8

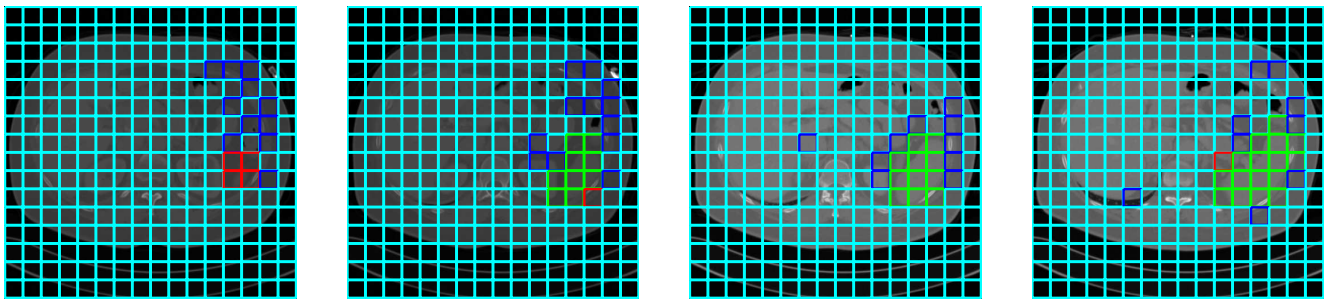
Figure 3.10: Some sample results of right kidney detection for two test subjects. The true-positive examples (green), false-positive examples (blue), true-negative examples (cyan) and false-negative examples (red) are shown.



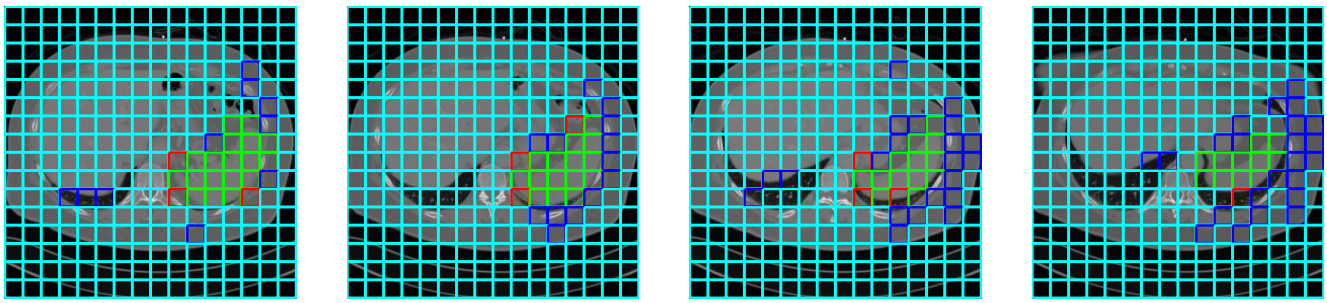
(a) Subject 1, Image 1 (b) Subject 1, Image 2 (c) Subject 1, Image 3 (d) Subject 1, Image 4



(e) Subject 1, Image 5 (f) Subject 1, Image 6 (g) Subject 1, Image 7 (h) Subject 1, Image 8

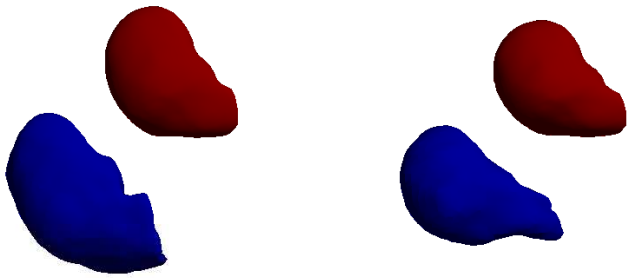


(i) Subject 2, Image 1 (j) Subject 2, Image 2 (k) Subject 2, Image 3 (l) Subject 2, Image 4



(m) Subject 2, Image 5 (n) Subject 2, Image 6 (o) Subject 2, Image 7 (p) Subject 2, Image 8

Figure 3.11: Some sample results of spleen detection for two test subjects. The true-positive examples (green), false-positive examples (blue), true-negative examples (cyan) and false-negative examples (red) are shown.



(a) Sample 1 (b) Sample 2

Figure 3.12: Shape comparison between the ground truth shape models (in blue) and the initial shape model (in red) (i.e., the mean shape model).

Chapter 4

PROPOSAL

In this proposal, we suggest several directions to improve our framework. First, we propose to incorporate nonlinearity and robustness into the MDL-based algorithm in order to get better 3D point correspondence algorithms. Second, for organ detection problems, we propose to apply boosting and bagging to marginal space learning [124][125] and investigate Rodrigues parameters or other better representations of rotation matrices for marginal space learning. Third, to design better boundary refinements algorithms, we propose to incorporate robust optimization and total unimodularity constraints as shape priors into existing EM-like graph cut algorithms.

To the best of our knowledge, we have not seen the proposed directions in any previous work. We will not only describe these ideas, but also provide our motivations in more detail in the following sections.

4.1 Better 3D Point Correspondence Algorithm: Nonlinearity and Robustness

4.1.1 Nonlinear 3D Point Correspondence Algorithm: Minimum Description Length for Tensor-based and Kernel-based 3D Point Correspondence

From our experimental results using 2DPCA (see Section 3.1.1), we can see that there still is much room for improvement. Recently, different tensor-based decompositions [120][111][117] have been proposed and have achieved better performances than PCA in many different applications [120][111][117]. In addition, from a theoretical viewpoint, De Lathauwer, Bart and Vandewalle [38] proved that a high-order tensor singular value decomposition generalizes singular value decomposition for matrices. In other words, singular value decomposition for matrices is a special case of high-order tensor singular value decomposition.

Similarly, there have been many recent algorithms that apply kernel tricks for different linear algorithms and result in corresponding nonlinear algorithms. For example, kernel

principal component analysis (KPCA) and kernel independent component analysis are two representative examples of applying kernel tricks to principal component analysis and independent component analysis. The nonlinear algorithms perform better, because they are more powerful than linear algorithms. Although there has been some previous research [88][108] that use KPCA in active shape models to model shapes and achieve better performances than PCA, there has been no previous work that applies kernel tricks to generalize MDL-base objective functions to KPCA.

We want to generalize MDL-base objective functions [34] to tensor-based decompositions. We propose to develop a MDL objective function for tensor-based decomposition and a gradient descent approach to minimizes it. In addition, motivated by the success of the kernel tricks for different algorithms, we plan to generalize the MDL-based objective function [34] to KPCA by using kernel tricks. Our focuses are a generalized MDL-based objective function [34] for kernel principal component analysis and a gradient descent approach to minimize the generalized objective function.

4.1.2 Robust 3D Point Correspondence Algorithm

Noise or uncertainty in data is very common and may degrade the performance of a 3D point correspondence algorithm that does not take it into consideration. Unfortunately, the current MDL approach [34] is one such example, because it uses PCA to measure the quality of the correspondence but PCA is not very robust to noise or outliers [54][37]. Robust statistics and their applications in computer vision are well reviewed in [79][101]. For example, the idea of M-estimators [54][79][101] has been applied to principal component analysis [37], point registration [42] and template matching [27]. We are interested in applying robust statistics to 3D point correspondence algorithms. For example, we want to replace PCA with robust PCA [37] in the MDL-based approach [34] to make a robust 3D point correspondence algorithm.

4.2 Boosting and Bagging Marginal Space Learning for Organ Detection

Marginal space learning [124][125] or incremental parameter learning [75] was reviewed in detailed in Section 2.1.5.1. This recently proposed organ detection algorithm has been used

extensively to detect livers [74], heart chambers [124][125], and Ileo-Cecal Valve [75] in CT volumes. Recently, Zheng et al. [126] further improved MSL by using better representation of a rotation matrix in which Euler angles are replaced with quaternions and constrained spaces for object position, orientation and scale.

In contrast with previous work that uses MSL as a single classifier, we want to apply boosting [46] and bagging [18] to MSL as a base classifier and combine multiple resultant MSL classifiers together to further improve MSL's accuracy. In fact, AdaBoosting [46], which combines many weak classifiers to produce a powerful committee of classifier, serves as a main building block in many state-of-the-art real-time detectors for different detection problems such as face detection [115] and pedestrians detection [116]. In addition, one disadvantage of representing a rotation matrix with a quaternion Zheng et al. used in [126] is that it requires four parameters, not three (as for Euler angles). Rodrigues parameters that requires three parameters can correct this problem. Hence, we aim to further improve their approach [126] by replacing quaternions with Rodrigues parameters.

4.3 Robust Graph Cuts and Totally Unimodular Hard Constraints as Shape Priors for Better Boundary Refinements

There are at least six assumptions/drawbacks of the basic boundary refinement procedure in Section 2.1.5 that we can relax or remove. First, given a vertex point, Step 1 only searches for the next position along its normal direction. We want to relax this limitation and consider all possible directions. Second, at Step 1, the search for the new position for one vertex point is independent of that of another vertex point. Instead, we would like to break the independence assumption and assume that the movements of two neighboring vertex points should be similar. In other words, we want to add smoothness terms [61][127] to regularize the search for new positions. Third, we are not limited to using only intensity profiles at one vertex. Instead, joint intensity and shape profiles from multiple vertices can be used. Fourth, while Step 1 uses intensity information, Step 2 uses shape information, so these two steps are independent of each other. We want to make these two steps correlated by using both intensity and shape information in both steps. In addition, we would use discriminative learning techniques to learn both intensity and shape features. Fifth, robustness is another

weakness of the current approach that we want to overcome. Sixth, we want to allow more different kinds of shape priors than only statistical shape models.

As mentioned in Section 3.1.3, the basic boundary refinement procedure in Section 2.1.5 can be easily solved by using iterative EM-like graph cut algorithms [45][119][4][100][77][128][122]. In fact, the first four improvements can be incorporated into EM-like graph cut algorithms. If we can make EM-like graph cut algorithms more robust, we can incorporate the fifth improvement into them. Similarly, if we can incorporate different kinds of shape priors into EM-like graph cut algorithms while keeping the resultant algorithms computationally efficient, we can achieve the sixth improvement. In the following, we will present our ideas on how to incorporate the last two improvements into EM-like graph cut algorithms.

4.3.1 Robust Graph Cuts

Different approaches have been proposed to deal with uncertainty in network flow problems [13][14][2][47][3][12][71]. For example, Kouvelis and Yu [71] proposed two approaches, minimax and minimax-regret. While the minimax approach seeks to minimize the worst case performance under a set of scenarios for the data, the minimax-regret approach tries to minimize the worst case loss in objective values. Unfortunately, under these two approaches, the robust counterpart of many polynomially solvable combinatorial optimization problems becomes NP-hard [71][2]. In contrast, Bertsimas and Sim [13] proposed a different minimax approach. Under their approach, the robust counterpart of a polynomially solvable 0-1 discrete optimization problem remains polynomially solvable. In addition, the robust counterpart of an NP-hard α-approximation 0-1 discrete optimization problem[1] remains an α-approximation [114][81][93].

To make existing graph cut algorithms or EM-like graph cut algorithms more robust, we propose to combine the minimax approach proposed by Bertsimas and Sim [13] with them. There has been no previous work that uses this proposed idea in computer vision applications. Thus, one of our contributions is to introduce this idea to the computer vision

[1] An NP-hard 0-1 discrete optimization problem is α-approximation if there is a polynomial time algorithm that can find an approximation solution whose objective function value is at most α-times of the optimal objective value.

research community. In addition, we plan to further improve their minimax approach by using the idea used for dynamic Markov random field [63] that reuses the flows in similar capacitated graphs to speed the computation.

4.3.2 Totally Unimodular Hard Constraints

As shown in Section 3.1.3, hard constraints can be helpful for image segmentation. However, (3.12) is 0-1 integer programming, which is generally NP-hard [81], so how to balance the performance and efficiency is a key question. We are interested in special cases for (3.12) whose global optimal solutions can be found in polynomial time. One such special case is related to totally unimodularity (TU). A matrix is TU iff the determinant of each square submatrix is equal to 0, 1 or -1 [81][93].

Consider the following standard integer program.

$$\begin{aligned} \min \quad & \mathfrak{c}^t \mathfrak{y} \\ s.t. \quad & \mathfrak{A}\mathfrak{y} \leq \mathfrak{b} \\ & \mathfrak{y} \in B^n \end{aligned} \tag{4.1}$$

Note that (3.12) can be easily represented as the above standard integer program, (4.1), by replacing $\mathbf{c}$, $\mathbf{l}$, $\begin{bmatrix} \mathbf{A} \\ \mathbf{W} \end{bmatrix}$, and $\begin{bmatrix} \mathbf{b} \\ \mathbf{d} \end{bmatrix}$ in (3.12), with $\mathfrak{c}$, $\mathfrak{y}$, $\mathfrak{A}$ and $\mathfrak{b}$ in (4.1), respectively. Given an integer program (e.g.,(3.12)), if its constraint matrix is TU and its right hand side is integral (e.g,, $\mathfrak{b} \in Z^m$), the following important proposition can be proved [81][93].

Proposition 4.3.1 *If $\mathfrak{A}$ is TU, then $P(\mathfrak{b}) = \{\mathfrak{y} \in R^n | \mathfrak{A}\mathfrak{y} \leq \mathfrak{b}, \mathfrak{y} \geq 0\}$ is integral for all $\mathfrak{b} \in Z^m$ for which it is not empty.*

If an integer program satisfies the conditions that its constraint matrix is TU and its right hand side is integral, Proposition 4.3.1 states that we can replace integrality constraints (e.g., $\mathfrak{y} \in B^n$ in (4.1)) with non-negativity constraints ($\mathfrak{y} \geq 0$), which generates a linear program and the optimal solutions to the linear program are integral if the optimal solutions exist. In other words, an integer program whose constraint matrix is TU and whose right hand

side is integral can be exactly[2] solved by linear programming in polynomial time [55][93], since all its basic feasible solutions are integral. If $\mathfrak{A}$ in (4.1) is TU and $\mathfrak{b} \in Z^m$ in (4.1), then the exact solutions if any to (4.1) are the same as the solutions to the following linear program (4.2) that can be solved in polynomial time.

$$\begin{aligned} \min \quad & \mathfrak{c}^t \mathfrak{y} \\ s.t. \quad & \mathfrak{A}\mathfrak{y} \leq \mathfrak{b} \\ & \mathfrak{y} \geq 0 \end{aligned} \tag{4.2}$$

We want to incorporate the concept of TU into (3.12) in order to generate an effective and efficient algorithm for organ segmentation. We focus on the following two directions. First, we want to find interesting semantic constraints that can be represented as TU constraints either by manual design or by learning from training images. Second, we will investigate polynomial time algorithms [55][93] specialized for solving a linear program with a totally unimodular constraint matrix.

[2]By exactly, we mean that both optimization problems have same optimal solutions and if we solve one optimally, we get the optimal solutions to the other.

Chapter 5

CONCLUSION

In this paper, we have presented our current approach for organ segmentation in CT scans. We have discussed our work in 3D point correspondence, statistical shape models and min s-t cuts with side constraints and have shown our initial results. These ideas together can serve as building blocks for building a system for organ segmentation in CT scans. The initial results identify and motivate several important research directions that merit further research. We will continue to extend the current work by following the research directions mentioned in our proposal.

- To design a better 3D point correspondence algorithm, we propose to incorporate nonlinearity and robustness into existing the MDL-based 3D point correspondence algorithm.

- To design a better organ detector, we propose to apply boosting and bagging to MSL and investigate Rodrigues parameters or other better representations of rotation matrices for MSL.

- To design better boundary refinements algorithm, we propose to incorporate robust optimization and TU constraints as shape priors into existing EM-like graph cut algorithms. This is a completely different algorithm.

We believe that the proposed research directions can lead to not only a better system for organ segmentation in CT scans, but also to a better understanding of the organ segmentation problem. Last, but not least, the methods we propose are general and can be widely used in many other applications, such as such as 2D point correspondence, image segmentation, stereo, image restoration, interactive image segmentation, face alignment and clustering.

BIBLIOGRAPHY

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[2] H. Aissi, C. Bazgan, and D. Vanderpooten. Complexity of the min-max (regret) versions of cut problems. *Lecture notes in computer science*, 3827:789, 2005.

[3] Hassene Aissi, Cristina Bazgan, and Daniel Vanderpooten. Min-max and min-max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197(2):427–438, September 2009.

[4] Asem M. Ali, Aly A. Farag, and Ayman El-Baz. Graph cuts framework for kidney segmentation with prior shape constraints. In *MICCAI'07*.

[5] Fadi A. Aloul, Arathi Ramani, Karem A. Sakallah, and Igor L. Markov. Solution and optimization of systems of pseudo-boolean constraints. *IEEE Trans. Computers*, 56(10), 2007.

[6] Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *J. Mach. Learn. Res.*, 3:1–48, 2003.

[7] Brett W. Bader and Tamara G. Kolda. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *ACM Trans. Math. Softw.*, 32(4):635–653, December 2006.

[8] Peter Barth. A Davis-Putnam based enumeration algorithm for linear pseudo-Boolean optimization. Technical Report MPI–I–95–2–003, Saarbrücken, 1995.

[9] Mokhtar S. Bazaraa, John J. Jarvis, and Hanif D. Sherali. *Linear Programming and Network Flows*. Wiley-Interscience, 2004.

[10] Andreas Beck and Volker Aurich. Hepatux – a semiautomatic liver segmentation system. *Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pages 225–233, 2007.

[11] Reinhard Beichel, Christian Bauer, Alexander Bornik, Erich Sorantin, and Horst Bischof. Liver segmentation in ct data: A segmentation refinement approach. *Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pages 235–245, 2007.

[12] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411–424, 2000.

[13] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Math. Program*, 98:49–71, 2003.

[14] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, pages 35–53, 2004.

[15] A. Billionnet and M. Minoux. Maximizing a supermodular pseudoboolean function: A polynomial algorythm for supermodular cubic functions. *Discrete Applied Mathematics*, 12:1–11, 1985.

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

[17] Yuri Y. Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, pages 105–112, 2001.

[18] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[19] JF Brinkley. Knowledge-driven ultrasonic three-dimensional organ modeling. *IEEE transactions on pattern analysis and machine intelligence*, 7(4):431–441, 1985.

[20] JF Brinkley. Spatial anatomic knowledge for 2-D interactive medical image segmentation and matching. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 460. American Medical Informatics Association, 1991.

[21] Paul Camion. Characterization of totally unimodular matrices. 16(5):1068–1073, 1965.

[22] P. Campadelli, E. Casiraghi, and S. Pratissoli. Fully automatic segmentation of abdominal organs from ct images using fast marching methods. *Computer-Based Medical Systems, 2008. CBMS '08. 21st IEEE International Symposium on*, pages 554–559, June 2008.

[23] Paola Campadelli, Elena Casiraghi, and Gabriele Lombardi. Automatic liver segmentation from abdominal ct scans. In *ICIAP '07: Proceedings of the 14th International Conference on Image Analysis and Processing*, pages 731–736, Washington, DC, USA, 2007. IEEE Computer Society.

[24] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, February 1997.

[25] Donald Chai and Andreas Kuehlmann. A fast pseudo-boolean constraint solver. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 24(3), 2005.

[26] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

[27] J.H. Chen, C.S. Chen, and Y.S. Chen. Fast algorithm for robust template matching with M-estimators. *IEEE Transactions on Signal Processing*, 51(1):230–243, 2003.

[28] Jiun-Hung Chen and Linda G. Shaprio. Medical image segmentation via min s-t cuts with side constraints. In *ICPR*, 2008.

[29] Jiun-Hung Chen and Linda G. Shaprio. 3d point correspondence by minimum description length with 2dpca. In *EMBC*, 2009.

[30] Jiun-Hung Chen and Linda G. Shaprio. Pca vs. tensor-based dimension reduction methods: An empirical comparison on active shape models of organs. In *EMBC*, 2009.

[31] Ying Chi, Peter M. M. Cashman, Fernando Bello, and Richard I. Kitney. A discussion on the evaluation of a new automatic liver volume segmentation method for specified ct image datasets. *Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pages 167–175, 2007.

[32] Daniel Cremers, Mikael Rousson, and Rachid Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *Int. J. Comput. Vision*, 72(2):195–215, 2007.

[33] Hidefumi Kobatake Daisuke Furukawa, Akinobu Shimizu. Automatic liver segmentation method based on maximum a posterior probability estimation and level set method. *Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pages 117–124, 2007.

[34] R.H. Davies, C.J. Twining, T.F. Cootes, J.C. Waterton, and C.J. Taylor. A minimum description length approach to statistical shape modeling. *IEEE Trans. on Medical Imaging*, 2002.

[35] Rhodri Davies, Carole Twining, and Chris Taylor. *Statistical Models of Shape: Optimisation and Evaluation*. Springer Publishing Company, 2008.

[36] Benoit M. Dawant, Rui Li, Brian Lennon, and Senhu Li. Semi-automatic segmentation of the liver and its evaluation on the miccai 2007 grand challenge data set. *Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pages 215–221, 2007.

[37] F. De la Torre and M.J. Black. Robust principal component analysis for computer vision. In *ICCV01*, volume 1, pages 362–369, 2001.

[38] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.

[39] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.

[40] A. Ericsson and K. strm. Minimizing the description length using steepest descent. In *Proc. British Machine Vision Conference, Norwich, United Kingdom*, volume 2, pages 93–102, 2003.

[41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[42] A.W. Fitzgibbon. Robust registration of 2D and 3D point sets. *Image and Vision Computing*, 21(13-14):1145–1153, 2003.

[43] Charles Florin, Nikos Paragios, Gareth Funka-Lea, and James Williams. Liver segmentation using sparse 3d prior models with optimal data support. In *IPMI*, pages 38–49, 2007.

[44] Daniel Freedman and Petros Drineas. Energy minimization via graph cuts: Settling what is possible. In *CVPR*, pages 939–946, 2005.

[45] Daniel Freedman and Tao Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR'05*.

[46] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[47] S. Gast and A. Keimer. Robust Optimization in Network Flows. Technical report, Technical report, University of Erlangen-Nurnberg, 2009.

[48] P Hansen and B Simeone. Unimodular functions. *Discrete Appl. Math.*, 14(3):269–281, 1986.

[49] R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16(1):84, 1970.

[50] T. Heimann, I. Oguz, I. Wolf, M. Styner, and H. Meinzer H. Implementing the automatic generation of 3d statistical shape models with itk. In *Open Science Workshop at MICCAII*, 2006.

[51] Tobias Heimann, Hans-Peter Meinzer, and Ivo Wolf. A statistical deformable model for the segmentation of liver ct volumes. *Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pages 161–166, 2007.

[52] Tobias Heimann, Sascha Mnzing, Hans peter Meinzer, and Ivo Wolf. I.: A shape-guided deformable model with evolutionary algorithm initialization for 3d soft tissue segmentation. In *In: IPMI 2007. Volume 4584 of LNCS., Springer-Verlag*, pages 1–12, 2007.

[53] Heng Huang, F. Makedon, and R. McColl. High dimensional statistical shape model for medical image analysis. *ISBI'08*, pages 1541–1544, May 2008.

[54] P.J. Huber and E.M. Ronchetti. *Robust statistics*. Wiley-Blackwell, 2009.

[55] K. Truemper J. f. Maurras and M. Akgul. Polynomial algorithms for a class of linear programs. *Mathematical Programming*, 21:121–136, 1981.

[56] Olivier Juan and Yuri Boykov. Active graph cuts. In *CVPR*, pages 1023–1029, 2006.

[57] Dagmar Kainmüller, Thomas Lange, and Hans Lamecker. Shape constrained automatic segmentation of the liver based on a heuristic intensity model. *Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pages 109–116, 2007.

[58] G. Kanizsa. *Organization in Vision: Essays on Gestalt Perception*. Praeger, 1979.

[59] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, V1(4):321–331, January 1988.

[60] Masaharu Kobashi and Linda G. Shapiro. Knowledge-based organ identification from ct images. *Pattern Recognition*, 28(4):475–491, 1995.

[61] K. Koffka. *Principles of Gestalt psychology*. Routledge & Kegan Paul, 1955.

[62] Pushmeet Kohli, M. Pawan Kumar, and Philip H. S. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[63] Pushmeet Kohli and Philip H. S. Torr. Effciently solving dynamic markov random fields using graph cuts. In *ICCV '05*, pages 922–929, 2005.

[64] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*. to appear (accepted June 2008).

[65] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007*, pages 1–8, 2007.

[66] Vladimir Kolmogorov, Yuri Boykov, and Carsten Rother. Applications of parametric maxflow in computer vision. *ICCV'07*.

[67] Vladimir Kolmogorov and Carsten Rother. Minimizing nonsubmodular functions with graph cuts-a review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(7):1274–1279, 2007.

[68] Vladimir Kolmogorov and Carsten Rother. Minimizing nonsubmodular functions with graph cuts-a review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(7):1274–1279, 2007.

[69] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):147–159, 2004.

[70] Suryaprakash Kompalli, Mohammed Alam, Raja S. Alomari, Stanley T. Lau, and Vipin Chaudhary. Design of a benchmark dataset, similarity metrics, and tools for liver segmentation. volume 6915, page 691537. SPIE, 2008.

[71] P. Kouvelis. *Robust discrete optimization and its applications*. Kluwer Academic Publishers, 1997.

[72] Hans Lamecker, Thomas Lange, and Martin Seebaß. Segmentation of the liver using a 3D statistical shape model. ZIB Preprint 04-09, 2004.

[73] Yuanzhong Li and W. Ito. Shape parameter optimization for adaboosted active shape model. volume 1, pages 251–258 Vol. 1, Oct. 2005.

[74] H.B. Ling, S.K. Zhou, Y.F. Zheng, B. Georgescu, M. Suehling, and D. Comaniciu. Hierarchical, learning-based automatic liver segmentation. pages 1–8, 2008.

[75] Le Lu, Adrian Barbu, Matthias Wolf, Jianming Liang, Luca Bogoni, Marcos Salganicoff, and Dorin Comaniciu. Simultaneous detection and registration for ileo-cecal valve detection in 3d ct colonography. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 465–478, Berlin, Heidelberg, 2008. Springer-Verlag.

[76] Li Ma and L. Yang. Liver segmentation based on expectation maximization and morphological filters in ct images. *Bioinformatics and Biomedical Engineering, 2007. ICBBE 2007. The 1st International Conference on*, pages 690–693, July 2007.

[77] J. Malcolm, Y. Rathi, and A. Tannenbaum. Graph cut segmentation with nonlinear shape priors. In *IEEE International Conference on Image Processing, 2007. ICIP 2007*, volume 4, 2007.

[78] L. Massoptier and S. Casciaro. Fully automatic liver segmentation through graph-cut technique. *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 5243–5246, Aug. 2007.

[79] P. Meer, D. Mintz, A. Rosenfeld, and D.Y. Kim. Robust regression methods for computer vision: A review. *International journal of computer vision*, 6(1):59–70, 1991.

[80] A. Mintz, D. Raicu, and J. Furst. Liver Segmentation Combining Gabor Filtering and Traditional Vector Field Snake. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, 2008.

[81] George L. Nemhauser and Laurence A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience, New York, NY, USA, 1988.

[82] T. Okada, R. Shimada, Y. Sato, M. Hori, K. Yokota, M. Nakamoto, Y. W. Chen, H. Nakamura, and S. Tamura. Automated segmentation of the liver from 3d ct images using probabilistic atlas and multi-level statistical shape model. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 10(Pt 1):86–93, 2007.

[83] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[84] Théodore Papadopoulo and Manolis I. A. Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. In *ECCV (1)*, pages 554–570, 2000.

[85] N. Paragios, O. Mellina-Gottardo, and V. Ramesh. Gradient vector flow fast geodesic active contours. volume 1, pages 67–73 vol.1, 2001.

[86] M. Pham, R. Susomboon, T. Disney, D. Raicu, and J. Furst. A comparison of texture models for automatic liver segmentation. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6512 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, March 2007.

[87] J. M. W. Rhys. A selection problem of shared fixed costs and network flows. *Management Science*, 17:220–207, 1970.

[88] Sami Romdhani, Shaogang Gong, and Ra Psarrou. A multi-view nonlinear active shape model using kernel pca. In *In BMVC*, pages 483–492, 1999.

[89] Laszlo Rusko, Gyorgy Bekes, Gabor Nemeth, and Marta Fidrich. Fully automatic liver segmentation for contrast-enhanced ct images. *Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pages 143–150, 2007.

[90] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. In *COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory*, pages 80–91, New York, NY, USA, 1998. ACM.

[91] Guenter Schmidt, Maria Athelogou, Ralf Schoenmeyer, Rene Korn, and Gerd Binnig. Cognition network technology for a fully automated 3d segmentation of liver. *Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pages 125–133, 2007.

[92] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, 1998.

[93] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[94] Dieter Seghers, Pieter Slagmolen, Yves Lambelin, Jeroen Hermans, Dirk Loeckx, Frederik Maes, and Paul Suetens. Landmark based liver segmentation using local shape and local intensity models. *Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pages 135–142, 2007.

[95] J. A. Sethian. Level set methods and fast marching methods - evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. In *Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1998.

[96] P. D. Seymour. Decomposition of regular matroids. *J. Combin. Theory*, 1980.

[97] Shiguang Shan, Bo Cao, Yu Su, Laiyun Qing, Xilin Chen, and Wen Gao. Unified principal component analysis with generalized covariance matrix for face recognition. *CVPR'08*, pages 1–7, June 2008.

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

[99] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. Technical Report 8, 2000.

[100] Greg Slabaugh and Gozde Unal. Graph cuts segmentation using an elliptical shape prior. In *ICIP*, volume 2, pages 1222–1225, 2005.

[101] C.V. Stewart. Robust parameter estimation in computer vision. *SIAM review*, pages 513–537, 1999.

[102] M. Styner, K. Rajamani, L. P. Nolte, G. Zsemlye, G. Szekely, C. Taylor, and R. H. Davies. Evaluation of 3d correspondence methods for model building. In *IPMI*, pages 63–75, July 2003.

[103] R. Susomboon, D.S. Raicu, and J.D. Furst. A hybrid approach for liver segmentation. *Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pages 151–160, 2007.

[104] G. J. Edwards T. F. Cootes and C. J. Taylor. Active appearance models. *IEEE Trans. on PAMI*, 23:681–685, January 2001.

[105] Hans Henrik Thodberg. Minimum description length shape and appearance models. In *IPMI*, pages 51–62, 2003.

[106] K. Truemper. Unimodular matrices of flow problems with additional constraints. *Networks*, 7:343–358, 1977.

[107] Ledyard Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, September 1966.

[108] C. J. Twining and C. J. Taylor. Kernel principal component analysis and the construction of non-linear active shape models. In *In BMVC*, pages 23–32, 2001.

[109] M. Üzümcü, A. F. Frangi, J. H. C. Reiber, and B. P. F. Lelieveldt. Independent component analysis in statistical shape models. In M. Sonka and J. M. Fitzpatrick, editors, *SPIE*, volume 5032 of *SPIE*, 2003.

[110] Eva van Rikxoort, Yulia Arzhaeva, and Bram van Ginneken. Automatic segmentation of the liver in computed tomography scanswith voxel classification and atlas matching. *Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge*, pages 101–108, 2007.

[111] M. A. O. Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *ECCV '02*, pages 447–460, London, UK, 2002. Springer-Verlag.

[112] M.A.O. Vasilescu and D. Terzopoulos. Multilinear image analysis for facial recognition. *ICPR'02*, 2:511–514 vol.2, 2002.

[113] M.A.O. Vasilescu and D. Terzopoulos. Multilinear independent components analysis. *CVPR'05*, 1:547–553 vol. 1, June 2005.

[114] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[115] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:511, 2001.

[116] Paul Viola, Michael J. Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. *Computer Vision, IEEE International Conference on*, 2:734, 2003.

[117] Hongcheng Wang and Narendra Ahuja. A tensor approximation approach to dimensionality reduction. *Int. J. Comput. Vision*, 76(3):217–229, 2008.

[118] W. Xiong, J. Zhou, Q. Tian, J. J. Liu, Y. Qi, W. K. Leow, T. Han, and S.-c. Wang. Performance benchmarking of liver CT image segmentation and volume estimation. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6919 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, April 2008.

[119] Ning Xu, Narendra Ahuja, and Ravi Bansal. Object segmentation using graph cuts based active contours. *Comput. Vis. Image Underst.*, 107(3):210–224, 2007.

[120] Jian Yang, David Zhang, Alejandro F. Frangi, and Jing yu Yang. Two-dimensional pca: A new approach to appearance-based face representation and recognition. *IEEE Transactions on PAMI*, 26(1):131–137, 2004.

[121] Yuri Boykov Yuri Y. Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *EMMCVPR '01*, pages 359–374, 2001.

[122] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2.

[123] Yun Zeng, Dimitris Samaras, Wei Chen, and Qunsheng Peng. Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in n-d images. 2007.

[124] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu. Fast automatic heart chamber segmentation from 3d ct data using marginal space learning and steerable features. In *11th International Conference on Computer Vision (ICCV). Rio de Janeiro, Brazil. October 2007*, pages 84–0.

[125] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. 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*, 27(11):1668–1681, 2008.

[126] Y. Zheng, B. Georgescu, H. Ling, M. Suehling, and D. Comaniciu. Constrained Marginal Space Learning for Efficient 3D Anatomical Structure Detection in Medical Images. In *CVPR'09*.

[127] S.C. Zhu. Embedding gestalt laws in markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1170–1187, 1999.

[128] Jie Zhu-Jacquot and R. Zabih. Segmentation of the left ventricle in cardiac mr images using graph cuts with parametric shape priors. pages 521–524, 31 2008-April 4 2008.