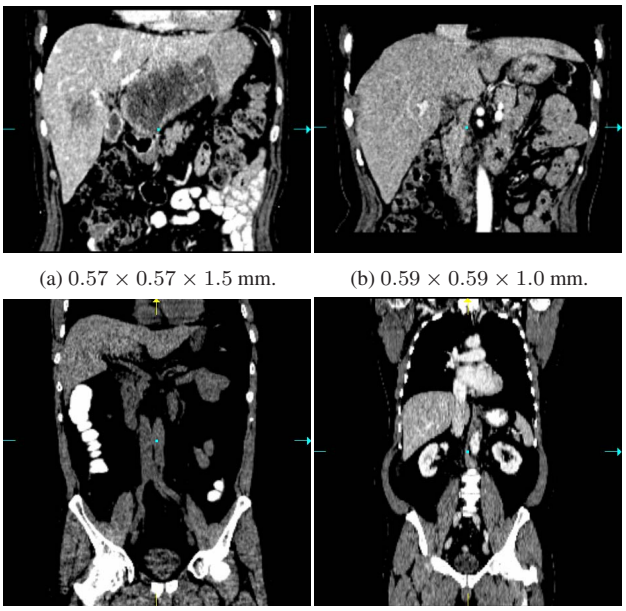
Hierarchical, Learning-based Automatic Liver Segmentation

Haibin Ling[1], S. Kevin Zhou[1], Yefeng Zheng[1], Bogdan Georgescu[1]
Michael Suehling[2], and Dorin Comaniciu[1]
[1]Integrated Data Systems Department [2]Siemens Medical Solutions
Siemens Corporate Research, USA Germany
{haibin.ling, shaohua.zhou, yefeng.zheng, bogdan.georgescu,michael.suehling dorin.comaniciu}@siemens.com

Abstract

In this paper we present a hierarchical, learning-based approach for automatic and accurate liver segmentation from 3D CT volumes. We target CT volumes that come from largely diverse sources (e.g., diseased in six different organs) and are generated by different scanning protocols (e.g., contrast and non-contrast, various resolution and position). Three key ingredients are combined to solve the segmentation problem. First, a hierarchical framework is used to efficiently and effectively monitor the accuracy propagation in a coarse-to-fine fashion. Second, two new learning techniques, marginal space learning and steerable features, are applied for robust boundary inference. This enables handling of highly heterogeneous texture pattern. Third, a novel shape space initialization is proposed to improve traditional methods that are limited to similarity transformation. The proposed approach is tested on a challenging dataset containing 174 volumes. Our approach not only produces excellent segmentation accuracy, but also runs about fifty times faster than state-of-the-art solutions [7, 9].



(a) $0.57 \times 0.57 \times 1.5$ mm. (b) $0.59 \times 0.59 \times 1.0$ mm.
(c) $1.0 \times 1.0 \times 1.0$ mm. (d) $0.89 \times 0.89 \times 5.0$ mm.

Figure 1. Examples of CT volumes in our dataset and their resolutions. Note the variance in the intensity, position of livers, volume dimensions. Hounsfield unit (HU) window [-15, 155] is used for visualization purpose (same for other figures).

1. Introduction

Liver analysis plays an important role in the therapeutic strategies for hepatic diseases. Segmentation of a liver from a three dimensional computed tomography (CT) volume often serves as the first step in image-based hepatic studies and continuously attracts research attention [19, 6, 16, 11, 1, 13, 7, 5, 18]. Despite a large body of literature, fully automatic liver segmentation from a 3D volume remains a challenge, due to the large variations in liver shapes and in the intensity pattern inside and along liver boundaries.

Unlike in previous work, the CT volumes in our study come from patients with diseases in different organs, and are scanned under largely diverse protocols. Specifically, there are several diversities in our dataset: 1) The volumes have diseases in six different organs (the liver counts only one third); 2) Some volumes are enhanced by contrast agent while some not; 3) The volume dimension and liver position in the volume substantially vary; 4) The inter-slice resolution changes from 1.0 mm to 5.0 mm. Some examples are shown in Figures 1 and 6. More details can be found in Sec. 6.1. These diversities increase the variability of livers in both shape and texture patterns, and therefore render the problem more challenging. Furthermore, the automatic detection (or initialization) is made more difficult especially because of 3). To the best of our knowledge, there is no previous work that handles all the above issues in an automatic and robust fashion (cf. Sec. 2 and Table 2).

We propose an automatic segmentation approach that ad-

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

dresses all the challenges mentioned above. There are three major contributions in our system. First, the system uses a hierarchical shape representation, which efficiently and effectively handles the shape inference and accuracy propagation. Second, a learning-based boundary localization technique is utilized. By using this technique, the system not only achieves accurate boundary responses, but also becomes reliable to the heterogeneous intensity patterns. In addition, a liver surface is decomposed into patches according to surrounding anatomic structures, and patch dependent classifiers are used for further improvement. Third, a subspace shape initialization is introduced to improve traditional pose initialization. The proposed approach is tested on a database containing 174 3D CT volumes. Our method demonstrates excellent performance, achieving an accuracy of 1.76±0.92 mm (or 1.59±0.49 mm after excluding outliers). In addition, it is more than fifty times faster than state-of-the-art approaches (e.g., [7] and [9]).

The rest of the paper is organized as follow. Sec. 2 summarizes related work. Sec. 3 introduces the hierarchical representation for liver shapes. Sec. 4 describes the detection and robust initialization in our approach. Then, Sec. 5 describes the hierarchical and learning-based boundary refinement. Sec. 6 gives the experiments on the challenging liver database. Finally, Sec. 7 concludes the paper.

2. Related Work

Early work on liver segmentation mainly focused on two dimensional images and often required manual or semi-manual interaction. For example, Gao et al. [6] presented a semiautomatic liver segmentation system for CT data that combines several image processing techniques. Some recent works on 3D liver segmentation systems were built from the processing of 2D slices. Soler et al. [16] proposed a framework for a complete anatomical, pathological and functional segmentation of the liver from CT scans. Liu et al. [13] proposed a semiautomatic method for liver segmentation from contrast-enhanced CT images using a gradient vector flow field [20]. Florin et al. [5] used the level-set techniques for 2D key slice segmentation and then a 3D shape was interpolated from 2D contours.

Some recent work used probabilistic atlas for the task [24, 14]. While convenient and fast, these methods deal with volumes with the same dimension and are roughly aligned (usually acquired by the same scanning protocol). For example, volumes used in both [24] and [14] are of fixed size and resolution. Consequently, these approaches have difficulty in generalizing to datasets such as the one in our experiment.

The work [11, 9, 7, 14] using *statistical shape model* (SSM) [4] are the most related to ours. Lamecker et al. [11] modeled liver surface with SSM and applied it for segmentation tasks. It was later extended [9] with some detailed processing on boundary intensity. Another work using SSM is proposed by Heimann et al. [7], where an evolution algorithm is applied for better shape initialization. The SSM used in [14] is in a multi-level fashion, where a liver surface is recursively decomposed into subsurfaces (patches) and each subsurface is attached with a local SSM.

More recently, several approaches have been tested in the MICCAI workshop on segmentation challenge [18]. The SSM based approach in [9] achieved the best accuracy, and a region growing approach in [15] ranked the second.

Our work is different from previous approaches in several aspects. First, a hierarchical shape representation is presented. Note that our hierarchy lies in the shape (and volume) representation while the "multilevel" in [14] means multilevel patch decomposition. Second, learning based techniques are used for boundary deformation as well shape detection. Third, we use the marginal space learning [22] for the pose and subspace initialization. In addition, as mentioned in the introduction, none of previous works has been tested on a heterogeneous dataset as ours. A summary of some related works is given in Table 2.

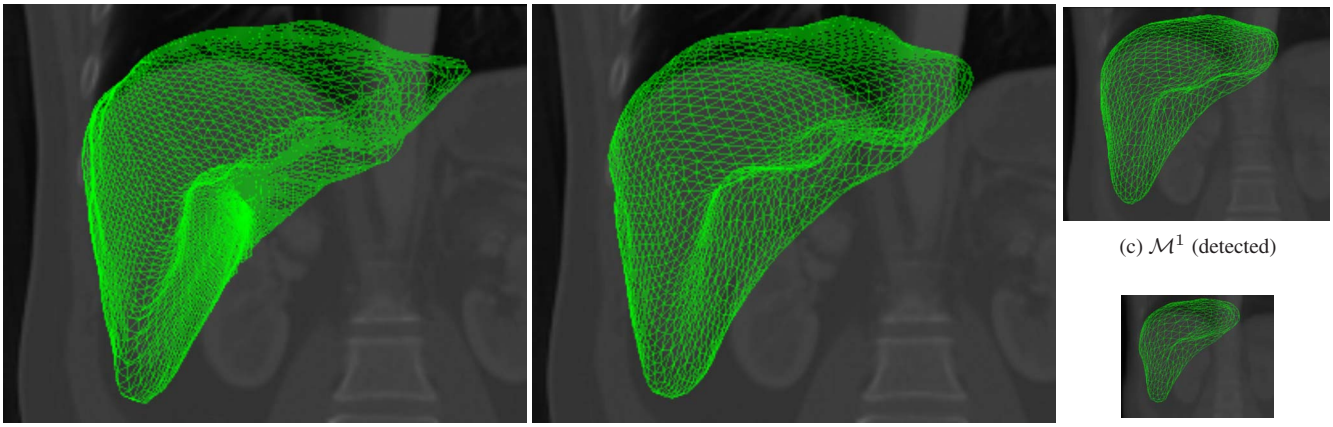
3. Hierarchical Representation for Livers

3.1. Mesh Representation

We represent the shape of a liver by a closed triangle mesh $\mathcal{M}(P,T)$, where $P = \{p_i \in \mathbb{R}^3\}_{i=1}^n$ is the set of n mesh points, and $T = \{t_i \in \mathbb{Z}_+^3\}_{i=1}^m$ is the set of m triangle indices. A canonical mapping from the liver surface to a unit sphere is built. Specifically, a continuous liver surface S can be parameterized as $S(\theta,\varphi) : [0,2\pi) \times [-\pi/2,\pi/2] \rightarrow \mathbb{R}^3$. With this representation, a dense mesh is generated by uniformly sampling the space of spherical coordinates (θ,φ). In other words, the point set P is generated by first cutting S into half-circle slices and then uniformly sampling along each half-circle slice. Accordingly, the triangle set T is built by connecting points in neighboring slices sequentially. In practice, we use a dense mesh for a liver shape with 120 half-slices and 42 points per slice. This mesh serves as a reference mesh that provides point correspondence and spherical coordinates. For this reason, it is called a *base mesh* and denoted as $\mathcal{M}^b = (P^b, T^b)$. An example of base mesh is shown in Fig. 2(a).

The base mesh is inappropriate for direct use. First, the dimension of its shape space is too high for effective shape modeling. Second, it is computationally expensive for boundary deformation. Third, it is not efficient because it is not uniformly sampled along surfaces. For example, the sampling around the lung region is much denser than that around the cusp area.

To avoid these problems, a new mesh $\mathcal{M}^0 = (P^0, T^0)$ is sampled from $\mathcal{M}_b$ (approximately) uniformly along mesh surfaces, i.e., in the sense of geodesic distance. This is



(a) $\mathcal{M}^b$ (annotated groundtruth)

(b) $\mathcal{M}^0$ (detected result)

(c) $\mathcal{M}^1$ (detected)

(d) $\mathcal{M}^2$ (detected)

Figure 2. Mesh pyramid.

done through a standard mesh simplification routine [8]. We denote this downsampling process with operator $\downarrow$, i.e., $\mathcal{M}^0 = \downarrow(\mathcal{M}^b)$. One example is shown in Fig. 2 (b).

Note that P^0 is actually a subset of P^b. Therefore, it can be written as $P^0 = P^b(I^0) \doteq \{p_i^b \in P^b : i \in I^0\}$, where $I^0 \subset \{1, \ldots, |P^b|\}$ is called index set for $\mathcal{M}^0$. The index set implicitly maintains correspondence from a "sparse" mesh to the unit sphere. This is important because it is used to compute the spherical coordinates that are needed for boundary inference (Sec. 5).

3.2. The Hierarchical Shape Model

Starting from the mesh $\mathcal{M}^0$, a mesh pyramid is built by recursively applying the downsampling process mentioned above. As a result, our hierarchical mesh representation includes a dense mesh $\mathcal{M}_b$ and a mesh pyramid $\{\mathcal{M}^l\}_{l=0}^L$, where $L+1$ is the number of pyramid layers ($L = 2$ in our implementation). Specifically, we have

$$\begin{cases} \mathcal{M}^0 &= \downarrow(\mathcal{M}^b) \\ \mathcal{M}^l &= \downarrow(\mathcal{M}^{l-1}) \quad l = 1 \ldots L \\ P^l &= P^b(I^l) \quad l = 0 \ldots L \,. \end{cases} \tag{1}$$

In practice, about half of mesh points are kept during downsampling. That is, $n_l \simeq n_{l-1}/2$, where $n_l = |P^l|$ denoting the number of points for layer l. A volume pyramid is built such that meshes at different layers correspond to volumes at different resolutions. An example of mesh pyramid is illustrated in Fig. 2.

Using the hierarchical shape representation, we build statistical shape models [4] for each layer. Specifically, the model for layer l contains a pair $(\mu^l, V^l = [\nu_1^l, \ldots, \nu_{k_l}^l])$, where $\mu^l \in \mathbb{R}^{3n_l}$ is the mean shape, and $V^l \in \mathbb{R}^{3n_l \times k_l}$ contains k_l modes that capture sufficiently large shape variations at layer l. In summary, our hierarchical model is denoted by $(\mu^l, V^l, P^l, T^l),\ l = 0, ..., L$.

4. Detection and Initialization

In this section we describe our approach to automatic liver detection and initialization. Both steps are performed on the coarsest layer. This not only helps improving the efficiency, but also implicitly captures more global information that is important for the initialization purpose.

4.1. Liver Detection Via Marginal Space Learning

The task of detection is to find the best pose of a liver in a given volume vol. The pose of a liver is determined by nine parameters, $\mathtt{p} = (p_1, p_2, p_3), \theta = (\theta_1, \theta_2, \theta_3), \mathtt{s} = (s_1, s_2, s_3)$, corresponding to location, orientation, and (anisotropic) scale respectively. Using a probabilistic framework, our task can be formulated as

$$(\hat{\mathtt{p}}, \hat{\theta}, \hat{\mathtt{s}}) = \arg\max_{\mathtt{p},\theta,\mathtt{s}} \Pr(\mathtt{p}, \theta, \mathtt{s}|vol) \,. \tag{2}$$

Solving equation (2) involves a search in a 9D space, which is too expensive in practice. Instead, an efficient inference scheme, *marginal space learning* (MSL) [22], is applied. Intuitively, MSL reduces the size of the searching space by marginal space inference and sequentially propagates to the whole space. In our task, the 9D parameter space is decomposed to three marginal spaces as follows:

$$\Pr(\mathtt{p}, \theta, \mathtt{s}|vol) = \Pr(\mathtt{p}|vol)\Pr(\theta|\mathtt{p}, vol)\Pr(\mathtt{s}|\theta, \mathtt{p}, vol) \,, \tag{3}$$

which is consistent with the decomposition used in [22].

To learn the marginal probabilities (i.e., $\Pr(\mathtt{p}|vol)$, $\Pr(\theta|\mathtt{p}, vol)$, and $\Pr(\mathtt{s}|\theta, \mathtt{p}, vol)$), the *probabilistic boosting tree* (PBT) [17] is used. Another choice is the probabilistic boosting network [21]. Moreover, 3D Haar features are used for location detection and steerable features are used for orientation and scale inferences. The detected shape can be described as:

$$\mathtt{x} = f_{\hat{\mathtt{s}}} \circ f_{\hat{\theta}} \circ f_{\hat{\mathtt{p}}}(\mu^L) \,, \tag{4}$$

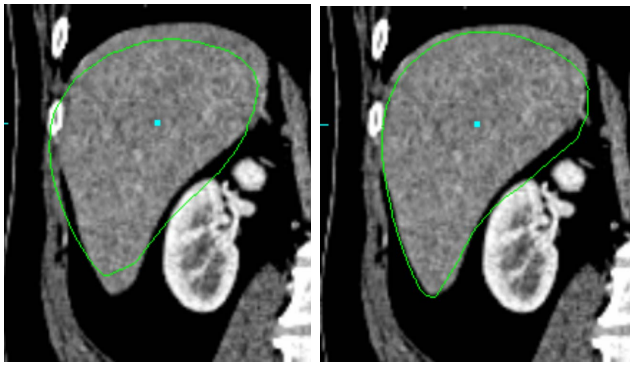


Figure 3. Initialization. Left: after pose detection. Right: after shape space initialization.

where $f_{\hat{\mathtt{s}}}$, $f_{\hat{\theta}}$, and $f_{\hat{\mathtt{p}}}$ denote the scaling, rotation, and translation using the detected parameters respectively. In other words, we simply put the mean shape in the detected box. An example detection is shown in Fig. 3.

4.2. Subspace Initialization

SSM-based methods usually start boundary refinement right after the pose detection (or manual initialization). However, as noted in [7], for highly deformable shapes such as livers, the pose detection can be improved by further initialization [23].

We propose using learning techniques for the inference of shape initialization. The basic idea is to learn the coefficients corresponding to the first three shape components[1]. In other words, the initialization now estimates another three parameters $\mathtt{c} = (c_1, c_2, c_3)$ after pose detection:

$$\mathbf{x} = f_{\mathtt{s}} \circ f_{\theta} \circ f_{\mathtt{p}} \left(\mu^L + \sum_{i=1}^{3} c_i \nu_i^L \right). \tag{5}$$

In the MSL framework, this can be viewed as an additional inference of a conditional probability $\Pr(\mathtt{c}|\mathtt{s}, \theta, \mathtt{p}, vol)$. The PBT is used again for learning. The problems left are how to obtain training samples and what features to use.

The training samples are extracted from the 3D parameter spaces $\mathcal{C} \doteq [c_{1,0}, c_{1,1}] \times [c_{2,0}, c_{2,1}] \times [c_{3,0}, c_{3,1}]$, where $[c_{i,0}, c_{i,1}]$ is the range for coefficient c_i determined by the corresponding eigenvalue. Positive samples in $\mathcal{C}$ are extracted from annotated meshes by standard projection in the subspace. Let $\mathcal{C}^+$ be positive sample set. The negative sample set $\mathcal{C}^-$ is built by first uniformly sampling the space $\mathcal{C}$, and then eliminating any samples that are within a certain distance to positive samples in $\mathcal{C}^+$.

Given the shape model (μ^L, V^L) and warping parameters $(\mathtt{p}, \theta, \mathtt{s})$, each $\mathtt{c} \in \mathcal{C}$ corresponds to a mesh, denoted as $\mathcal{M}_{\mathtt{c}}^L$. The feature set for $\mathtt{c}$ should be robust to the warping process. For this purpose, we use intensity-based features as in [22], but the features are sampled along the surface of mesh $\mathcal{M}_{\mathtt{c}}^L$.

Fig. 3 shows an example before and after subspace initialization. The improvement is obvious. Furthermore, compared to the evolution algorithm used in [7] that requires about six minutes, our solution runs as fast as about four seconds using similar machine configurations (Sec. 6).

[1] More coefficients can be included, but we did not observe significant improvement after three.

5. Boundary Refinement

After initialization, boundary refinement is used for accurate boundary localization. There are three key components in our refinement method: hierarchical scheme, probabilistic boundary response, and patch-based boundary classifiers.

5.1. Hierarchical Boundary Refinement

Starting from the coarsest layer $\mathcal{M}^L$, the mesh at current layer is first refined and then upsampled to a finer layer. The procedure continues till the finest layer $\mathcal{M}_0$ is reached. Algorithm 1 summarizes the algorithm, where Π^l denotes the subspace projection at the l-th layer.

There are several advantages by this hierarchical scheme. First, the coarse-to-fine fashion helps to achieve reliability as well as accuracy. In our application, the refinement at a coarser level gathers more information from a larger surrounding (note that volumes are organized hierarchically too). As a result, the boundary accuracy improves gradually and in a steady fashion. This can be seen in our experiments summarized in Table 3. Second, efficiency is improved as a "standard" benefit from the hierarchical scheme. This is verified by the fact that our solution is more than fifty times faster than state-of-the-art solutions. Third, the framework introduces the flexibility of treating layers differently. For example, as shown in the following subsection, at the finest layer, we use an array of boundary classifiers for different regions of a liver (see 5.3).

For the upsampling between layers, the *thin plate spline* (TPS) warping [2] is used. Given two point sets with correspondence between them, TPS finds a nonlinear warping by minimizing a second order "bending energy". In our task, mesh points at a coarse level correspond to a subset of mesh points at the finer level. Therefore, we can use the mean shapes at different levels for the warping. For a given point $\mathbf{x} \in \mathbb{R}^3$, its TPS warping has the following formula

$$\mathcal{T}(\mathbf{x}; \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{Q}) = \left(a_i \mathbf{x} + b_i + \sum_{j=1}^{n_a} c_{i,j} \parallel \mathbf{x} - \mathbf{q}_j \parallel \right)_{i=1}^{3},$$

where subscript i denote the i-th coordinate; $\mathbf{Q} = \{\mathbf{q}_j \in \mathbb{R}^3\}_{j=1}^{n_a}$ is the anchoring point set, and $\mathbf{a} = (a_1, a_2, a_3)$, $\mathbf{b} = (b_1, b_2, b_3)$, $\mathbf{c} = \{c_{i,j}\}$ are warping parameters.

Algorithm 1 Hierarchical Mesh Refinement

$\mathcal{M}^L(P^L, T^L) \leftarrow$ initialization.
for $l = L$ down to 0 **do**
  for $i = 1$ to i_{max} **do** {/* iteration for i_{max} times*/}
    for $\mathbf{p} \in P^l$ **do** {/* local boundary refinement */}
      $Q_p \leftarrow$ candidate points close to $\mathbf{p}$.
      $\mathbf{p} \leftarrow \arg\max_{\mathbf{q} \in Q_p} \Pr(bdry|\mathbf{q}, vol^l)$.
    end for
    $P^l \leftarrow \Pi^l(P^l)$. {/* subspace projection */}
  end for
  if $l > 0$ **then** {/* upsampling with TPS */}
    Estimate TPS parameters $(\mathbf{a}, \mathbf{b}, \mathbf{c})$.
    $P^{l-1} \leftarrow \mathcal{T}(\mu^{l-1}; \mathbf{a}, \mathbf{b}, \mathbf{c}, \mu^l)$.
  end if
end for

Specifically, for the mesh $\mathcal{M}^l = (P^l, T^l)$, the parameters $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ are estimated by warping from μ^l to P^l. Then P^l is upsampled to get P^{l-1} by $\mathcal{T}(\mu^{l-1}; \mathbf{a}, \mathbf{b}, \mathbf{c}, \mu^l)$.

5.2. Learning Based Boundary Detection

A key issue in boundary refinement is boundary localization, which usually involves locally searching around current shape boundaries. Specifically, for a current boundary point $\mathbf{p} \in P^l$, a candidate point set Q_p is formed by including points along the normal direction at $\mathbf{p}$ and within some distance. Then the point in Q_p with the maximum boundary probability[2] (response) is used to replace $\mathbf{p}$, i.e.,

$$\mathbf{p} \leftarrow \arg\max_{\mathbf{q} \in Q_p} \Pr(bdry|\mathbf{q}, vol^l) \ . \tag{6}$$

Previous works usually approximate the boundary response by simple checking gradients or intensity distribution along surface normals. However, the information gathered this way is not enough for our task because the texture pattern of livers has a large variability. When dealing with data from different scanning protocols, the variability is even larger.

To attack this problem, we decide to learn $\Pr(bdry|\mathbf{q}, vol^l)$ using PBT and steerable features, similarly to [22]. In addition, the spherical coordinates of mesh points are included as features. These coordinates provide important distinctive information because the intensity patterns around boundary are closely tied to their positions on the liver surface (see for example Fig. 4). By checking the learned PBT classifiers, we found that these coordinates have been frequently selected, which validates our idea. Computation of spherical coordinates is through the mapping from the base mesh $\mathcal{M}^b$ to a unit sphere. In particular, let $\mathbf{p}_i \in P^l$ be the i-th point in the l-th layer. Its spherical coordinates is determined by $sph(I_i^l)$, where

[2]Strictly speaking, it is also conditioned on the normal direction at $\mathbf{p}$.

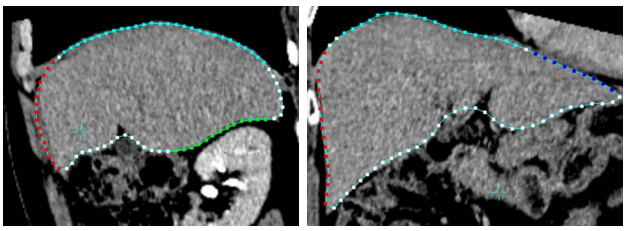


Figure 4. Patch clustering. Different colors indicate points from different patches.

I_i^l is the i-th index in index set I^l and $sph(.)$ is a function that converts an index to spherical coordinates via the base mesh (cf. Sec. 3.1).

5.3. Patch Based Boundary Refinement

The heterogeneity of texture pattern along liver boundaries suggests the use of patch dependent boundary classifiers. To this end, we decompose a liver surface to five patches: liver-lung, liver-heart, liver-kidney, liver-tissue, and liver-misc. We annotate twenty groundtruth (base) meshes. Two slices of such annotation are shown in Fig. 4. For the i-th point in the base mesh, its prior probability belonging to the k-th patch is estimated as the patch frequency $\mathbf{w}^i(k) = \frac{n_{i,k}}{n_{all}}$, where $n_{i,k}$ is the number of base meshes that its i-th point is annotated as belonging to the k-th patch; n_{all}=20; and $k = 1, ..., 5$ corresponds to the five different patches.

There are two schemes to use the prior information $\mathbf{w}^i$ when computing boundary responses (6): soft-patch and hard-patch. The soft-patch method computes a weighted probability as

$$\Pr(bdry|\mathbf{q}, vol) = \sum_{k=1}^{5} \mathbf{w}^{\mathbf{q}}(k) \Pr_k(bdry|\mathbf{q}, vol) \ , \tag{7}$$

where $\Pr_k$ is the learned conditional probability for the k-th patch, $\mathbf{w}^{\mathbf{q}}$ is the prior probability at point $\mathbf{q}$.

In contrast, the hard-patch scheme takes only the response from the patch with maximum prior probability, i.e.

$$\Pr(bdry|\mathbf{q}, vol) = \Pr_{\hat{k}(\mathbf{q})}(bdry|\mathbf{q}, vol) \ , \tag{8}$$

where $\hat{k}(\mathbf{q}) \doteq \arg\max_{1 \le k \le 5} \mathbf{w}^{\mathbf{q}}(k)$ is precomputable.

While the soft-patch scheme sounds more natural than the hard-patch one, there is no significant difference observed in our experiments. There are two reasons for this. First, the patch decomposition of liver surfaces is relatively stable. Second, when training classifiers for each patch, we also include sample points from its neighborhood. Considering its apparent speed benefit, the hard-patch scheme is chosen in our final system.

Note that, the patch-dependent boundary classifiers are used only for the finest layer due to the sparseness of meshes at coarse layers and the lack of training samples.

Table 1. Disease distribution of the dataset.

diseased organ	liver	colon	lymphnode	kidney	pancreas	peritoneum
percentage	33%	39%	19%	4%	4%	1%

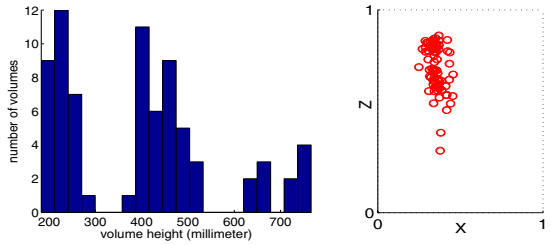


Figure 5. Left: the distribution of volume sizes of our dataset. Right: the distribution of liver locations. Each circle indicates the normalized x and z coordinates of a liver centroid (i.e., a volume dimension is normalized to $1 \times 1 \times 1$).

6. Experiments

6.1. Dataset

Our database contains 174 3D CT volumes, each with an annotated groundtruth dense mesh. As mentioned in the introduction, the dataset is very challenging in that the volumes come from largely diverse sources. In particular, the patients have diseases in six different organs (cf. Table 1) and therefore are often scanned under different protocols. This heterogeneity causes a large variation in both shape deformation and texture patterns of livers. Moreover, diagnosis of different diseases often request different contrast agent to be injected into the patient, or no contrast at all. For example, in our database, most volumes with liver diseases are enhanced with contrast agent, while those with colon diseases are usually not.

Due to the different scanning protocols, the volumes have various dimensionality: the inter-slice resolution varies from 1.0 mm to 5.0 mm; the number of slices varies from 105 to 524; the actual volume height varies from 183 mm to 766 mm. Moreover, the position of livers changes a lot, which presents significant challenges to atlas based approaches such as [24, 14]. The distributions of volume heights and liver center positions are shown in Fig. 5. Some example volumes are shown in Figures 1 and 6.

6.2. Evaluation

Two experiments are conducted for evaluation. The first experiment uses the whole dataset for both training and testing. The second one uses five-fold cross validation. Errors are measured using the average symmetric surface distance.

Experiment I: testing with all volumes. The purpose of this experiment is two-fold. First, it aims to study the limit of our approach, which provides a reasonable expectation when there are enough training samples. Second, it illustrates how the hierarchical scheme helps in the procedure.

Table 3. Performance using all 174 volumes. The fourth and fifth columns show the average symmetric surface errors in millimeters. The last column shows the average running time of the corresponding stage (initialization is included in layer 2).

Layer	# points	Voxel size	After init. or upsamping	After mesh refinement	Run time (seconds)
2	602	6 mm	5.29±1.49	2.52±0.43	7.8
1	1202	3 mm	3.11±0.45	1.82±0.33	1.6
0	2402	varies	2.01±0.32	1.26±0.37	2.2

The average segmentation performance at different stages are shown in Table 3.

The table shows clearly how the segmentation accuracy increases in the hierarchical framework. At the coarser layers, our system efficiently reduces the error to nearly a half-voxel precision. The results are then propagated to finer layers for further refinement. At the final layer, the accuracy reaches 1.26 millimeter, which is smaller than the average inter-slice resolution of the database. The average running time for one volume, including all steps, is around 12 seconds (on an Intel 3.2 GHz processor). This is much faster than many state-of-the-art solutions (cf. Table 2).

Experiment II: cross validation. In most previous studies, liver segmentation approaches were evaluated by dividing the dataset into training and testing sets. This can be easily biased given the limited number of samples (cf. Table 2) and the large shape and texture variation of livers. To give a thorough evaluation, we conduct a five-fold cross validation on 75 volumes selected from the dataset[3]. Specifically, the dataset is divided to five sets, each containing 15 volumes. Each time one set is chosen as the testing set and the rest as the training set (for both the shape model and boundary classifiers). This is done for five times and the average performance is reported. The mean error measured in the average symmetric surface distance is 1.76±0.99 mm, and the median is 1.45 mm. For comparison reason, after removing five outliers (similar to [7]), the mean error becomes 1.59±0.50 mm, and the median becomes 1.38 mm. Some typical segmentation results are shown in Fig. 6.

Comparison to previous works. It is difficult to directly compare different liver segmentation approaches due to the use of different datasets as well as different annotations. That said, it is worth summarizing the previous experiments to comprehend the status of the study. Table 2 summarizes recent works of fully automatic liver segmentation with reported average symmetric surface distances[4] along with datasets used. The table shows clearly that our precision is among the the best reported. Furthermore, compared to the two methods [9, 7] with similar reported precisions, our approach have two apparent strengths. First, the dataset used in our experiments has more diversities than previous tested datasets. Second, our method runs more than fifty

[3] These are all the volumes we had when we conduct cross-validation.

[4] We exclude [24] because it uses different error measures.

Table 2. Comparison to recent automatic liver segmentation experiments, sorted by the reporting time. Note: ♮Experiments from the MICCAI liver segmentation challenge [18]. The online testing scores are shown (http://www.sliver07.org/miccai.php). The top two scores are included. †The shape model is built from additional 43 volumes as in [11]. ‡Reported scores are after excluding outliers.

Method	Mean Error (mm)	Run time	# volume involved	# volume tested	contrast	Interslice reso.	# slice
Soler et al. [16]	2‡	15min	35	35	Yes	2-3	n/a
Lamecker et al. [11]	2.3±0.3	n/a	43	33	Yes	5	n/a
Heimann et al. [7]	1.6±0.5‡	10 min	86	54	Yes	3	60-130
Okada et al. [14]	2.15	n/a	28	8	Yes	2.5	159
♮Ruskó et al. [15]	2.01	56 sec	40	10	Yes	1-3	n/a
♮Kainmueller et al. [9]	1.37	15 min	40(+43)†	10	Yes	1-3	n/a
Our approach	1.59±0.50‡ / 1.76±0.99	12 sec	174	174	Mixed	1-5	105-524

times faster.

7. Conclusion

In this paper we propose a hierarchical, learning based approach for automatic liver segmentation from 3D CT volume. We target on general data from patients with difference diseased organs and scanned under different protocols. Despite the challenges, our approach demonstrates excellent performance in accuracy and runs more than fifty times faster than state-of-the-art solutions.

There are two important future directions along our study. First, more data should be included to achieve further improvement. Second, post-processing steps can potentially achieve better accuracy. In particular, graph theory could be used for boundary adjustment [7, 12, 3].

References

[1] J. M. Blackall, G. P. Penney, A. P. King, and D. J. Hawkes. "Alignment of sparse freehand 3-D ultrasound with preoperative images of the liver using models of respiratory motion and deformation". *IEEE Trans. Med. Imaging*, 24(11):1405-1416, 2005. 1

[2] F. Bookstein. "Principal Warps: Thin-Plate-Splines and Decomposition of Deformations", *IEEE Trans. PAMI*, 11(6):567-585, 1989. 4

[3] Y. Boykov and G. Funka-Lea. "Graph Cuts and Efficient N-D Image Segmentation." *IJCV*, 70(2):109-131, 2006. 7

[4] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models - their training and application", *Comput. Vis. Image Underst.*, 61(1):38-59, 1995. 2, 3

[5] C. Florin, N. Paragios, G. Funka-Lea, and J. Williams. "Liver Segmentation Using Sparse 3D Prior Models with Optimal Data Support". *IPMI*, 2007. 1, 2

[6] L. Gao, D. G. Heath, B. S. Kuszyk, and E. K. Fishman. "Automatic liver segmentation technique for three-dimensional visualization of CT data", *Radiology*. 201(2):359-64, 1996. 1, 2

[7] T. Heimann, S. Münzing, H.-P. Meinzer, I. Wolf. "A Shape-Guided Deformable Model with Evolutionary Algorithm Initialization for 3D Soft Tissue Segmentation", *IPMI*, 2007. 1, 2, 4, 6, 7

[8] H. Hoppe. "Progressive Meshes". *SIGGRAPH*, 99-108, 1996. 3

[9] D. Kainmueller, T. Lange, and H. Lamecker. "Shape Constrained Automatic Segmentation of the Liver based on a Heuristic Intensity Model", *MICCAI Wshp. 3D Segmentation in the Clinic: A Grand Challenge*, 2007. 1, 2, 6, 7

[10] H. Lamecker, T. Lange, and M. Seebass. "A Statistical Shape Model for the Liver". *MICCAI*, 2:421-427, 2002.

[11] H. Lamecker, T. Lange, and M. Seebaee. "Segmentation of the Liver using a 3D Statistical Shape Model", *ZIB Tech. Report*, 2004. 1, 2, 7

[12] K. Li, X. Wu, D. Z. Chen, and M. Sonka. "Optimal Surface Segmentation in Volumetric Images – A Graph-Theoretic Approach." *PAMI*, 28(1)119-134, 2006 7

[13] F. Liu, B. Zhao, P. K. Kijewski, L. Wang, and L. H. Schwartz. "Liver segmentation for CT images using GVF snake", *Medical Physics*, 32(12):3699-3706, 2005. 1, 2

[14] T. Okada, R. Shimada, Y. Sato, M. Hori, K. Yokota, M. Nakamoto, Y. Chen, H. Nakamura, and S. Tamura. "Automated segmentation of the liver from 3D CT images using probabilistic atlas and multilevel statistical shape model", *MICCAI*, 2007. 2, 6, 7

[15] L. Ruskó, G. Bekes, G. Németh, and M. Fidrich. "Fully automatic liver segmentation for contrast-enhanced CT images", *MICCAI Wshp. 3D Segmentation in the Clinic: A Grand Challenge*,2007. 2, 7

[16] L. Soler, H. Delingette, G. Malandain, J. Montagnat, N. Ayache, C. Koehl, O. Dourthe, B. Malassagne, M. Smith, D. Mutter, and J. Marescaux, "Fully automatic anatomical, pathological, and functional segmentation from CT scans for hepatic surgery," *Computer Aided Surgery*, 6(3):131-142, 2001. 1, 2, 7

[17] Z. Tu, "Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering", *ICCV*, II:1589-1596, 2005. 3

[18] B. van Ginneken, T. Heimann, and M. Styner. "3D Segmentation in the Clinic: A Grand Challenge", *MICCAI Wshp. 3D Segmentation in the Clinic: A Grand Challenge*, 2007. 1, 2, 7

[19] C-M Wu, Y-C Chen, and K-S Hsieh. "Texture Features for Classification of Ultrasonic Liver Images". *IEEE Trans. Med. Imaging*, 11(2):141-152, 1992. 1

[20] C. Xu and J. L. Prince. "Snakes, shapes, and gradient vector flow". *IEEE Transactions on Image Processing*, 7(3):359-369, 1998. 2

[21] J. Zhang, S. Zhou, L. McMillan, and D. Comaniciu. "Joint real-time object detection and pose estimation using probabilistic boosting network", *CVPR*, 2007. 3

[22] 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", *ICCV*, 2007. 2, 3, 4, 5

[23] S. Zhou, D. Comaniciu. "Shape regression machine," *IPMI*, 2007. 4

[24] X. Zhou, T. Kitagawa, T. Hara, H. Fujita, X. Zhang, R. Yokoyama, H. Kondo, M. Kanematsu, H. Hoshi. "Constructing a Probabilistic Model for Automated Liver Region Segmentation Using Non-contrast X-Ray Torso CT images", *MICCAI*, 2006. 2, 6

Figure 6. Typical results. From left to right: sagittal, coronal, and transversal slices. The errors (inter-slice resolutions, diseased organs) for from top to bottom: 1.09 (1.5, liver) mm, 1.30 (1.0, colon) mm, 1.63 (1.0, liver) mm, and 2.38 (5.0, lymphnode) mm.