Automatic Liver Segmentation from CT Images Using Latent Semantic Indexing

Chun-Yao Hsieh\textsuperscript{a}, Shyi-Chyi Cheng\textsuperscript{b}, Chin-Chun Chang\textsuperscript{a}, Chin-Lang Lin\textsuperscript{b}

\textsuperscript{a}Department of Computer Science & Engineering, National Taiwan Ocean University
2, Peining Rd., Keelung, Taiwan
csc@mail.ntou.edu.tw

\textsuperscript{b}Liver Research Unit, Chang Gung Memorial Hospital
222, Maijin Rd., Keelung, Taiwan
wn49792000@yahoo.com.tw

Abstract—In this paper, we present an indexing structure of data-driven cuboid patterns to speed up the process of liver detection and segmentation from computed tomography (CT) scans using the cube-based generalized Hough transform (CGHT). Most existing approaches to automatic liver segmentation from CT scans use a statistical shape model (SSM) integrated with a searching algorithm to recover the deformation. However, establishing the correspondences among landmark points of training shapes for the construction of the average shape of SSM remains as a challenge due to the high variation of liver shapes in CT scans. The proposed method is a fully automatic segmentation method that combines four steps. Firstly, a test CT volume is partitioned into multiple non-overlapped sub-volumes with each of them consisting of variable numbers of consecutive slices. Secondly, we locate the cube-based liver shapes in all sub-volumes via Hough voting and dynamic programming. Thirdly, we construct the basic 3D model through a level-set framework for liver shape segmentation. Finally, to introduce neighbors statistical analysis into the above model, we deform the 3D liver shape to overcome disturbances caused by noise and inhomogeneity. The MICCAI 2007 liver segmentation challenge datasets are used to verify the effectiveness of the proposed method. Experimental results demonstrate the good performance of the proposed method in terms of segmentation accuracy and execution speed.

I. INTRODUCTION

Automatic liver segmentation from medical images is the primary step of many computer-aided diagnosis (CAD) applications for hepatic diseases diagnosis, liver volume measurements, and surgery planning. Computed tomography (CT) volume is often used for liver segmentation, 3D rendering and subsequent liver vasculature analysis [1]. Although liver segmentation has been extensively studied, it remains as a challenging task due to the non-rigid shape of liver and ill-defined edges to characterize the CT images [2, 3]. In [4], based on well-organized datasets, a platform for testing and comparing different approaches was provided.

The advances on machine learning techniques offer many approaches of liver segmentation, which can be divided into three categories, i.e. interactive, semi-automatic and automatic schemes. Interactive and semi-automatic methods require a tedious user-feedback loop to complete the task. On the contrast, fully automatic methods focus on segmenting 3D liver in a CT volume without any time-consuming interference of users. This attracts radiologists’ attention since it saves the operator from both tedious and hard work in manually annotating CT volumes. Extensive and recent reviews on these methods can be found in [2-4]. In a nutshell, in the literature the approaches on liver segmentation use a variety of computer vision techniques including graph-based methods, level-set segmentation, statistical shape models, and atlas-based registration. A common disadvantage on these methods is that they focus on contrast-enhanced CT images and are limited to clinical use [5].

In the SILVER07 segmentation challenge [1, 4], the statistical shape model (SSM) based method [6] outperforms all compared methods. Cootes et al. [7] first presented the SSM-based segmentation framework which combines three steps: shape correspondence, shape representation, and deformation searching. The primary step to construct the average shape model is to establish correspondences among landmark points of all training shapes using a registration method [8]. Classical principal component analysis (PCA) method can be followed to compactly represent the shape model. In [9], the authors present an approach for automatic liver segmentation which first build the SSM of liver with the 3D GHT; then initializes the subspace of SSM; finally, deforms the shape model to adapt to liver contour through an optimal-surface detection approach based on graph theory. The method achieves higher accuracy compared with previous model based methods. However, both the memory and time complexity of the 3D GHT are very high.

Recent approaches on automatic liver segmentation on CT volumes focus on a top-to-down approach which combines the detection of liver area and the segmentation of liver contour [5]. In [10], the authors first use a spatial fuzzy c-mean clustering combining with anatomical prior knowledge to extract liver region automatically. Then a distance regularized level set [11] is used for refinement. In this work, every training CT volumes is first partitioned into multiple sub-volumes in which the symmetric patterns of non-overlapped 4x4x4 cubes are extracted analytically. In the training phase, all CT cubes in the training sub-volumes are inversely indexed using the latent-semantic indexing (LSI) [12] in terms of symmetric patterns. To achieve the goal of liver detection, the cube-based GHT (CGHT) model is used to exploit additional spatial relationships between latent codewords of LSI. The median of training CT volumes is then used as the template which is modelled by a sequence of CGHT models, where each of them characterizes the liver shape in a sub-volume of the template. In the detecting phase, instead of detecting, segmenting and tracking 2D liver shape in a single CT slice, the test CT volume is first partitioned into a sequence of CT sub-volumes; in fact, a sub-volume consists of variable numbers of consecutive slices. The dynamic time warping algorithm is next applied to optimally align the test CT sub-volumes with template CGHT models [15]. The symmetrical patterns in each
test sub-volume are then matched against the codewords of LSI of the corresponding CGHT model to detect the initial liver shape which is then refined by a level-set segmentation [11, 18]. Finally, to introduce neighbors statistical analysis into the above level-set models, we deform the segmented liver shapes to overcome disturbances caused by noise and inhomogeneity. Experimental results demonstrate the good performance of the proposed method in terms of segmentation accuracy and execution speed.

The remainder of this paper is organized as follows. Section 2 presents the close form solutions to detect the symmetric pattern in a CT cube. Section 3 describes the approach to dealing with the liver detection and segmentation using the cube-based representation. Section 4 describes the experimental tests to illustrate the effectiveness of the proposed method. Finally, brief conclusions are drawn in Section 5.

II. CUBE-BASED SYMMETRIC PATTERN DETECTION

![Fig.1. Modelling a 3D liver shape with a set of CT cubes: (a) four successive CT slices form a sub-volume; (b) the four enlarged blocks at the same location but in different slices from (a); (c) the intensity transition plane obtained by stacking the blocks in (b).](image)

Figure 1 shows that the liver shape in a CT sub-volume of four slices can be described by a set of CT cubes in which the symmetric patterns can be extracted using the proposed moment based feature detector. The cube-based approach to detecting liver shapes on CT volumes estimates the symmetric pattern of a cube shown in Fig. 2(a). Every cube is a 4x4x4 parallelepiped, where the first two dimensions are determined by the slice size and the third one is the time. Moreover, a CT cube is a part of a CT sub-volume which is constituted by four consecutive slices. Feature parameters of a cube describe the variation of the image content along the x-y plane and t-axis and are estimated by a 3D moment based symmetry detector.

The 3D moments of a cube C are computed by

\[
M_{pqr} = \iiint_{(x,y,t)\in C} \psi(x, y, t) x^p y^q t^r f(x, y, t) dx dy dt
\]

where \(\psi(x,y,t)\) and \(f(x,y,t)\) are the weighting and the intensity of voxel \((x,y,t)\) in \(C\), respectively. The weighting of a voxel is the ratio of the voxel area in the inscribed sphere to the voxel area in the cube.

The voxel values of a CT cube can be classified into two classes by a 3D plane \(E\) shown in Fig. 2(b). Based on the 3D plane model, the content of the cube is approximated by six parameters, namely, two representative voxel values \(h_1\) and \(h_2\), plane translation \(l\), and the 3D skeleton \(\mathbf{h} = (h_1, h_2, l)\) specified by two angles \(\alpha\) and \(\beta\), where \(\alpha\) is the angle between the projection of \(\mathbf{h}\) on the x-y plane and the x-axis; \(\beta\) is the angle between axis \(t\) and \(\mathbf{h}\). The transition is approximated as simply a step transition from representative value \(h_1\) to representative value \(h_2\). In practice, the translation \(l\) is defined as the length from the centre of the cube to \(E\), and is confined within the range of -1 to +1. Note that \(\mathbf{h}\) is on the symmetrical plane which is perpendicular to \(E\).

![Fig. 2. Creating 3D CT cubes. (a) A given CT volumes is partitioned into a set of non-overlapping cubes. (b) A 3D step transition model in a cube C, which is inscribed by a sphere.](image)

Let \(R\) be the 3D rotation matrix which aligns \(\mathbf{h}\) with the axis \(t\). Matrix \(R\) is defined as

\[
R = \begin{bmatrix}
\cos \alpha \cos \beta & -\sin \alpha & \cos \alpha \sin \beta \\
\sin \alpha \cos \beta & \cos \alpha & \sin \alpha \sin \beta \\
-\sin \beta & 0 & \cos \beta
\end{bmatrix}.
\]

(2)

The rotated coordinates \(x', y', t'\) of every voxel \((x,y,t)\) in a cube can then be computed by

\[
\begin{align*}
x' &= x \cos \alpha \cos \beta + y \sin \alpha \cos \beta - z \sin \beta \\
y' &= -x \sin \alpha + y \cos \alpha \\
t' &= x \cos \alpha \sin \beta + y \sin \alpha \sin \beta + z \cos \beta
\end{align*}
\]

(3)

If \(\mathbf{h}\) is on the symmetric plane of the input cube \(C\), we can use the rotation moments \(M'_{100}\) of \(C\) to compute the values of \(\alpha\) and \(\beta\) since both \(M_{100}'\) and \(M_{100}'\) are zeros. Then, we have

\[
\begin{bmatrix}
M_{100}' \cos \alpha \cos \beta + M_{001}' \sin \alpha \cos \beta - M_{001}' \sin \beta \\
-M_{100}' \sin \alpha + M_{001}' \cos \alpha \end{bmatrix} = \begin{bmatrix}
0 \\
0
\end{bmatrix}.
\]

(4)

Eq. (4) can be further transformed into

\[
\begin{bmatrix}
M_{100} n_1 + M_{001} n_1 - M_{001} \sqrt{n_1^2 + n_1^2} \\
-M_{100} n_1 + M_{001} n_1
\end{bmatrix} = \begin{bmatrix}
0 \\
0
\end{bmatrix}
\]

(5)

since \(\cos \beta = n_x, \cos \alpha = n_1\) and \(\sin \alpha = n_y\). Using (5), we get

\[
\begin{bmatrix}
M_{001} \\
M_{001}
\end{bmatrix} = \begin{bmatrix}
M_{100} \\
M_{100} + M_{001} + M_{001}
\end{bmatrix}
\]

(6)

where \(\gamma\) is the regulator to keep \(\sqrt{n_1^2 + n_1^2 + n_1^2} = 1\).

Once the values of \(\alpha\) and \(\beta\) are obtained, the solutions for \(h_1, h_2, p_1,\) and \(p_2\) can be found by employing the moment-preserving technique [13]. The remaining unknown parameter \(l\) is simple to calculate from the fact that \(p_2 = V_2/V\), which is the ratio of the area covered by the voxel value \(h_2\) to the total volume of the inscribed sphere of \(C\). The value of \(V_2\) can be computed by

\[
V_2 = \frac{1}{3} \pi (1-l)^2 (2+l)
\]

(7)

Then, we get
\[ l^3 - 3l^2 + 2 - 4p_2 = 0 \]  
(8)

The solutions of (8) are

\[ l = 2 \cos(\frac{\pi}{2} \theta), \]
(9)

\[ l = 2 \cos(\frac{\pi}{4} \theta + 120^\circ), \]
(10)

\[ l = 2 \cos(\frac{\pi}{4} \theta + 240^\circ), \]
(11)

where \( \cos \theta = 2p_2 - 1 \). The solution for \( l \) can be determined by imposing the constraint \(-1 < l < 1 \) on \( l \).

The representative point \( \pi(x, y, T) \) of the cube \( C \) is thus determined by

\[(\pi, y, T) = (x_c, y_c, t_c) + l(n_x, n_y, n_z)\]

(12)

\((x_c, y_c, t_c)\) are the coordinates of the centre of \( C \).

Based on the parameters of a video cube \( C \), the voxels of \( C \) can be quantized into two parts – one of them is represented as \( h_1 \) and the other is represented as \( h_2 \).

The plane to characterize the transition of voxel values in the cube can be described as

\[ x \times n_x + y \times n_y + t \times n_z = l, -1 \leq x, y, t, l \leq 1. \]

(13)

The distance from the original point of the image coordinates to the transition plane can thus be computed by

\[ l = x \times n_x + y \times n_y + t \times n_z. \]

(14)

The gradient value of the cube is thus computed by

\[ \nabla = \|h_1 - h_2\|. \]

(15)

When the gradient value of \( C \) is less than a pre-defined threshold, \( C \) is judged as a uniform cube; otherwise, \( C \) is an edge cube. In this work, only the edge cubes are included to describe the liver shape in a CT sub-volume.

During the process of 3D liver modeling, cooperating the LSI approach with the vector codebook, the reference table (R-table) of each cube \( C \) which is indexed by the LSI in terms of symmetric pattern. For the sake of illustration, the structure which is the fusion of cube-based LSI codebook and the R-table is called as the cube-based GHT (CGHT) model, shown in Fig. 4.

Let \( V \) be a CT volume of \( n \) slices. Given a training set \( T = \{V_j\}_{j=1}^n \), our goal is to construct the most stable liver template from the median of \( T \). Applying the key-frame detection presented in our previous work [15] to all training CT volumes in \( T \), we obtain sub-volume segmentation maps in which the partitioning variable \( \{\tau_{ij}^{(0)}, \lambda_{ij}^{(1)}\}_{j=1}^k \) records the temporal boundaries of sub-volumes in \( V \). In fact, CT slices of temporally indexes \( t_s \) satisfying \( \tau_j^{(0)} \leq t_s < \tau_{j+1}^{(0)} \) constitute the \( j \)-th sub-volume of \( V \). Without loss of generality, we set \( \tau_1 = 1 \) and \( \lambda_k = n_s + 1 \) where \( m \) is the number of slices in \( V \). The 4x4x4 cubes in each new sub-volume are then collected and used to train a model that represents the content of liver in the sub-volume. Performing the symmetric pattern detector on these cubes, sub-volume \( SV_j \) of \( V_j \) is represented by a set of cube features \( F \), which is factorized as \( F = UWV^T \) where \( U \) is an \( m \times m \) rectangular diagonal matrix with \( n \) non-negative eigenvalues on the diagonal, and \( V^T \) (the transpose of \( V \)) is an \( n \times n \) real unitary matrix. By preserving first \( k \) largest eigenvalues in \( \Sigma \), we construct the latent semantic indexing (LSI) for \( F \) by projecting every feature vector onto the corresponding eigenvector. These projected features are then clustered with the cluster representatives as the LSI codewords [12].

During the process of 3D liver modeling, cooperating the LSI codewords, the reference table (R-table) is employed to describe the average shape in the given training sub-volume \( SV_j \) using the 3D liver centroid \( \bar{C} \) as the reference point. The R-table is constructed by storing the offset vector \( \bar{C} - \bar{C}_j \) of each cube \( C \) which is indexed by the LSI in terms of symmetric pattern. For the sake of illustration, the structure which is the fusion of cube-based LSI codebook and the R-table is called as the cube-based GHT (CGHT) model, shown in Fig. 4.

Fig. 3. The workflow of the proposed cube-based automatic liver segmentation on CT volumes.

Fig. 4. The CGHT model which is the fusion of cube-based LSI codebook and R-table for modeling 3D liver shape in a sub-volume.
corresponding set of reference vectors to be retrieved from the R-table is \( \left\{ \mathbf{r}_{m,b} \right\}_{b=1}^{K_{m}} \). Next, the possible location of reference point in the parameter space is calculated from \( \mathbb{R} + A_{j}^{(i)} F_{m,b}^{T}, b = 1, \ldots, K_{m} \), where \( A_{j}^{(i)} \) is the affine transformation specified by five parameters, \( i.e., \) the 3D rotation angles \( \alpha \) and \( \beta \) and the scaling factors \( s_{c}, s_{r}, \) and \( s_{s} \). An accumulator array \( \mathcal{H} \) for parameter space saves the votes of edge cubes in \( V \) to determine the most probable centre of the target liver. Finally, to perform the inverse Hough transform on the peak of \( \mathcal{H} \), the liver contours which locate the liver shape in \( V \) are determined. Obviously, the time complexity of the 3D Hough voting is extremely high since the space of eight parameters is very large. To tackle the difficulty, feasible ranges of these parameters are learned and stored into the database in the training phase in order to reduce both the search space and time complexity of Hough voting using 3D GHT models.

Once the initial 3D liver shape is detected using Hough voting, to further improve the robustness of the liver detection method, the affine transformation parameters \( A_{j}^{(i)} \) are refined by the iterative closest point (ICP) algorithm [17] using the cube correspondences between the training liver defined by \( \phi_{j}^{(i)} \) and the detected shape in \( V \). In fact, for each cube \( C \) in \( \phi_{j}^{(i)} \), the affine transformation parameters \( A_{j}^{(i)} \) determines the corresponding cube \( C' \) in \( V \); thus, the error of liver detection is computed by

\[
\varepsilon(\phi_{j}^{(i)}, V) = \sum_{(C,C')} \| F(C) - F(C') \| \tag{17}
\]

where \( \Gamma_{j} \) is the set of cube correspondences determined by ICP.

\[
\mathcal{F}(V, V^*) = \arg \min_{V} \sum_{r \in CT} \phi(V, V^*).
\tag{19}
\]

Fig. 5 shows the construction of the template CGHT models using the star-like pair-wise sequence alignment. Once the CT volume template \( V^* \) is obtained, in the next step, for each sub-volume inside \( V^* \), we compute the variations of affine parameters which are stored into the parameter database in order to speed up the parameter searching in the further liver detection using the Hough voting.

Let \( V^* \) be represented by the set of CGHT models \( \phi_{j}^{*}, j = 1, \ldots, k \), where \( k \) is the number of sub-volumes of \( V^* \). As shown in Fig. 5, for each training CT volume \( V \), their sub-volumes are aligned with \( \phi_{j}^{*} \) using DP and this defines training sub-volume clusters \( \Omega_{j}^{*}, j = 1, \ldots, k \). In the computation of the variations of affine parameters, we characterize every CGHT model \( \phi_{j}^{*} \) by the mean affine \( \bar{A}_{j}^{*} \) and standard variation \( \sigma_{j}^{*} \) of the distances of affine parameters between sub-volumes in \( \Omega_{j}^{*} \). The former is computed by

\[
\bar{A}_{j}^{*} = \frac{1}{| \Omega_{j}^{*} |} \sum_{SV \in \Omega_{j}^{*}} A(\phi_{j}^{*}, SV), j = 1, \ldots, k
\tag{20}
\]

where \( A(\phi_{j}^{*}, SV) \) is obtained by performing the Hough voting to detect the liver shape in \( SV \) with the CGHT model \( \phi_{j}^{*} \) and the ICP algorithm to compute the affine parameters to transform the liver shape of \( \phi_{j}^{*} \) into the liver shape of \( V \). The affine parameters variation can thus be computed by

\[
\sigma_{j}^{*} = \frac{1}{| \Omega_{j}^{*} |} \sum_{SV \in \Omega_{j}^{*}} \| A(\phi_{j}^{*}, SV) - \bar{A}_{j}^{*} \|^{2}, j = 1, \ldots, k
\tag{21}
\]

Finally, the cascade of each CGHT model \( \phi_{j}^{*} \) in \( V^* \) thus constructs the final CGHT models \( M = \{ \phi_{j}^{*} \}_{j=1}^{k} \) for further 3D liver detection and segmentation. For the sake of illustration, we denote the 3D liver shape defined by \( M \) as \( \Phi^{*} \).

B. The Detection and Segmentation Algorithm

Given a test CT volume \( V \), the slices of \( V \) are first segmented into multiple sub-volumes using the same key-frame detection used in the training phase. Again, each sub-volume is partitioned into a set of 4x4x4 cubes to detect their symmetric patterns. Next, to integrate the Hough voting into the dynamic programming, the sub-volumes of \( V \) are optimally aligned with the learned CGHT model sequence \( M = \{ \phi_{j}^{*} \}_{j=1}^{k} \). If the \( i \)-th sub-volume of \( V \) is in \( \Omega_{j}^{*} \), \( i.e., \) the set of sub-volumes of \( \phi_{j}^{*} \), the Hough voting framework locates the 3D liver shape \( \Phi_{j}^{*} \). The cascade of these detected 3D liver shapes then constructs the initial 3D liver shape of \( V \):

\[
\Phi_{V} = \bigcup_{j=1, \ldots, k} (\Phi_{j,1} \otimes \ldots \otimes \Phi_{j,|\Omega_{j}^{*}|})
\tag{22}
\]

where \( \otimes \) is the concatenation operator which reconstructs the 3D liver shape in \( V \) that is similar to the template 3D shape defined in \( M \). Inspired form the 3D level-set framework [18], 3D volumetric CT
data surrounding the detected liver shape $\Phi$ define a region-of-interest as an open subset $V_1$ of $V$. The evolution surface $S \in \mathbb{R}^4$ can be defined as boundary of $V_1$, i.e. $S = \partial V_1$. Thus, the surface $S$ divides $V$ into two areas, i.e. $V_1$ (the area inside $S$) and $V_2$ (the area outside $S$), which correspond to the liver body and the background, respectively. Then the level-set framework compute the surface $S$ by embedding it as zero level sets of a time-varying and high dimensional function $\Psi_i(x)$:

$$S_i = \{ x \in \mathbb{R}^4 \mid \Psi_i(x) = 0 \}$$  \hspace{1cm} (23)

where $x$ is a voxel of $V$. The function $\Psi_i(x)$ can be defined by $\Psi_i(x) = d(x)$ where $d$ is a signed distance from $x$ to the surface $S$. To compute the segmentation model, i.e. the surface $S$, the energy function to be minimized can be defined as \[18\]:

$$E(S) = \lambda_i \int_{V_1} -g_L(I(x))dx + \lambda_B \int_{V_2} -g_B(I(x))dx + \nu \int S \partial_\delta S \partial_\delta S \hspace{1cm} (24)$$

where $I(x)$ is the intensity of voxel $x$; $g_L(g_B)$ is the Gaussian distribution to measure the probability belonging to the liver body (background) in terms of voxel intensity; $\nu$, $\lambda_L$ and $\lambda_B$ are adjustable parameters.

The mean $\mu_i$ and the variance $\sigma_i$ define the Gaussian distribution $g_i$ of the liver body:

$$g_i(I(x)) = \frac{1}{\sqrt{2\pi \sigma_i(x)}} \exp\left(\frac{(I(x) - \mu_i(x))^2}{\sigma_i(x)^2}\right).$$  \hspace{1cm} (25)

Similarly, we can use $\mu_B$ and $\sigma_B$ of the Gaussian distribution $g_B$ for the background. For the sake of the limitation on paper length, the details to calculate these Gaussian parameters using the neighbors statistical analysis can refer to \[18\]. Moreover, the gradient-descent approach can be used to minimize function $E(S)$ in Eq. (24) using calculus of variations \[18, 19\].

**IV. EXPERIMENTAL RESULTS**

A series of experiments was conducted on an Intel CORE i7 3G Hz PC and tested on the training and testing datasets of MICCAI 2007 liver-segmentation challenge. There are 20 CT volumes of abdomen with contrast agent in the training datasets, and ten in the testing datasets. All datasets have an in-plane resolution of $512 \times 512$ pixels and inter-slice spacing from 0.5 to 5.0 mm. In the training phase, other 20 CT volumes with normal liver anatomy obtained clinically are used to generate the template and the sequence of CGHT models.

![Fig. 6. An example of the cub-cubed CT volume modelling: (a) the 3D liver shape in a test CT volume; (b) the point cloud of (a) by representing each edge cube as its representative point.](image)

The first experiment illustrates the effectiveness of the proposed cube-based CT volume modelling. Given a CT volume $V$, the volume is divided into a set of 4x4x4 CT cubes in which the symmetric patterns are detected using the proposed 3D moment based feature detector. A reverse process to render the slice of $V$ is also implemented using the extracted cube parameters. The accuracy of the cube-based representation is measured by the distance between the centre of a given 3D liver shape and the centre of the point cloud which is the set of cube representative points defined in (12). For all training CT volumes, the centre drifts are less than 2 voxels. Fig. 5 shows an example of the cube-based representation of 3D liver shape.

The second experiment is on the accuracy evaluation of liver detection and segmentation. The liver shapes in a test CT volume are defined as the targets to detect using the fusion of CGHT models of the template and dynamic programming. The human-annotated liver shapes in the CT test volume play the role as the ground truth to measure the detection and segmentation accuracy. To improve the robustness of the proposed method, the median volume of the training CT volumes is selected to construct the CGHT models. As shown in Fig. 7, the liver shape in a key-frame of the template actually plays a role as the representative shape of a cluster in which similar shapes from other training CT slices are grouped using the learning approach. Fig. 8 shows an example of liver detection and segmentation using the models of the template. Segmentation results are compared to the reference results according to the following five metrics: volumetric overlap error (OE), signed relative volume difference (SVD), symmetric average surface distance ($D_{AVG}$), symmetric root-mean-squared surface distance ($D_{RMS}$), and maximum surface distance ($D_{MAX}$). Table I shows the performance comparison with the state-of-the-art methods. Accordingly, the proposed method outperforms the compared methods.

**TABLE I. PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART METHODS.**

<table>
<thead>
<tr>
<th>Method</th>
<th>OE [%]</th>
<th>SVD [%]</th>
<th>$D_{AVG}$ [mm]</th>
<th>$D_{RMS}$ [mm]</th>
<th>$D_{MAX}$ [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kainmüller [1]</td>
<td>6.1</td>
<td>-2.9</td>
<td>0.9</td>
<td>1.9</td>
<td>18.7</td>
</tr>
<tr>
<td>Heimann [1]</td>
<td>7.7</td>
<td>1.7</td>
<td>1.4</td>
<td>3.2</td>
<td>30.1</td>
</tr>
<tr>
<td>Sadri [1]</td>
<td>8.9</td>
<td>2.6</td>
<td>1.5</td>
<td>3.4</td>
<td>29.3</td>
</tr>
<tr>
<td>Schmidt [1]</td>
<td>10.4</td>
<td>-4.9</td>
<td>1.7</td>
<td>3.1</td>
<td>24.0</td>
</tr>
<tr>
<td>Chi [1]</td>
<td>9.1</td>
<td>2.6</td>
<td>1.7</td>
<td>3.3</td>
<td>30.8</td>
</tr>
<tr>
<td>Rusko [1]</td>
<td>10.1</td>
<td>-3.8</td>
<td>1.7</td>
<td>3.5</td>
<td>26.7</td>
</tr>
<tr>
<td>Seghers [1]</td>
<td>10.7</td>
<td>-6.8</td>
<td>1.8</td>
<td>3.2</td>
<td>25.2</td>
</tr>
<tr>
<td>Furukawa [1]</td>
<td>10.8</td>
<td>-7.3</td>
<td>1.9</td>
<td>3.7</td>
<td>31.6</td>
</tr>
<tr>
<td>VanRikxoort [1]</td>
<td>12.5</td>
<td>1.8</td>
<td>2.4</td>
<td>4.4</td>
<td>32.4</td>
</tr>
<tr>
<td>Susomboon [1]</td>
<td>26.4</td>
<td>-11.5</td>
<td>10.2</td>
<td>17.1</td>
<td>74.0</td>
</tr>
<tr>
<td>Zhang [9]</td>
<td><strong>5.25</strong></td>
<td>0.73</td>
<td>0.93</td>
<td>2.23</td>
<td>24.8</td>
</tr>
<tr>
<td>Proposed</td>
<td>6.17</td>
<td><strong>0.71</strong></td>
<td><strong>0.58</strong></td>
<td><strong>1.01</strong></td>
<td><strong>10.34</strong></td>
</tr>
</tbody>
</table>
The efficiency of the proposed method is verified by comparing the time complexity with Zhang et al.’s method [9] which also performs 3D GHT to locate liver shapes from the input CT volume. The average time to locate the liver shapes for proposed method and Zhang et al.’s method are 20.4 and 61.2 seconds, respectively. The proposed method has 2.375 speedup as compared with Zhang et al.’s method.

V. CONCLUSION

In this paper we have presented a cube-based representation for detecting and segmenting 3D liver shapes in CT volumes. Combining the dynamic programming with the CGHT modelling, the effectiveness of the proposed liver detection and segmentation is verified by an application to computer-aided diagnosis using CT volumes. The system enhances the quality of liver segmentation to achieve the goal of constructing a high-performance CAD applications. Our approach inherited the robustness of the Hough voting framework in object detection without the main disadvantage of high-computational complexity for traditional 3D GHT. The speed of detection can also be increased without affecting the robustness of the system by incorporating fast k-NNR searches into the cube searching framework. Future work will deal with adding complex medical usages to the proposed system, and increasing the database size.

ACKNOWLEDGMENT

This work was supported in part by Ministry of Sciences and Technology, Taiwan under grant number MOST 103-2221-E-019-018-MY2.

REFERENCES


