



Published in final edited form as:

*Comput Med Imaging Graph.* 2015 July ; 43: 1–14. doi:10.1016/j.compmedimag.2015.01.006.

## Iterative Mesh Transformation for 3D Segmentation of Livers with Cancers in CT Images

Difei Lu, PHD<sup>1,2</sup>, Yin Wu, PHD<sup>1</sup>, Gordon Harris, PHD<sup>1</sup>, and Wenli Cai, PHD<sup>1,\*</sup>

<sup>1</sup>Department of Radiology, Massachusetts General Hospital and Harvard Medical School

<sup>2</sup>Department of Informatics, Zhejiang Police College

### Abstract

Segmentation of diseased liver remains a challenging task in clinical applications due to the high inter-patient variability in liver shapes, sizes and pathologies caused by cancers or other liver diseases. In this paper, we present a multi-resolution mesh segmentation algorithm for 3D segmentation of livers, called *iterative mesh transformation* that deforms the mesh of a region-of-interest (ROI) in a progressive manner by iterations between mesh transformation and contour optimization. Mesh transformation deforms the 3D mesh based on the deformation transfer model that searches the optimal mesh based on the affine transformation subjected to a set of constraints of targeting vertices. Besides, contour optimization searches the optimal transversal contours of the ROI by applying the dynamic-programming algorithm to the intersection polylines of the 3D mesh on 2D transversal image planes. The initial constraint set for mesh transformation can be defined by a very small number of targeting vertices, namely *landmarks*, and progressively updated by adding the targeting vertices selected from the optimal transversal contours calculated in contour optimization. This iterative 3D mesh transformation constrained by 2D optimal transversal contours provides an efficient solution to a progressive approximation of the mesh of the targeting ROI. Based on this iterative mesh transformation algorithm, we developed a semi-automated scheme for segmentation of diseased livers with cancers using as little as five user-identified landmarks. The evaluation study demonstrates that this semiautomated liver segmentation scheme can achieve accurate and reliable segmentation results with significant reduction of interaction time and efforts when dealing with diseased liver cases.

### Keywords

image segmentation; mesh deformation; dynamic-programming; liver segmentation

---

© 2015 Elsevier Ltd. All rights reserved.

\*Corresponding author: Wenli Cai, PhD, Department of Radiology, Massachusetts General Hospital, Address: 25 New Chardon St., 400C, Boston, MA02114, USA, Cai.Wenli@mgh.harvard.edu, Tel. (617)726-0515, Fax: (617) 724-6130.

**Publisher's Disclaimer:** This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

## I. Introduction

Liver cancer is the third most common cause of death from cancer worldwide [1], and liver is the most common metastatic spread sites of cancers after the lymph nodes, which have been found in 30-70% of patients who are dying of cancer [2]. With the technical advances of computed tomography (CT), hepatic CT scanning has become one of the major routinely-used clinical imaging modality in cancer diagnosis, staging, and treatment evaluation because of the relatively low cost and wide availability [3]. Segmentation of liver in CT images is crucial for liver-related clinical applications. However, accurate and reliable liver segmentation continues to be a challenging task especially for diseased livers because of the deformation of the liver shape caused by large tumors, the complexity of the diseased liver pathologies, the variability of image presence of different hepatic diseases (such as hypodense or hyperdense lesions), and the fuzzy boundaries between liver and the adjacent organs (such as heart and stomach). Therefore, liver segmentation attracts continuously researchers' attentions in medical imaging and graphics community [4, 5].

Numerous automated, semi-automated and interactive methods for liver segmentation have been developed, including statistical shape model [6-8], atlas matching method [9-11], deformable model [12, 13], level-set method [14-17], and other hybrid methods based on user-interaction [18], graph-cut [19], region-growing [20], machine-learning [21], etc. With regard to segmentation accuracy and reliability, there is a clear tendency that interactive methods outperform semi-automated methods, which in turn outperform fully automated approaches. However, user interaction time and efforts tend to have a reversed tendency: interactive segmentation of a liver on CT images is labor-intensive, time-consuming and prone to inter-operator variability. Clinical applications seek an accurate and reliable segmentation method for diseased livers with minimum amount of required user interactions.

The technique challenges of the segmentation of diseased livers result from the high inter-patient variability in liver shapes, sizes and pathologies as well as the longitudinal intra-patient variability. It is not uncommon that cancers and metastases change the liver size and deforms the liver shape, and thus result in a very large range of variations among patients. Figure 1 demonstrates the inter-patient variability in shapes and sizes of diseased livers from patients with cancers or metastases. The sizes of livers vary significantly with more than four times of difference. More importantly, the shapes of livers show substantial difference caused by cancers or metastases or treatments: 1) liver shapes may be bended or twisted or deformed severely due to the growing or shrinking of tumors; 2) different hepatic lobes may be shrunk or expanded extraordinarily; and 3) tumor tissues show substantially different CT intensity compared to normal liver parenchyma. This inter-patient variability makes the modeling of a statistic shape for liver segmentation an extreme challenging task. To deal with the complex shape variations of diseased livers, sparse shape composition model [22] and auto context multi-atlas model [23] were proposed to handle errors or outliers of statistic models and preserve local details for different shapes of liver. These atlas-based or shape-prior-based automatic methods require learning from available shapes of segmented livers, which may be itself a difficult task.

As long as fully-automated segmentation methods fail to produce satisfactory segmentations on these diseased cases, certain amounts of user interaction are still required. Deformable models offer the essential options of interactive control over the segmentation process via interactive manipulating of a deformable surface [24]. Traditional mesh deformation methods simulate the mechanism behavior of a surface by minimizing the stretching and bending energies of the surface, such as active geometric deformed models [25] and active contour models or snakes [26]. However, one limitation of these energy-based mesh deformable methods is that the difference between the initial mesh and resulting mesh must be sufficiently small, which may turn out a large amount of interactive manipulations. Motivated by the deformation transfer model [27], which transfers the mesh deformation using deformation gradients instead of local energies, mesh may be deformed efficiently using a small set of targeting points while allowing accurate detail-preserving interactions.

In this paper, we proposed a multi-resolution mesh segmentation algorithm, called *iterative mesh transformation*, that deforms the mesh of a region-of-interest (ROI) in a progressive manner by iterations between mesh transformation and contour optimization. The initial constraint set for mesh transformation can be defined by a very small number of targeting vertices, namely *landmarks*, and progressively updated by adding the targeting vertices selected from the optimal transversal contours calculated in contour optimization. This iteration between 3D mesh transformation and 2D transversal contours optimization provides an efficient solution to a progressive approximation of the targeting ROI mesh with minimum user interactions. By using this iterative mesh transformation algorithm, we developed a semi-automated image segmentation scheme for diseased livers with cancers. This scheme can effectively segment the diseased livers with cancers using as little as five user-identified landmarks. It achieves accurate and reliable segmentation results when dealing with these diseased liver cases, and reduces significantly the interaction time and efforts. The resulting 3D liver mesh provides the meta-data for quantitative image analysis of diseased livers in different clinical applications.

The major contributions of the proposed method are twofold.

- Technical contributions: Diseased livers tend to have high inter-patient variability of shapes and sizes, which makes the modeling of a statistic shape for liver segmentation an extreme challenging task. The transformation optimization in iterative mesh transformation provides an effective solution for the global approximation of a diseased liver, which is superior to those of energy-based mesh deformation that tends to converge to a local boundary caused by the inhomogeneity in diseased regions. On the other hand, the transversal contours optimization provides the local refinement for detecting the precise boundaries of a liver. As a result, the combination of 3D transformation optimization and 2D contour optimization provides an accurate and efficient technical solution for the segmentation of diseased livers.
- Clinical contributions: Manual contouring remains the major clinical procedure for segmentation of diseased livers with pathologies, which is labor-intensive, time-consuming and prone to inter-operator variability. The proposed iterative liver segmentation scheme is easy-of-use and reduces the interaction effort as little as

five user-identified landmarks. It takes in average 5 minutes to segment a diseased liver case, which is more efficient than most of existing liver segmentation methods. The accuracy and running time have been evaluated by the MICCAI cases as well as the real clinical cases collected in our institute.

## II. Method

### A. Mesh Segmentation Model

Let  $I(\mathbf{v})$  denote the pixel value at a vertex  $\mathbf{v} = [x, y, z]^T \in R^3$  in a CT volume, where  $R^3$  is 3-dimensional (3D) Euclidean space. A mesh can be defined as a 3-tuple  $\mathbf{M} = (\mathbf{V}, \mathbf{E}, \mathbf{Z})$ , where  $\mathbf{V}$  is a vector of vertices,  $\mathbf{E}$  is a set of edges connecting two vertices,  $\mathbf{Z}$  is a set of faces defined by a closed set of edges, in which a triangle face has three edges. A *mesh segmentation* is defined as an optimization process that deforms an initial mesh (such as a polyhedron) under certain constrains for segmentation of a ROI, which can be represented as an evolution process  $\mathbf{A}(\theta) = (M_0, M_\theta, \Omega)$ , where  $M_0$  is the initial mesh,  $M_\theta$  is the resulting ROI mesh,  $\Omega$  is the constrain set, and  $\theta$  is the steps of evolution.

Figure 2 illustrates the mesh evolution in the segmentation of a sphere, in which the initial mesh  $M_0$  is defined as an icosahedron. During each evolution, a triangle face is decomposed into four smaller triangle faces by adding a middle point on each edge of the triangle (see point  $D$  on edge  $AC$  in Figure 2(a)). The corresponding targeting vertex of the newly-added middle point is the boundary point on the sphere surface (see point  $D_I$  in Figure 2(b)). Mesh segmentation is a very effective way to approximate the surface or boundary of an object, because the number of faces increases in an exponential manner during the evolution. Table 1 lists the properties of the meshes at each step of evolution. The vertices set of  $M_\theta$  contains all the vertices of  $M_{\theta-1}$ .

Without the loss of generality, we may assume that the targeting ROI is a closed region and thus can be represented by a 2-dimensional (2D) manifold, *aka* a boundary surface. This assumption indicates that the segmented ROI is locally homeomorphic to a 2D Euclidean space (such as a plane), and thus its boundary surface can be modeled by a triangle mesh. Considering the fact that a sphere is a 2D manifold and is homeomorphic to the boundary surface of a targeting ROI, the homeomorphic theory states that there is a continuous one-to-one mapping between these two surfaces, as indicated below:

If  $\mathbf{X}$  and  $\mathbf{Y}$  are homeomorphic, there exists a mapping  $\psi: \mathbf{X} \rightarrow \mathbf{Y}$  that satisfies the following criteria: (1)  $\psi$  is a one-to-one correspondence, (2)  $\psi$  is continuous, and (3)  $\psi^{-1}$  is continuous.

If such a mapping  $\psi$  exists, we can transform a sphere mesh  $M^{Sphere}$  to the ROI mesh  $M^{ROI}$  by using  $\psi$  defined by a set of transformations between  $M^{Sphere}$  and  $M^{ROI}$ . Therefore, mesh segmentation is defined as a multi-resolution mesh optimization process by using an objective function subject to a set of correspondence constraints ( $\Omega$ ):

$$M_R^*(\theta) = \underset{s.t. \Omega}{\operatorname{argmin}}_{M^{ROI}} (w_0 f(M_0) + w_1 f(M_1) + \dots + w_\theta f(M_\theta)), \quad (1)$$

where  $M_R^*(\theta)$  is the approximation of the ROI mesh  $M^{ROI}$  after the evolution of  $\theta$  times,  $f$  is the objective function of the transformation optimization defined on mesh  $M_i$ ,  $w_i$  is the weight of  $f(M_i)$ . Constraint  $\Omega$  is a set of known correspondence between the sphere mesh  $M^{Sphere}$  and the ROI mesh  $M^{ROI}$ .

The weights in Equation (1) control the “level-of-detail” (or granularity) of deformation at different resolutions of the resulting mesh. In general,  $w_0$  controls the coarsest or overall deformation and  $w_\theta$  does the finest deformation for detail preservation. Rigid organs tend to have larger value of  $w_0$  and smaller value of  $w_\theta$  whereas flexible ones employ smaller value of  $w_0$  and larger value of  $w_\theta$ . In our study,  $\theta$  is set to 5, which indicates the number of vertices of the resulting mesh of a liver is 10242 ( $\theta = 5$ ).  $w_0 \sim w_5$ , are set to 0.1, 0.1, 0.1, 0.2 and 0.5, respectively. These values were estimated empirically based on the flexibility of a liver.

## B. Mesh Transformation

The homeomorphic assumption indicates that a local boundary of a ROI can be mapped to a local surface of a sphere if the local boundary is small enough. By dividing each transversal slice of a ROI into  $n$  sectors, as illustrated in Figure 3, we may establish the mapping correspondence  $\psi$  between the sphere surface and a ROI boundary when the sector is divided small enough. It results in a continuous one-to-one mapping between the ROI boundary point ( $v_i^{ROI}$ ) and the sphere mesh vertex ( $v_i^S$ ).

Let  $V^S$  and  $V^{ROI}$  be the vertex set of a sphere surface and a ROI boundary respectively, and the mapping function  $\psi$  be a linear transformation. We may represent  $v_i^{ROI}$  as the transformed vertex of  $v_i^S$ :

$$T \times v_i^S + D = v_i^{ROI}, \quad (2)$$

where  $v_i^S$  is the vertex of the initial sphere mesh,  $v_i^{ROI}$  is the vertex of the ROI boundary,  $T$  is a  $3 \times 3$  affine transformation matrix, and  $D$  is a translation vector.

Considering the geometric continuity during a mesh deformation, we assume that the four vertices  $v_1 \sim v_4$  around an edge  $E_i$  in a mesh share the same affine transformation  $T$  and translation  $D$ , as shown in Figure 4(a). Thus,  $v_i^{ROI}$  can be represented below:

$$\begin{aligned} T \times v_1^S + D = v_1^{ROI}, T \times v_2^S + D = v_2^{ROI} \\ T \times v_3^S + D = v_3^{ROI}, T \times v_4^S + D = v_4^{ROI} \end{aligned} \quad (3)$$

By eliminating  $D$  in Equation (3), the affine transformation  $T$  for the edge  $E_i$  between  $V^S$  and  $V^{ROI}$  can be determined by Equation (4):

$$T(E_i) = T^{ROI} \times [T^S]^{-1}, \quad (4)$$

where both  $T^{ROI}$  and  $T^S$  are  $3 \times 3$  matrix,

$$T^{ROI} = [v_2^{ROI} - v_1^{ROI}, v_3^{ROI} - v_1^{ROI}, v_4^{ROI} - v_1^{ROI}], \text{ and}$$

$$T^S = [v_2^S - v_1^S, v_3^S - v_1^S, v_4^S - v_1^S].$$

Because all initial vertices are defined on a sphere, any four vertices  $v_1^S \sim v_4^S$  around an edge  $E_i$  are non-coplanar and thus  $[T^S]^{-1}$  always exists. Therefore, the transformation matrix  $T(E_i)$  for an edge  $E_i$  represented by Equation (4) is a linear function of the unknown coordinates of four deformed vertices  $v_i^{ROI}$ ,  $i=1 \sim 4$ , as illustrated in Figure 4(a). This indicates that the mapping  $\psi$  between  $V^S$  and  $V^{ROI}$  is a function of unknown vertices  $v_i^{ROI}$  and can be represented by a linear combination of  $T(E_i)$ , where  $E_i$  is an edge in the edge set of all the triangles that share the vertex of  $v_i^S: E_i \in \{E(Z_i), \text{ where } v_i^S \in V(Z_i)\}$ .

### C. Mesh Objective Function

The objective function  $f(M_i)$  in Equation (1) is formed by the weighted sum of three types of constraints: smooth constraint ( $C_S$ ) deforming constraint ( $C_D$ ) and distance constraint ( $C_L$ ):

$$f = w_S C_S + w_D C_D + w_L C_L, \quad (5)$$

where  $w_S$ ,  $w_D$ ,  $w_L$  are the weights of the corresponding constraints.

The smooth constraint  $C_S$  is so defined that the transformation matrix  $T(E_i)$  (see Equation (4)) for three edges of a triangle face should be as equal as possible. In specific, for each triangle face  $Z_i$  in a mesh  $M$ , as shown in Figure 4(b and c),  $C_S$  is defined below:

$$C_S(M) = \sum_{Z_i \in Z(M)} \sum_{\substack{j \neq k=1 \\ E_j \in E(Z_i) \\ E_k \in E(Z_i)}}^3 \|(T(E_j) - T(E_k))\|_F^2, \quad (6)$$

where  $Z(M)$  indicates the triangle face set of mesh  $M$ ,  $Z_i$  is the  $i^{th}$  face,  $T(E_j)$  and  $T(E_k)$  are the transformation matrix of edge  $E_j$  and  $E_k$  in the triangle  $Z_i$  respectively,  $\|\cdot\|_F^2$  is the Frobenius norm of a matrix.

The deforming constraint  $C_D$  transforms the vertices ( $v_i^{ROI}$ ) of the mesh as close as possible to the targeting vertices ( $v_i^{TC}$ ) of a ROI:

$$C_D(M) = \sum_{(v_i^{ROI}, v_i^{TC})} \|(v_i^{ROI} - v_i^{TC})\|^2, \quad (7)$$

where  $\Omega$  is the set of known vertex correspondence between the sphere mesh and the object boundary of a ROI.

Targeting vertices are known boundary points of a ROI, which may be established by the user-identified landmarks of the ROI  $\mathbf{V}^{landmark}$ , the segmented neighboring organs such as  $\mathbf{V}^{chest}$  (for liver segmentation), and the selected contour points  $\mathbf{V}^{contour}$  on optimal transversal contours. For our application of liver segmentation, we can define

$$\Omega = \{(\mathbf{v}_i^{ROI}, \mathbf{v}_i^{TC}) | \mathbf{v}_i^{ROI} \in \mathbf{V}(\mathbf{M}) \text{ and } \mathbf{v}_i^{TC} \in (\mathbf{V}^{landmark} \cup \mathbf{V}^{chest} \cup \mathbf{V}^{contour})\}.$$

In order to maintain the shape regularity of a triangle and prevent the mesh from drastic deformation, we add the distance constraint  $\mathbf{C}_L$  for maintaining the minimum signed distance change before and after deformation:

$$\mathbf{C}_L(\mathbf{M}) = \sum_{\substack{E_i \in \mathbf{E}(\mathbf{M}), \\ \mathbf{v}_1 \in \mathbf{V}(E_i), \\ \mathbf{v}_2 \in \mathbf{V}(E_i)}} \|(\mathbf{v}_1^{ROI} - \mathbf{v}_2^{ROI}) - (\mathbf{v}_1^s - \mathbf{v}_2^s)\|^2, \quad (8)$$

where  $\mathbf{E}(\mathbf{M})$  indicates the edge set of mesh  $\mathbf{M}$ ,  $\mathbf{V}(E_i)$  is a set of end points of edge  $E_i$ , i.e.,  $\mathbf{v}_1$  and  $\mathbf{v}_2$  are two end points of an edge  $E_i$ .

The weights in Equation (5) balance three deformations applied to each triangle:  $w_S$  and  $w_L$  are smoothing terms in which  $w_S$  controls the continuity of transformations among adjacent triangles and  $w_L$  controls the shape regularity of a triangle;  $w_D$  is a positive term which controls the transformation between mesh vertices and the targeting vertices, i.e. the detected boundary points. In our study, we set  $w_S = 1$ ,  $w_D = 100$  and  $w_L = 1$ . The large value of  $w_D$  implies the strong preservations of the known boundary vertices.

#### D. Mesh Optimization

Based on Equation (1), mesh segmentation is an optimal process for minimizing the objective function by searching a new set of unknown vertices  $\mathbf{V}^{ROI}$ . Because all the vertices in the resulting deformed mesh  $\mathbf{M}_\theta = (\mathbf{V}, \mathbf{E}, \mathbf{Z})$  are independent, Equation (1) can be represented by a vector-valued system of  $\mathbf{V}^{ROI} = [\mathbf{v}_1^{ROI}, \mathbf{v}_2^{ROI}, \dots, \mathbf{v}_n^{ROI}]$  which includes all vertices of meshes  $M_0 \sim M_\theta$ . In addition, because  $\mathbf{T}(E_i)$  is a function of  $\mathbf{v}_i^{ROI}$  (see Equation (4)), all constrains in the objective function  $\mathbf{f}$  form a second order polynomial of  $\mathbf{v}_i^{ROI}$ , i.e.,  $\mathbf{f}$  can be represented by a linear system:

$$\mathbf{f} = \sum_{\xi=1}^{N^*} \|\mathbf{A}_\xi \mathbf{V} + \mathbf{B}_\xi\|_F^2, \quad (9)$$

where  $N^*(=S, D, L)$  is the total number of items in Equation (6), (7) and (8),  $\mathbf{A}_\xi$  is a  $N^* \times n$  sparse coefficient matrix and  $\mathbf{B}_\xi$  is a  $N$  column vector.

Therefore, the object function  $\mathbf{f}$  in Equation (5) can be reformed to a system of linear equations of variable vector  $\mathbf{V}^{ROI}$ . Because function  $\mathbf{f}$  has a representation of a sparse

matrix, we may use the sparse LU factorization [28] to efficiently solve the optimized  $\mathbf{V}^{ROI}$  in the linear system, which results in the transformed ROI mesh. The detailed process of solving the minimization problem in Equation (1) can be found in Appendix 1.

## E. Contour Optimization

Initial mesh segmentation based on the constraint set  $\Omega$  without  $\mathbf{V}^{contour}$  in Equation (7) may result in the mismatches between the intersection polylines of the mesh and the true ROI boundaries on 2D transversal image planes in CT volume due to the lack of local boundary constraints ( $\mathbf{V}^{contour}$ ) in  $\Omega$ . Thus, the constraint set  $\Omega$  needs to be updated before the next evolution of mesh deformation to achieve accurate segmentation. This constraints updating is performed automatically by adding targeting vertices ( $v_i^{TC}$ ) selected from the local boundaries of a ROI into  $\mathbf{V}^{contour}$ , which are searched in the neighborhood of the initial intersection polylines.

Assuming that the boundary of a ROI is located in the neighborhood of the intersection polylines of the initial mesh on the 2D transversal images. This neighborhood is defined as a band on the transversal images centered by the intersection polylines of the initial mesh, as shown in Figure 5. The width of the band can be adjusted in terms of the size of the ROI. In our study, we set the band width to be 40 pixels. The local boundary of a ROI will be detected by optimal-path searching within this local band.

A transversal CT image can be represented by a 2D weighted bi-directed graph, in which one node corresponds to a pixel in the image. Each node (pixel) has 8 edges connecting to its neighborhood pixels, and each node and each edge have a related cost, respectively. A local ROI boundary on a transversal image plane is defined as the optimal path with the minimum cost between two corresponding nodes in the graph. We defined five cost functions on a node or an edge employing the forms similar to those described in [29].

**(1) The Laplacian zero-crossing cost**—The Laplacian zero-crossing is defined as the points with local maximal or minimal gradient magnitude, which corresponds to a boundary point. Thus, a node has a low cost if it is a zero-crossing point:

$$S_{zx}(v) = \begin{cases} 0; & \text{if } \nabla^2(v)\nabla^2(u) < 0 \text{ and } |\nabla^2(v)| < |\nabla^2(u)| \\ 1; & \text{Otherwise} \end{cases} \quad (10)$$

where  $\nabla^2()$  is the Laplacian operator  $u$  is a neighboring pixel of  $v$ .

**(2) Gradient magnitude cost**—A large gradient magnitude tends to indicate a local boundary. Thus, a gradient magnitude is considered as the local cost of a boundary:

$$S_g(v, u) = k \left( 1 - \frac{|\nabla(u)|}{\max(|\nabla(u_i)|)} \right), \quad (11)$$

where  $\nabla()$  is the gradient operator,  $k$  is 1 if  $v$  and  $u$  are diagonal neighbors, and  $k$  is  $\sqrt{2}$  if  $v$  and  $u$  are horizontal or vertical neighbors.

**(3) Gradient direction cost**—We assume that a local boundary is perpendicular to the gradient direction. Thus, the cost is low if the direction of an edge is perpendicular to the gradient direction of the pixel:

$$S_d(\mathbf{v}, \mathbf{u}) = \left| \frac{\mathbf{vu}}{|\mathbf{vu}|} \cdot \frac{\nabla(\mathbf{v})}{|\nabla(\mathbf{v})|} \right|, \quad (12)$$

where  $\mathbf{u}$  is a neighboring node of  $\mathbf{v}$ .

**(4) Contour direction cost**—Because the edge in the graph is bi-direction, we may define that the ROI region is located at the right side of the contour when walking through the contour in the clockwise direction. In order to separate a ROI region with low or high intensity compared to its background, we construct a cost function that compares the normal direction of the contour and the gradient direction:

$$S_c(\mathbf{v}) = \frac{1}{2} \left( 1 + k \cdot \mathbf{n} \cdot \frac{\nabla(\mathbf{v})}{|\nabla(\mathbf{v})|} \right), \quad (13)$$

where  $\mathbf{n}$  is the normal of the contour (see Figure 6(b)),  $k$  is 1 for brighter ROI such as liver parenchyma (enhanced liver tissue is brighter than its neighborhood), and is -1 for darker ROI such as tumor (tumor tends to be darker than normal liver tissues). The default value of  $k$  is 1 and is set to -1 manually for tumor ROI.

**(5) Isocontour cost**—Assume that a local ROI boundary has a similar profile. We define an iso-distance contour map (see Figure 6) of a local contour: if two neighboring pixels are located at the same level of isocontour, its cost is low; otherwise the cost is high. The maximum isocontour value is defined as 8 pixels and the minimum isocontour is -8 pixels. The initial zero iso-contour is the intersection polylines of the initial mesh. To normalize the cost, the value of the cost is divided by 16:

$$S_{Iso}(\mathbf{v}, \mathbf{u}) = |e(\mathbf{v}) - e(\mathbf{u})| / 16, \quad (14)$$

where  $e(\mathbf{v})$  is the isocontour value at pixel  $\mathbf{v}$ .

The local cost function from vertex  $\mathbf{v}$  to vertex  $\mathbf{u}$  is the weighted sum of five aforementioned costs defined on vertex  $\mathbf{v}$  and edge  $\mathbf{vu}$ :

$$S(\mathbf{v}, \mathbf{u}) = \begin{cases} \sum_{Cost}, & \text{if } \mathbf{u}, \mathbf{v} \in B \\ \infty, & \text{otherwise} \end{cases}, \quad (15)$$

where  $\sum_{Cost} = w_{ZX} S_{ZX}(\mathbf{v}) + w_g S_g(\mathbf{v}, \mathbf{u}) + w_d S_d(\mathbf{v}, \mathbf{u}) + w_c S_c(\mathbf{v}) + w_{Iso} S_{Iso}(\mathbf{v}, \mathbf{u})$ ,  $\mathbf{v}, \mathbf{u}$  are two vertices located within the band  $B$  shown in Figure 5. We set  $w_{ZX}=0.15$ ,  $w_g=0.15$ ,  $w_d=0.1$ ,  $w_c=0.5$ , and  $w_{Iso}=0.1$ . These values were set empirically, which may be fine-tuned in different clinical applications and imaging modalities.

We use the dynamic-programming (DP) graph search algorithm [29] to search the local boundaries of a ROI within the band, such as the Dijkstra's algorithm or the A\* algorithm [30]. To accelerate the searching of an optimal path with minimum cost, we employed an active list that sorts nodes by the total cost of starting nodes. This sorted active list takes computation time  $\mathcal{O}(N)$  for sorting of  $N$  nodes (pixels) [30].

The seed points ( $v_i^{Seed}$ ) for searching the optimal path are selected on the intersection polyline at an interval of  $n$  pixels ( $n=60$  in our study). Contour optimization searches the optimal path with minimal cost between each pair of seed points ( $v_i^{Seed}, v_{i+1}^{Seed}$ ) by using Equation (15) and then the resulting optimal paths are connected to form the optimized transversal contour  $C_{Opt}$  of the ROI.

The local boundary constraints ( $\mathbf{V}^{contour}$ ) in constraint set  $\mathcal{Q}$  in Equation (7) is updated by adding a set of boundary points that are selected from the optimized transversal contour  $C_{Opt}$  as the constraint targeting vertices ( $v_i^{TC}$ ). To ensure that the selected  $v_i^{TC}$  is at or near the boundary of the ROI, we applied the following selection criteria:

- The middle segment of each optimal path between ( $v_i^{Seeds}, v_{i+1}^{Seeds}$ ) is selected to reduce the influence of inaccurate positions of initial seed points.
- For each point at the middle segment of a path, the vertex with the lowest cost is selected as the candidate of  $v_i^{TC}$ .
- All the  $v_i^{TC}$  candidates on the transversal contour  $C_{Opt}$  are filtered to remove those that have a roof-edge, considering the segmented ROI has a step-edge.

Mesh transformation and contour optimization is performed in an iterative manner: mesh transformation initializes the transversal contours, and contour optimization updates the constraint targeting vertices for the refinement of the targeting mesh in the next iteration of mesh transformation. At each mesh transformation, ROI mesh is evolving from low to high resolution, as indicated by Equation (1). This iteration between mesh transformation and contour optimization is repeated in a multi-resolution progressively-optimized manner to ensure an efficient and accurate approximation of the ROI mesh.

### III. Liver Segmentation Scheme

By applying the iterative mesh deformation algorithm, we developed a semi-automated segmentation scheme for diseased livers with cancer on CT images. The algorithmic steps of the semi-automated liver segmentation scheme are illustrated in Figure 7.

The mesh segmentation of a liver is initialized by 5 manually identified liver anatomical landmarks and a set of chest wall points  $\mathbf{V}^{chest}$  that are detected automatically [31]. The five initial liver landmarks are illustrated in Figure 8, which are the liver top (superior) point ( $v^t$ ), the liver bottom (inferior) point ( $v^b$ ), the liver posterior point ( $v^p$ ), the liver anterior point ( $v^a$ ), and the liver left point ( $v^l$ ), respectively. Because the right boundary of a liver is adjacent to the chest wall, a set of points at the boundaries of the chest wall is selected as the initial targeting constraint vertices, which are shown as the red points  $v_i^c$  in Figure 8(a).

The initial homeomorphic sphere mesh for constructing  $\mathbf{M}^0$  is built by setting the center at  $\frac{1}{2}(\mathbf{v}^t + \mathbf{v}^b)$  and the radius of  $\frac{1}{2}|\mathbf{v}^t - \mathbf{v}^b|$  as shown in Figure 8(a). To establish the correspondence constraints  $\Omega$ , each targeting vertex on CT images ( $\mathbf{v}_i^{TC}$  in Equation (7)) needs to be mapped to the corresponding vertex on the sphere mesh, which is defined as the nearest vertex on the sphere mesh for each  $\mathbf{v}_i^{TC}$ , as shown in Figure 8(a). By using the initial correspondence constraints  $\Omega_0$ , the sphere mesh is deformed to the initial liver mesh  $\mathbf{M}^0$  based on mesh transformation in Equation (1). The resulting  $\mathbf{M}^0$  is displayed in Figure 8(b).

Because the deformed liver mesh  $\mathbf{M}^0$  intersects with all transversal slices of CT images between the top and bottom landmarks, we may calculate a series of intersection polylines of the mesh on each transversal slice. These intersection polylines are re-sampled at an interval of 60 pixels to generate a set of seed points  $\mathbf{v}_i^{Seed}$ , shown as the red points in Figure 9(a).

Between each pair of neighboring seed points ( $\mathbf{v}_i^{Seed}, \mathbf{v}_{i+1}^{Seed}$ ), an optimal path with minimal cost is searched. The optimal transversal contour is created by connecting all the optimal paths between every pair of seed points ( $\mathbf{v}_i^{Seed}, \mathbf{v}_{i+1}^{Seed}$ ) on the transversal image, shown as the yellow lines in Figure 9(b).

Due to the lack of local boundary constraint in  $\Omega_i$ , the initial intersection polylines and the seed points  $\mathbf{v}_i^{Seed}$  may not closely match the local liver boundary. By contour optimization, the updated transversal contours approximate more accurately to the local boundary than do the initial polylines, as shown the yellow lines in Figure 9(a) and (b) respectively. Thus, we use the middle points ( $\mathbf{v}_i^{mid}$ ) of each optimal path as the targeting points and replace constraint set  $\Omega_{i+1}$  with them to refine the targeting vertices. Occasionally, in case that the  $\mathbf{v}_i^{mid}$  is off the liver boundary, we may adjust or add or delete  $\mathbf{v}_i^{mid}$  interactively. In general, approximately 1~3 pairs of extra correspondences may be needed for one case. Based on the updated constraint set  $\Omega_{i+1}$ , a new mesh  $\mathbf{M}^{i+1}$  is calculated using Equation (1). Figure 9(c) compares the difference between the updated mesh  $\mathbf{M}^{i+1}$ (red) and the initial mesh  $\mathbf{M}^i$  (yellow).

## IV. Experiments and Results

Forty (40) hepatic CT cases with biopsy-confirmed liver cancers or metastases were retrospectively collected for the evaluation of the accuracy and efficiency of our proposed semi-automated liver segmentation scheme. These hepatic CT cases were obtained by multi-detector CT scanners with the following parameter settings: 2.5–5 mm collimation, 1.25–2.5 mm reconstruction interval, 175 mA tube current, and 120 kVp tube voltage. All cases were acquired with use of an intra-venous contrast agent (ISOVUE; GE Healthcare, Milwaukee, WI).

The livers in these 40 hepatic CT cases were segmented manually by two radiologists in consensus on CT images. The volumetric size of these 40 cancerous livers ranged from 1079.2 CC to 4652.3 CC, in which the tumor volume percentages (ratio of *tumor volume*/*liver volume*) ranged from 1.77% to 53.54%. Some of these cases are shown in Figure 1.

The manually-segmented liver volumes were compared with those obtained by the computerized liver volumes segmented by the proposed semi-automated scheme. The computerized liver volumes ranged from 1100.4 CC to 4443.5 CC.

Figure 10 to Figure 12 demonstrate some resulting images of three cases segmented by the proposed liver segmentation scheme. All these cases contain different sizes of tumors. Figure 10 demonstrates a case that was finished without any additional changes of the seeding points, i.e. by using only 5 user-identified liver landmarks. Figure 11 shows a case with a few manual adjustments on couple seed points on three slices. After the first run of the scheme, some seed points on the contours were adjusted manually by the user (see the upper row). After adjustments, the contours were finalized by the second run of the scheme (see the lower row). Figure 12 shows an example case with large tumors. After the initial mesh segmentation, nine seed points were adjusted manual for contour refinement. It takes less than 5 minutes to get the result shown in Figure 12.

We evaluated the performance of the proposed liver segmentation scheme by using the evaluation criterion proposed in [4], which includes five metrics: (1) VOE: volumetric overlap error (%), (2) RVD: relative absolute volume difference (%), (3) ASD: average symmetric surface distance (unit: mm), (4) RSD: root-mean-square of symmetric surface distance (unit: mm), and (5) MSD: maximum symmetric surface distance (unit: mm). According to the criterion [4], a score of 75 is “comparable to human performance” of liver segmentation, which corresponds to VOE=6.4%, RVD=4.7%, ASD=1.0 mm, RSD=1.8 mm, and MSD=19 mm.

The evaluated performance of the proposed scheme is listed in Table 2. Our proposed liver segmentation achieved the average score of  $84.75 \pm 3.69$ . This implies that the accuracy of the proposed liver segmentation scheme is superior to that of averaged “human performance”. In average, the proposed semi-automated liver scheme took approximately 5 minutes including about 2 minutes of computing time. It demonstrates that our scheme is accurate and efficient in segmentation of livers considering that our study used diseased liver cases with pathologies including cancers and metastases, which are in general difficult cases compared to those healthy liver cases (such as liver donor cases in transplantation) in the literatures,.

In order to compare our method with other existing liver segmentation methods in terms of accuracy, running time, ease of use, and consistency, we selected five semi-automated or interactive liver segmentation methods from open-source segmentation software ITK Snap ([www.itksnap.org](http://www.itksnap.org)) and Seg3D ([www.sci.utah.edu/cibc-software/seg3d.html](http://www.sci.utah.edu/cibc-software/seg3d.html)), and compared our method with these five selected methods by using 10 liver cases downloaded from MICCAI database ([www.sliver07.org](http://www.sliver07.org)). We conducted a user study by evaluating the performance of these methods in two user groups: 1) the novice group of five college students, and 2) the expert group of five 3D imaging analysts in radiology department. After one day training of these tools, each participant finished the 10 MICCAI cases using five selected tools as well as our tool. Because interactive manual segmentation methods tends to outperform semi-automated or automated segmentation methods at a high cost of time and effort, all participant were required to apply the same criteria of accuracy in the evaluation

of each method. We compared the accuracy scores, the running time, and the easy-of-use index of each method in two groups. The result is showed in Table 3.

The average accuracy scores of all methods were approximately 80, which indicate the similar levels of accuracy for all tested methods. Compared with the five selected liver segmentation methods, our method demonstrated the highest level of easy-of-use as that of manual contouring, but took only about 5 minutes for finishing a liver segmentation task, which was much more efficient than other methods, in particular about 30-40 minutes for the manual contouring methods. Overall, the expert group outperformed the novice group with higher accuracy score (80.3 vs79.6) and shorter running time (21.2 vs 30.3). However, we observed that the running time in our method was almost constant in both groups, whereas it was approximately 10 min less (about 25%-35%) in expert group than that in novice group for other methods. The low standard deviation of our methods indicate that our liver segmentation scheme achieved a consistent accuracy score among different user groups

## V. Discussion and Conclusion

The proposed semi-automated liver segmentation scheme based on iterative mesh transformation combines the advantages of 3D shape continuity and slice-to-slice constraints to achieve an quick and accurate segmentation with minimal user interaction. From our user study, we observed that our method is quicker and more consistent than other methods in both user groups. This outperformance of our method may result from the algorithm and user familiarity with our method.

- The energy-based contour deformable models (such as snakes) requires that the distances between the initial contours (or mesh) and the resulting contours (or mesh) of a ROI must be sufficiently small. User needs to manually initialize contours on each slice in the neighborhood of ROI or use interpolation to estimate the initial contours. The former may increase the interaction time and the latter is prone-to errors which may increase the error-correction time. In addition, deformable models tends to run a large number of iterations such as 100 or 200 iterations to approximate the contours or mesh of a ROI in images.
- Different from energy-based deformation models, 3D mesh transformation model relies on small number of targeting vertices, five landmarks (feature points) in our liver segmentation scheme, to approximate the initial mesh of a ROI. The sparse LU factorization algorithm is very efficient to solve the minimization of the objective function (solve the resulting mesh) and contour optimization also takes  $O(N\log N)$  to find the optimized ROI contours on each image. In addition, the mesh is updated automatically every time when user corrects one vertex on a contour, i.e. the error-correcting is propagated to neighboring contours, which may substantially reduce the error-correcting time.
- We had to admit that the authors and the participants in the user study were more familiar with the features of our software than those of other tools. This may cause bias to the user study, in particular the user time. The running time in Table 3 relies on the experience of the image analyst and their familiarity to the software to

certain degree. The results might not reflect the best performance of these selected methods.

We acknowledge that the proposed iterative liver segmentation method and the evaluation study have four major limitations.

1. Mapping between a sphere mesh and a ROI boundary: The mesh transformation is established based on the one-to-one homeomorphic mapping between a sphere mesh and a ROI boundary. This assumption of continuous one-to-one mapping may not be easily held in the case of a concave or convex boundary, as shown in Figure 13(a). Because the ROI contour is searched within the band centered by the intersection polyline of the transformed mesh, the resulting contour may miss the concave hole if the band cannot cover the entire hole, as shown in Figure 13(b). Our current solution relies on interactive adding of a targeting point to cover the concave hole. In future, we will replace the sphere mesh with an initial liver mesh that has the same topology including these major concave holes.
2. Parameter setting: There are several important parameters in our liver segmentation scheme that were set empirically, such as the weights in equations (1), (5) and (15). We did not explore the possibility to optimize them further in this study because an optimal setting of the parameters may be related to the specific liver scanning protocols. Thus, we chose the parameters to be general, not optimized for our specific liver scanning protocol. Another important parameter is the bandwidth for searching 2D optimal contours. In our current implementation, we used a bandwidth of 40 pixels in each iteration for simplicity. A large bandwidth may erroneously take the neighboring large gradient boundaries such as bone as the liver boundaries, as pointed by the white arrows in Figure 13(b). An ideal solution is to adjust the bandwidth parameter at each iteration: from a large bandwidth at initial iteration to a small bandwidth at the final iteration.
3. User interaction: In comparison to previous semi-automated segmentation methods, our method reduced the user-interaction to as little as identification of five landmarks. However, users interaction is still needed to review and adjust targeting points on 2D transversal contours after each iteration, in particular at first or second iterations. This is the major interaction time spent in our current segmentation scheme. In future, this interactive identification of five landmarks may be lifted by automated landmark labeling and neighboring organs segmentation. In addition, the selection of constraint targeting points on transversal contours will be refined to minimize the constraint set with the most salient features on the contours.
4. Evaluation study: The hepatic CT images for evaluation were retrospectively collected from a radiochemotherapy study for patients with primary liver cancer or metastasis, and scanned with our institutional abdominal imaging protocol by using an intra-venous contrast agent. These cancer cases may present different texture patterns compared to other liver pathologies such as fatty liver or cirrhosis, and other liver imaging protocols. Further evaluation is required to apply this proposed liver segmentation scheme to hepatic CT images with other liver pathologies.

In summary, we proposed a semi-automated segmentation scheme for diseased livers with cancers based on an iterative mesh transformation constrained by optimal transversal contours. The initial evaluation demonstrated that this semi-automated liver segmentation scheme can achieve an accurate and efficient 3D segmentation for diseased livers by using as little as five user-identified landmarks and an average of 5 minutes. Although this scheme was developed in the context of segmentation of diseased liver with tumors in hepatic CT images, this method may provide a general scheme for semi-automated segmentation of other organs or tumors.

## Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

## Acknowledgments

The authors thank Jeffrey Clark, MD, Anand Singh, MD, and Jennifer McGowan, RT from Department of Radiology at Massachusetts General Hospital for technical discussion, evaluation and statistical assistance with this manuscript.

## Appendix 1: Solving the minimization problem in Equation (1)

Based on Equation (1) in Section II, mesh transformation is a minimization problem to solve the objective function  $f(\mathbf{M}^i)$  formed by the weighted sum of three constraints: the smoothing constraint ( $C_S$ : Equation (6)), the deforming constraint ( $C_D$ : Equation (7)), and the distance constraint ( $C_L$ : Equation (8)).

For an edge  $\mathbf{E}_j$ , let the four vertices around  $\mathbf{E}_j$  be  $\mathbf{v}_{E_j[x]}$ , where  $E_j[x]$  is the index of the vertices in the mesh,  $x = 1 \sim 4$ . Let  $t_{E_j}^{(r,c)}$  be the element  $(r, c)$  of the  $3 \times 3$  matrix  $[\mathbf{T}^S]^{-1}$  in Equation (4), where  $r, c = 1 \sim 3$ . Equation (6) can be represented in the vector form of  $\mathbf{v}_i^{ROI}$ , shown in Equation (A.1).

$$\begin{aligned}
 C_S(\mathbf{M}) &= \sum_{i=1}^{|\mathbf{Z}(\mathbf{M})|} \sum_{\substack{j \neq k=1, \\ \mathbf{E}_j \in \mathbf{E}(\mathbf{Z}_i), \\ \mathbf{E}_k \in \mathbf{E}(\mathbf{Z}_i)}}^3 \|(T(\mathbf{E}_j) - T(\mathbf{E}_k))\|_F^2 \\
 &= \sum_{i=1}^{|\mathbf{Z}(\mathbf{M})|} \sum_{\substack{j \neq k=1, \\ \mathbf{E}_j \in \mathbf{E}(\mathbf{Z}_i), \\ \mathbf{E}_k \in \mathbf{E}(\mathbf{Z}_i)}}^3 \sum_{c=1}^3 \sum_{r=1}^3 \| \sum_{E_j} t_{E_j}^{(r,c)} (\mathbf{v}_{E_j[r+1]}^{ROI} - \mathbf{v}_{E_j[1]}^{ROI}) - \sum_{r=1}^3 t_{E_k}^{(r,c)} (\mathbf{v}_{E_k[r+1]}^{ROI} - \mathbf{v}_{E_k[1]}^{ROI}) \|^2 \quad (\text{A.1})
 \end{aligned}$$

Let  $N^S, N^D, N^L$  be the numbers of the items in Equation (A.1), (7) and (8), respectively.  $N^S = |\mathbf{Z}(\mathbf{M})| \times 3 \times 3$ ,  $N^D = |\mathcal{Q}|$ ,  $N^L = |\mathbf{E}(\mathbf{M})|$ , where  $|\mathbf{Z}(\mathbf{M})|$  is the number of triangle faces,  $|\mathcal{Q}|$  is the number of targeting constraint vertices in the constraint set, and  $|\mathbf{E}(\mathbf{M})|$  is the number of edges in the mesh.

Therefore, the three constraints  $C_*$  (\*= $S,D,L$ ) can be represented by a linear system:

$$C_* = \sum_{\xi=1}^{N^*} \|A_{\xi}^* V + B_{\xi}^*\|_F^2, \quad (A.2)$$

where  $A_{\xi}^*$  is an  $N^* \times n$  matrix,  $n$  is the number of vertices in the mesh, and  $B_{\xi}^*$  is an  $N^* \times 1$  vector. For the smooth constraint  $C_S$ , based on Equation (A.1),  $B_{\xi}^S$  is a zero vector and  $A_{\xi}^S$  is a sparse matrix as shown in Figure (A.1).

	...	$E_j[1]$	...	$E_j[2]$	...	$E_j[3]$	...	$E_j[4]$	...	$E_k[1]$	...	$E_k[2]$	...	$E_k[3]$	...	$E_k[4]$	...
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
$\xi$	...	$e_j[1]$	...	$e_j[2]$	...	$e_j[3]$	...	$e_j[4]$	...	$e_k[1]$	...	$e_k[2]$	...	$e_k[3]$	...	$e_k[4]$	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
$N^S$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**Figure A.1.**

Distribution of non-zero elements in  $A_{\xi}^S$ , which is a  $N^S \times n$  matrix and all the elements except those labeled as  $e_j$  or  $e_k$  in the  $\xi^{th}$  row are zero. The top labels are the indices of vertices, and the left labels are number of item in Equation (A.1), where  $\xi = 9i + 3k + c$ ,  $i, k$  and  $c$  are variables of Equation (A.1). If two indices are equal, for example,  $E_j[2]=E_k[3]=i$ , the element in  $(\xi, i)$  is  $e_j[2]+e_k[3]$ .

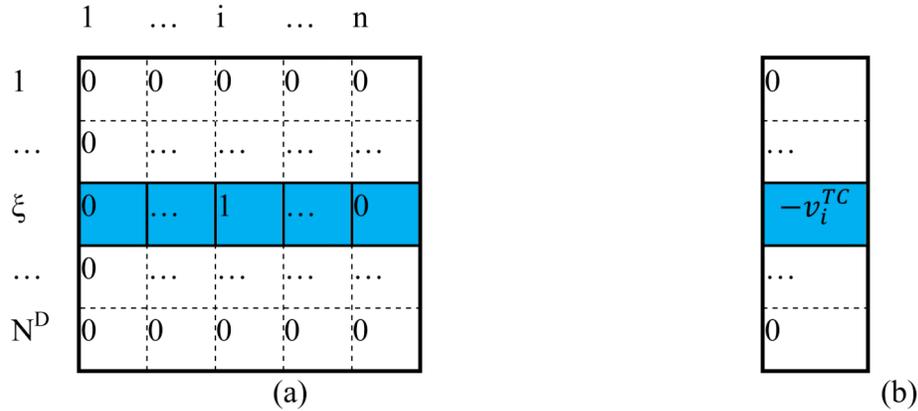
The non-zero elements in  $A_{\xi}^S$  can be calculated from Equation (A.1):

$$\begin{aligned} e_j[1] &= -\sum_{r=1}^3 t_{E_j}^{(r,c)}, e_j[i] = t_{E_j}^{(i-1,c)}, i=2, 3, 4. \\ e_k[1] &= \sum_{r=1}^3 t_{E_k}^{(r,c)}, e_k[i] = -t_{E_k}^{(i-1,c)}, i=2, 3, 4. \end{aligned} \quad (A.3)$$

For the deforming constraint  $C_D$ ,  $A_{\xi}^D$  and  $B_{\xi}^D$  in the form of linear system of Equation (A.1) can be represented in Figure (A.2). For the distance constraint  $C_L$ ,  $A_{\xi}^L$  and  $B_{\xi}^L$  are shown in Figure (A.3).

The distribution of the non-zero elements in  $A_{\xi}^*$  and  $B_{\xi}^*$  (Figure A.1, A.2 and A.3) indicates that:  $(A_i^*)^T B_j^* = 0 (i \neq j)$ . Thus, Equation (A.2) can be rewritten as:

$$\begin{aligned}
C_* &= \sum_{\xi}^{N^*} \|A_{\xi}^* \mathbf{V} \mathbf{ROI} + B_{\xi}^*\|_F^2 = \sum_{\xi}^{N^*} ((A_{\xi}^* \mathbf{V} \mathbf{ROI})^T + (B_{\xi}^*)^T) \cdot (A_{\xi}^* \mathbf{V} \mathbf{ROI} + B_{\xi}^*) \\
&= \sum_{\xi}^{N^*} \left( \mathbf{V} \mathbf{ROI}^T (A_{\xi}^*)^T A_{\xi}^* \mathbf{V} \mathbf{ROI} + \mathbf{V} \mathbf{ROI}^T (A_{\xi}^*)^T B_{\xi}^* + (B_{\xi}^*)^T A_{\xi}^* \mathbf{V} \mathbf{ROI} + (B_{\xi}^*)^T B_{\xi}^* \right) \quad (A.4)
\end{aligned}$$



**Figure A.2.**

(a). Distribution of non-zero elements in  $A_{\xi}^D$ , which is a  $N^D \times n$  matrix and all the elements except element  $(\xi, i)$  are zero, where  $\xi$  is the index of constraint in  $\Omega$ . (b) Distribution of non-zero elements in  $B_{\xi}^D$ .

The partial derivatives of  $C_*$  is:

$$\frac{\partial C_*}{\partial \mathbf{V} \mathbf{ROI}} = 2 \left( \sum_{\xi}^{N^*} (A_{\xi}^*)^T A_{\xi}^* \mathbf{V} \mathbf{ROI} + 2 \sum_{\xi}^{N^*} (A_{\xi}^*)^T B_{\xi}^* \right) \quad (A.5)$$

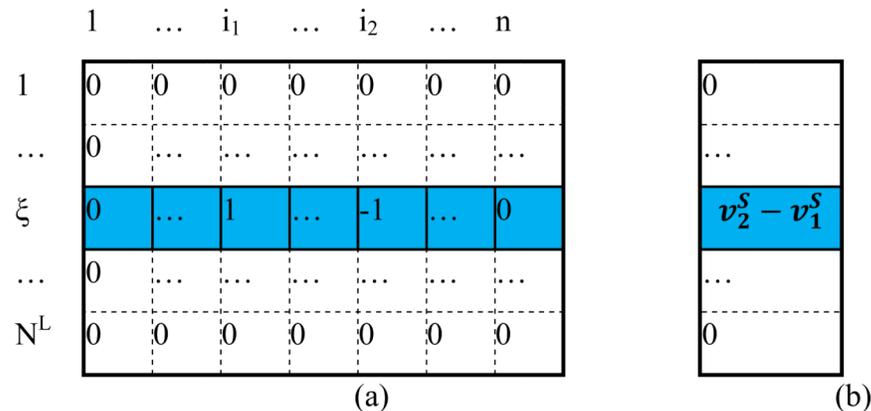
Thus, the partial derivatives of  $f$  has the form below:

$$\begin{aligned}
\frac{\partial f}{\partial \mathbf{V} \mathbf{ROI}} &= 2(w_s \sum_{\xi}^{N^S} (A_{\xi}^S)^T A_{\xi}^S \\
&\quad + w_D \sum_{\xi}^{N^D} (A_{\xi}^D)^T A_{\xi}^D \\
&\quad + w_L \sum_{\xi}^{N^L} (A_{\xi}^L)^T A_{\xi}^L) \mathbf{V} \mathbf{ROI} + 2(w_s \sum_{\xi}^{N^S} (A_{\xi}^S)^T B_{\xi}^S \\
&\quad + w_D \sum_{\xi}^{N^D} (A_{\xi}^D)^T B_{\xi}^D \\
&\quad + w_L \sum_{\xi}^{N^L} (A_{\xi}^L)^T B_{\xi}^L) \quad (A.6)
\end{aligned}$$

Finally, the minimization problem in Equation (1) can be solved by evaluating the following equation:

$$\frac{\partial w_0 \mathbf{f}(\mathbf{M}_0)}{\partial \mathbf{v}^{ROI}} + \frac{\partial w_1 \mathbf{f}(\mathbf{M}_1)}{\partial \mathbf{v}^{ROI}} + \dots + \frac{\partial w_\theta \mathbf{f}(\mathbf{M}_\theta)}{\partial \mathbf{V}^{ROI}} = 0 \quad (\text{A.7})$$

In terms of Equation (A.6), Equation (A.7) can be converted to a standard vector-formed linear system, which can be solved by using the sparse LU factorization algorithm [28].



**Figure A.3.**

(a) Distribution of non-zero elements in  $\mathbf{A}_\xi^L$ , which is a  $N^L \times n$  matrix and all the elements except elements  $(\xi, i_1)$  and  $(\xi, i_2)$  are zero, where  $i_1$  and  $i_2$  are the indices of  $\mathbf{v}_1^{ROI}$  and  $\mathbf{v}_2^{ROI}$  in the mesh, and  $\xi$  is the index of edge  $\mathbf{E}(\mathbf{M})$  in the mesh  $\mathbf{M}$ . (b) Distribution of non-zero elements in  $\mathbf{B}_\xi^L$ .

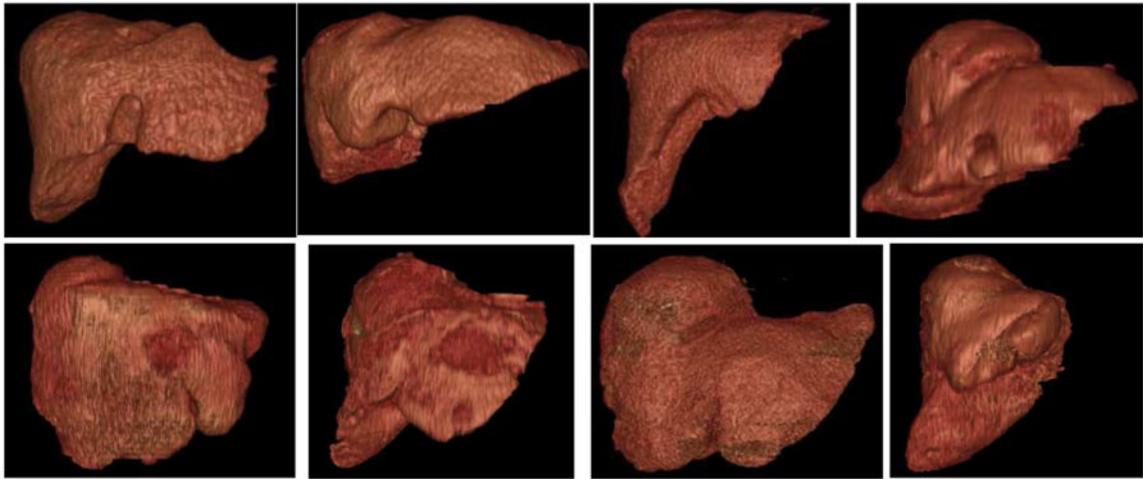
## References

1. Center MM, Jemal A. International trends in liver cancer incidence rates. *Cancer Epidemiol Biomarkers Prev.* 2011; 20(11):2362–2368. [PubMed: 21921256]
2. National Cancer Institute. Metastatic cancer fact sheet. National Cancer Institute; 2013.
3. Oto A, Tamm EP, Szklaruk J. Multidetector row CT of the liver. *Radiol Clin North Am.* 2005; 43(5):827–48, vii. [PubMed: 16098342]
4. Heimann T, van Ginneken B, Styner MA, et al. Comparison and evaluation of methods for liver segmentation from CT datasets. *IEEE Trans Med Imaging.* 2009; 28(8):1251–1265. [PubMed: 19211338]
5. Campadelli P, Casiraghi E, Esposito A. Liver segmentation from computed tomography scans: A survey and a new algorithm. *Artif Intell Med.* 2009; 45(2-3):185–196. [PubMed: 19059767]
6. Heimann T, Wolf I, Meinzer HP. Active shape models for a fully automated 3D segmentation of the liver--an evaluation on clinical data. *Med Image Comput Comput Assist Interv.* 2006; 9(Pt 2):41–48. [PubMed: 17354754]
7. Zhang X, Tian J, Deng K, et al. Automatic liver segmentation using a statistical shape model with optimal surface detection. *IEEE Trans Biomed Eng.* 2010; 57(10):2622–2626. [PubMed: 20615804]
8. Heimann T, Meinzer HP. Statistical shape models for 3D medical image segmentation: A review. *Med Image Anal.* 2009; 13(4):543–563. [PubMed: 19525140]
9. Okada T, Shimada R, Hori M, et al. Automated segmentation of the liver from 3D CT images using probabilistic atlas and multilevel statistical shape model. *Acad Radiol.* 2008; 15(11):1390–1403. [PubMed: 18995190]

10. Linguraru MG, Sandberg JK, Li Z, et al. Automated segmentation and quantification of liver and spleen from CT images using normalized probabilistic atlases and enhancement estimation. *Med Phys.* 2010; 37(2):771–783. [PubMed: 20229887]
11. Park H, Bland PH, Meyer CR. Construction of an abdominal probabilistic atlas and its application in segmentation. *IEEE Trans Med Imaging.* 2003; 22(4):483–492. [PubMed: 12774894]
12. Liu F, Zhao B, Kijewski PK, et al. Liver segmentation for CT images using GVF snake. *Med Phys.* 2005; 32(12):3699–3706. [PubMed: 16475769]
13. Zhang H, Yang L, Foran DJ, et al. 3d segmentation of the liver using free-form deformation based on boosting and deformation gradients. *Proc IEEE Int Symp Biomed Imaging.* 2009; 5193092:494–497. [PubMed: 19997530]
14. Yao J, Chen D. Live level set: A hybrid method of livewire and level set for medical image segmentation. *Med Phys.* 2008; 35(9):4112–4120. [PubMed: 18841864]
15. Fernandez-de-Manuel L, Rubio JL, Ledesma-Carbayo MJ, et al. 3D liver segmentation in preoperative CT images using a level-sets active surface method. *Conf Proc IEEE Eng Med Biol Soc.* 2009; 2009:3625–3628. [PubMed: 19964309]
16. Suzuki K, Kohlbrenner R, Epstein ML, et al. Computer-aided measurement of liver volumes in CT by means of geodesic active contour segmentation coupled with level-set algorithms. *Med Phys.* 2010; 37(5):2159–2166. [PubMed: 20527550]
17. Jimenez-Carretero D, Fernandez-de-Manuel L, Pascau J, et al. Optimal multiresolution 3D level-set method for liver segmentation incorporating local curvature constraints. *Conf Proc IEEE Eng Med Biol Soc.* 2011; 2011:3419–3422. [PubMed: 22255074]
18. Hu YC, Grossberg MD, Wu A, et al. Interactive semiautomatic contour delineation using statistical conditional random fields framework. *Med Phys.* 2012; 39(7):4547–4558. [PubMed: 22830786]
19. Beichel R, Bornik A, Bauer C, Sorantin E. Liver segmentation in contrast enhanced CT data using graph cuts and interactive 3D segmentation refinement methods. *Med Phys.* 2012; 39(3):1361–1373. [PubMed: 22380370]
20. Rusko L, Bekes G, Fidirich M. Automatic segmentation of the liver from multi- and single-phase contrast-enhanced CT images. *Med Image Anal.* 2009; 13(6):871–882. [PubMed: 19692288]
21. Foruzan AH, Zoroofi RA, Hori M, Sato Y. A knowledge-based technique for liver segmentation in CT data. *Comput Med Imaging Graph.* 2009; 33(8):567–587. [PubMed: 19747798]
22. Zhang S, Zhan Y, Dewan M, et al. Towards robust and effective shape modeling: Sparse shape composition. *Med Image Anal.* 2012; 16(1):265–277. [PubMed: 21963296]
23. Ji H, He J, Yang X, et al. ACM-based automatic liver segmentation from 3-D CT images by combining multiple atlases and improved mean-shift techniques. *IEEE J Biomedical and Health Informatics.* 2013; 17(3):690–698.
24. McInerney T, Terzopoulos D. Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE Trans Med Imaging.* 1999; 18(10):840–850. [PubMed: 10628944]
25. Miller JV, Breen DE, Lorensen WE, et al. Geometrically deformed models: A method for extracting closed geometric models from volume data. *SIGGRAPH Comput Graph.* 1991; 25(4): 217–226.
26. Chenyang, Xu; Prince, JL. Snakes, shapes, and gradient vector flow. *Image Processing, IEEE Transactions on.* 1998; 7(3):359–369.
27. Sumner RW, Popović Jovan. Deformation transfer for triangle meshes. *ACM Trans Graph.* 2004; 23(3):399–405.
28. Davis TA. Algorithm 832: UMFPACK V4.3---an unsymmetric-pattern multifrontal method. *ACM Trans Math Softw.* 2004; 30(2):196–199.
29. Mortensen, EN.; Barrett, WA. Intelligent scissors for image composition. Presented at Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques; 1995;
30. Sniedovich, M. *Dynamic Programming: Foundations and Principles.* Second. 2010.
31. Cai W, Lee EY, Vij A, et al. MDCT for computerized volumetry of pneumothoraces in pediatric patients. *Acad Radiol.* 2011; 18(3):315–323. [PubMed: 21216160]

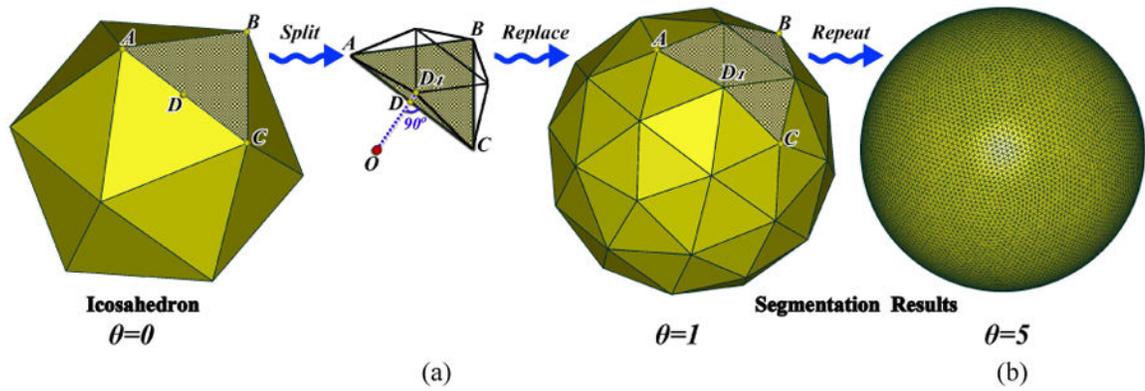
### Highlights

- An iterative mesh transformation algorithm for liver segmentation is proposed.
- Multi-resolution transformation optimization controls the 3D shape of a liver mesh.
- Dynamic-programming searches the 2D liver contours on CT images.
- A semi-automated segmentation scheme for diseased livers is developed.
- Interaction is reduced to as little as five user-identified landmarks.



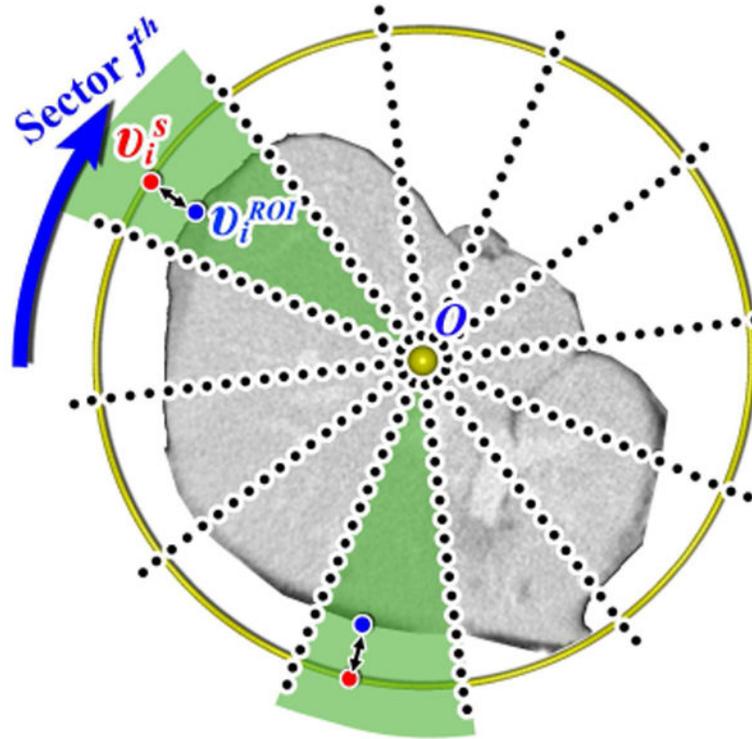
**Figure 1.**

Examples of the inter-patient variability in sizes and shapes of diseased livers from patients with cancers or metastasis. All the patients were scanned by the same protocol at supine position. All the resulting images are displayed in the anterior viewing orientation. The sizes of livers vary significantly with more than four times of difference and more importantly the shapes vary substantially too. The CT intensities of liver parenchyma and cancerous tissues are apparently different.



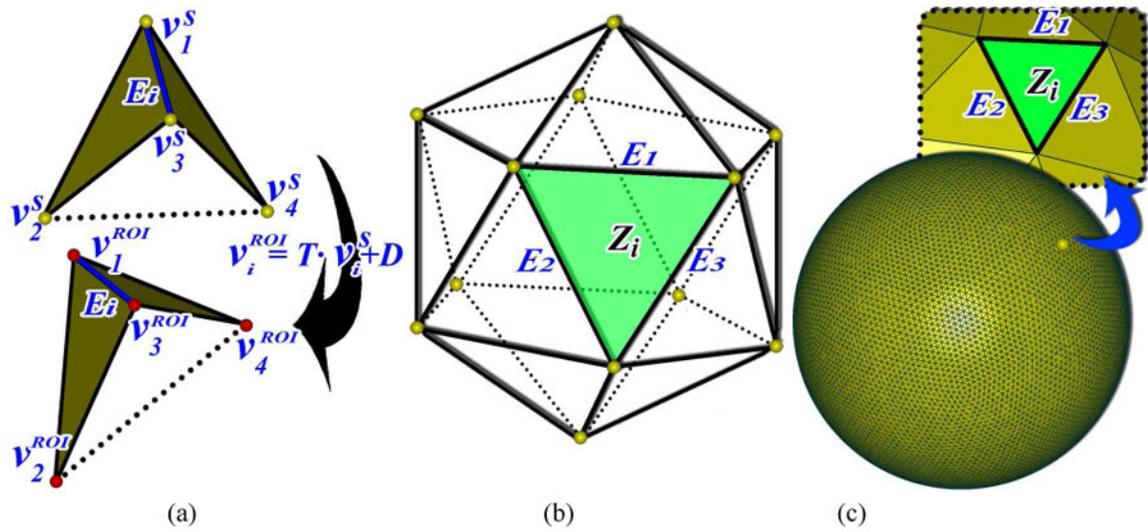
**Figure 2.**

Mesh evolution in the segmentation of a sphere. (a) Initial mesh of an icosahedron is evolved by replacing of a triangle face with 4 smaller triangle faces. The newly-added targeting vertex  $D_1$  is the boundary point of the middle point  $D$  of edge  $AC$  along the direction of  $OD$ , where  $O$  is the center point of the sphere. (b) The resulting mesh after 5 times of evolution, which consists of 10242 vertices, 30720 edges and 20480 triangle faces.



**Figure 3.**

$v_i^S$  is a vertex on the mesh of a sphere and  $v_i^{ROI}$  is a vertex on the boundary of a ROI, such as a liver. When the size of the section is small enough, there is a continuous one-to-one mapping between the ROI boundary point ( $v_i^{ROI}$ ) and the sphere mesh vertex ( $v_i^S$ ).



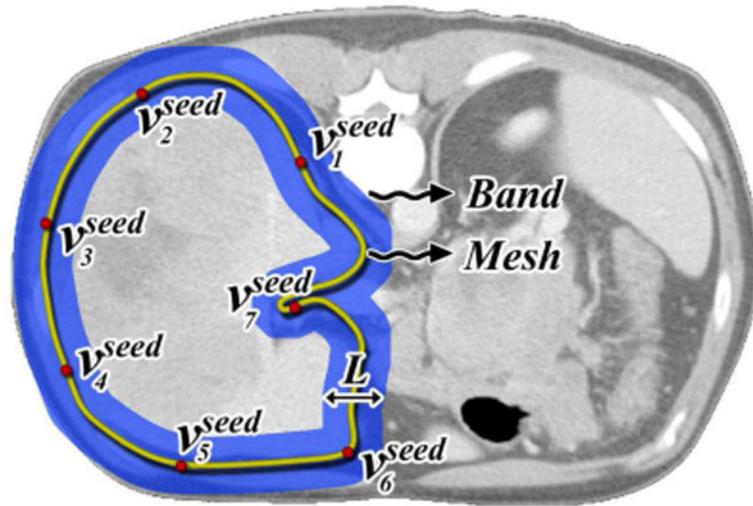
**Figure 4.**

(a)  $T$  is a  $3 \times 3$  affine transformation matrix defined by four non-coplanar vertices

( $v_1 \sim v_4$ ) around an edge  $E_i$ .  $v_i^S$  is the known variable before deformation and  $v_i^{ROI}$  is the

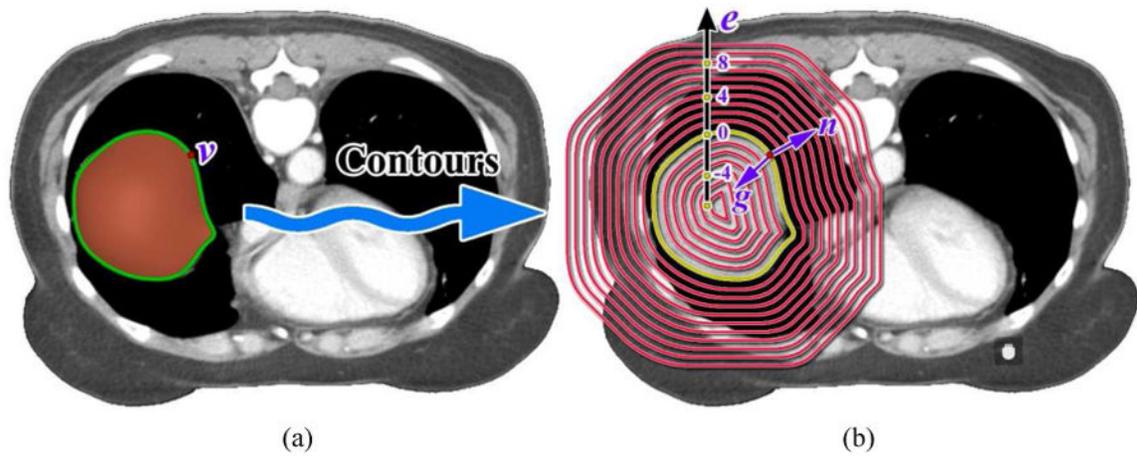
unknown variable after deformation. (b) The roughest face on the original icosahedron. (c)

The finest face on the segmented mesh by  $\theta$  times of evolution.  $Z_i$  is a triangle face of the mesh.



**Figure 5.**

The band (marked in blue) is centered by the intersection polylines of the mesh. The seed points ( $v_i^{seed}$ ) on the intersection polyline are used to search the local boundary of a ROI on the transversal image plane.  $L$  is the width of the band, which is set to 40 in this study.

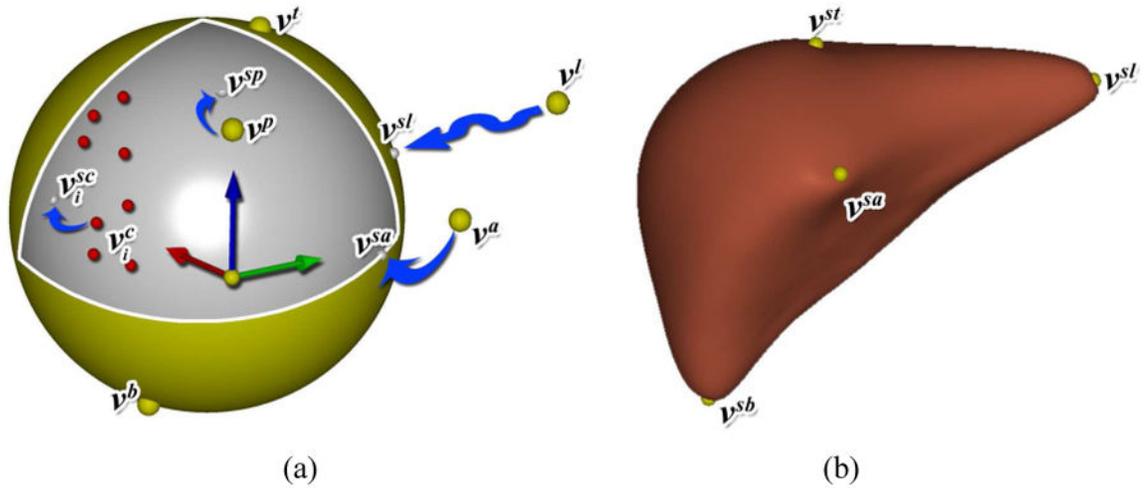


**Figure 6.**

The construction of an isocontour map on a transversal image plane. (a) The green contour is the intersection polylines of the initial mesh on the 2D transversal image, which is defined as the zero-isocontour. The region of the ROI is marked in light brown color, which is the inner region of the contour. (b) The  $e$  axis is the distance to the zero-isocontour: inner region has negative distances, whereas outer region has positive distance.  $n$  is the normal of the contour (from inner to outer) and  $g$  is the direction of the local image gradient. The maximum distance value is defined 8 pixels and the minimum is -8 pixels.

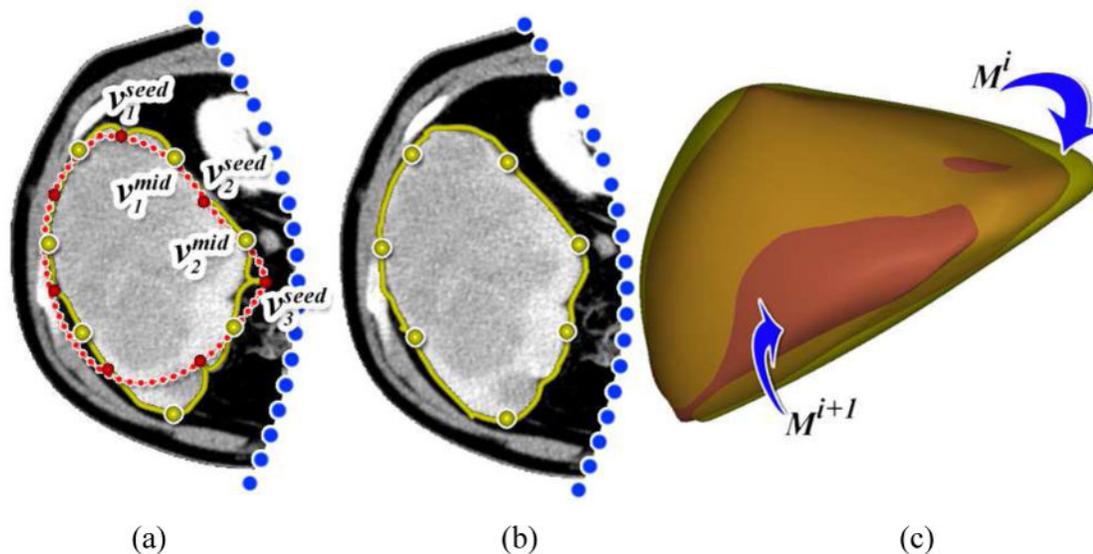
- a) Initialization: Initialize constraints  $\Omega_0$  with 5 liver landmarks identified manually and a set of chest wall points detected automatically.
- b) Initialize the liver mesh  $M^0$  using Equation (1) based on  $\Omega_0$ .
- c) Contour calculation: Calculate the intersection polylines ( $C_i^{init}$ ) of the liver Mesh  $M^i$  on each 2D transversal image plane in CT data.
- d) Contour optimization: Optimize the transversal contours on each transversal image by the DP-based optimal path searching algorithm, the result is marked as  $C_i^{opt}$ .
- e) Constraint updating: Update the constraint targeting vertices ( $v_i^{TC}$ ) with the selected boundary points  $V^{contour}$  from  $C_i^{opt}$  and assign the new correspondences constraints of  $V^{contour}$  to  $\Omega_{i+1}$ .
- f) Mesh transformation: Deform the liver mesh  $M^{i+1}$  using Equation (1) based on  $\Omega_{i+1}$ .
- g) If ( $i < \theta$ )  $i=i+1$ , goto c).

**Figure 7. The semi-automated liver segmentation scheme**



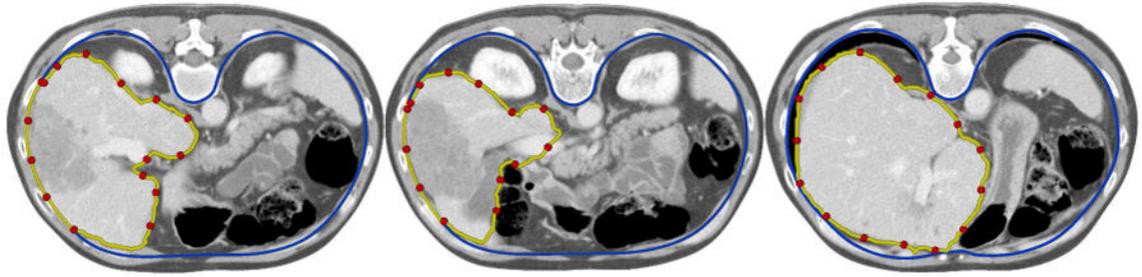
**Figure 8.**

Liver mesh segmentation is initialized by five user-identified liver anatomical landmarks and a set of points from the chest wall detected automatically. (a) Five liver-landmarks  $\mathbf{V}^{landmark} = \{v^t, v^b, v^p, v^a, v^l\}$  are manually identified and the chest wall points  $\mathbf{V}^{chest} = \{v_i^C\}$  are detected in an automated manner. Their correspondence points on the sphere mesh are  $\{v^{st}, v^{sb}, v^{sp}, v^{sa}, v^{sl}, v_i^{SC}\}$ . (b) The initialized liver mesh  $M^0$  is deformed based on the constraint set  $\Omega_0$  using Equation (1).



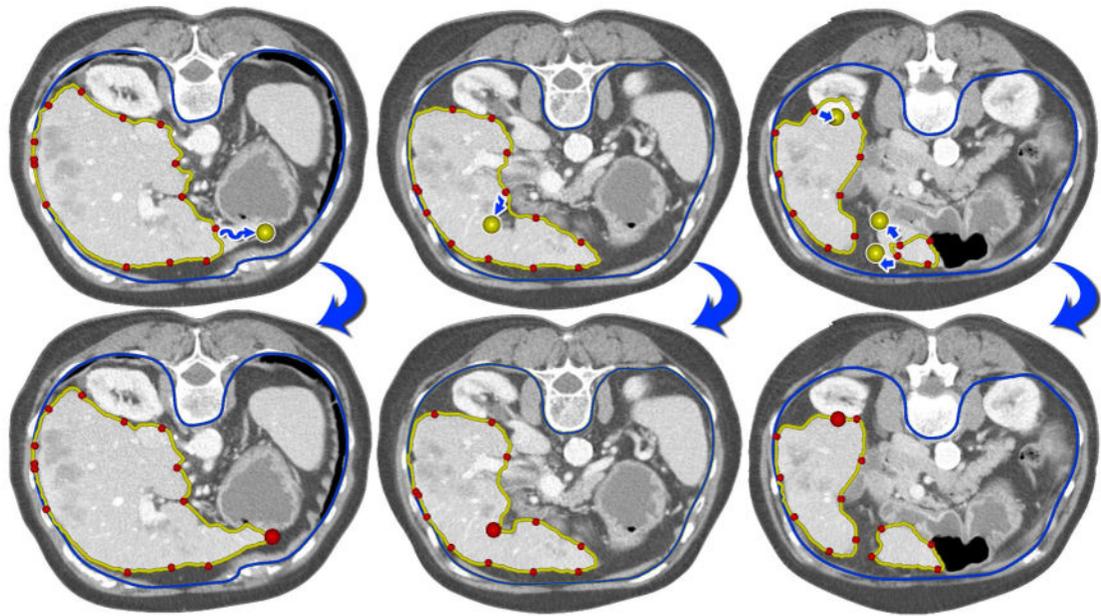
**Figure 9.**

Iterations between mesh transformation and contour optimization in a liver segmentation process. The red line is the intersection polylines of mesh  $M^i$  on the transversal slice. Red points are seed points re-sampled at an interval of 60 pixels on the intersection polylines. The yellow line is the optimal path calculated using the re-sampled seed points ( $v_i^{seed}$ ). The initial seed points are replaced by the middle point ( $v_i^{mid}$ ) at each segment of the optimal path, which is selected as the targeting point and added to the constraint set  $\Omega_{i+1}$ . (b) The intersection polylines are replaced by the optimal path re-calculated using middle points ( $v_i^{mid}$ ) in (a). Please note the re-calculated contour is closer to the liver boundary compared to the initial optimal path in (a). (c) The red mesh is the updated mesh  $M^{i+1}$  by using the updated constraints  $\Omega_{i+1}$ , whereas the yellow mesh is the initial mesh  $M^i$  in (a). The default number of iterations is 3.



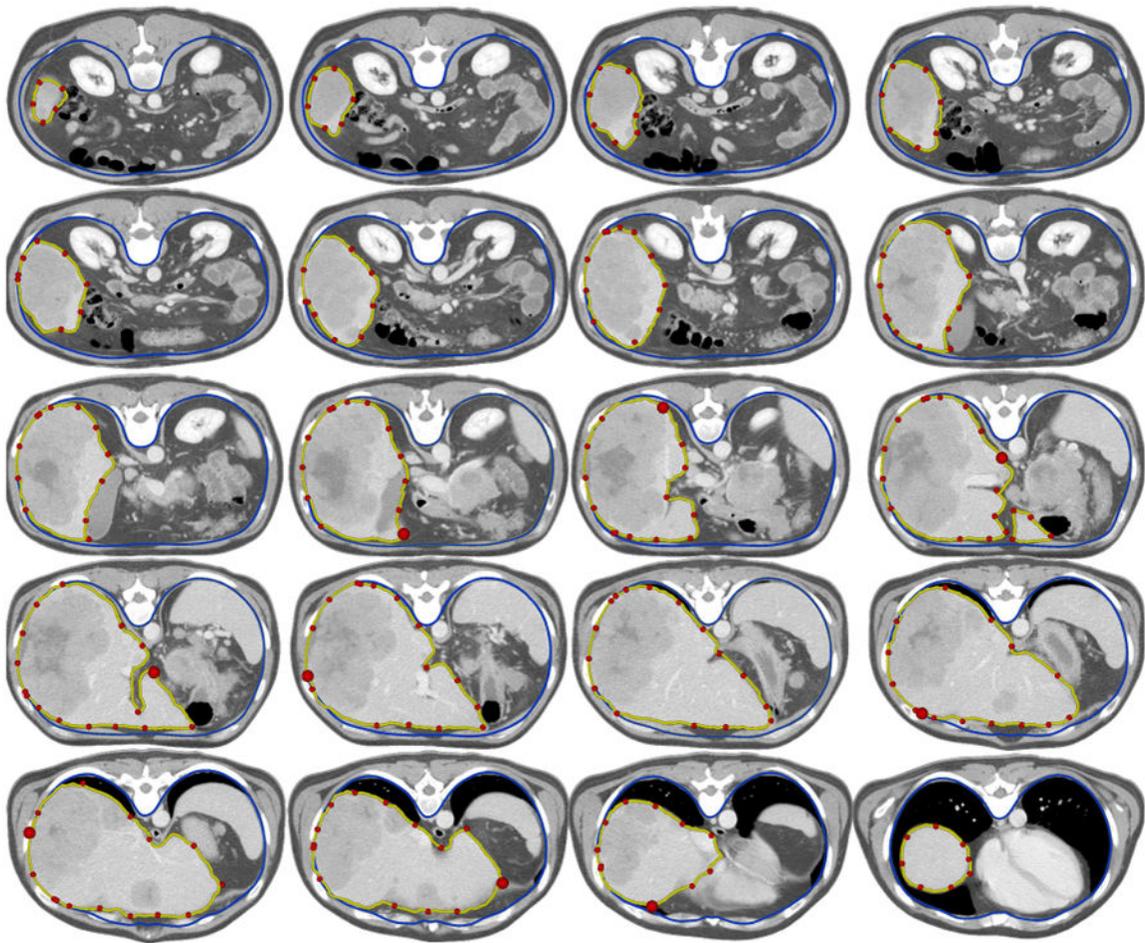
**Figure 10.**

Resulting liver contours (yellow curves) without additional manual adjusting seeds, i.e. the resulting contours were generated by using 5 liver anatomical landmarks identified by users. The contours of chest wall (blue curves) are generated by an automated segmentation method for chest cavity.



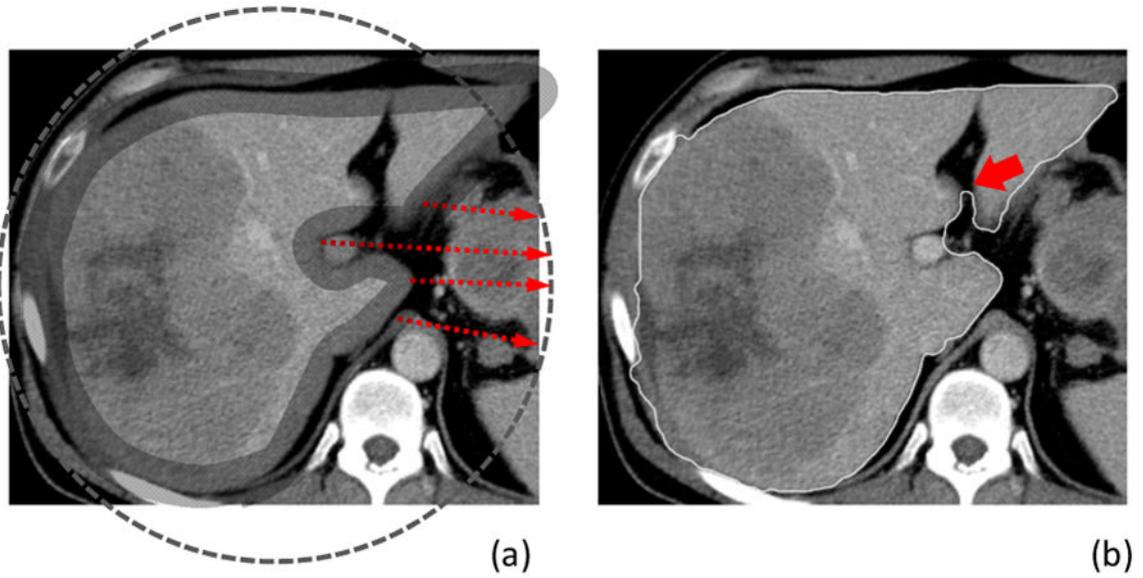
**Figure 11.**

The upper row is the initial contours generated by the scheme with 5 liver landmarks. The arrows indicate the seeds need to be adjusted interactively. The bottom row shows the resulting contours after the adjustments of the seeding points.



**Figure 12.**

Resulting contours of a liver with a large amount of tumors. In this case with 57 original CT slices, 9 seeding points (the large red points) were manually adjusted after the initial mesh segmentation. It took less than 90s to adjust the ambiguous seed points. The total time for the entire segmentation is approximately 5 minutes.



**Figure 13.**

(a) The continuous one-to-one mapping between a sphere mesh and a ROI mesh may not be easily held in the case of a concave hole, as indicated by the red arrow lines between the intersection polylines and a sphere, and the shading band area. (b) If the band cannot cover the entire concave hole, the resulting contour may miss part of the boundary of the concave hole, as indicated by the red arrow. On the other hand, large bandwidth may erroneously take the neighboring large gradient boundaries as liver boundary such as the bone or abdominal cavity pointed by the white arrows.

Number of vertices, edges, and faces of meshes at different step  $\theta$ . Mesh evolution indicates  $V(M_{\theta-1}) \subset V(M_{\theta})$ . In Figure 2, meshes of a sphere at  $\theta = 0, 1, \dots, 5$  are illustrated.

**Table 1**

$\theta$	0	1	2	3	4	5
Vertices	12	42	162	642	2562	10242
Edges	30	120	480	1920	7680	30720
Faces	20	80	320	1280	5120	20480

Performance of the proposed liver segmentation scheme in the 40 liver CT cases with cancers or metastases in our study.

**Table 2**

<i>Part I (40 cases)</i>	Accuracy					Score					
	VOE	RVD	ASD	RSD	MSD	VOE	RVD	ASD	RSD	MSD	Overall
<b>Average</b>	5.97	2.46	0.54	1.15	7.82	76.69	86.91	86.45	83.99	89.71	84.75
<b>Maximum</b>	7.53	5.54	1.24	3.72	25.00	81.79	99.09	92.64	89.84	93.15	89.21
<b>Minimum</b>	4.66	0.17	0.29	0.73	5.20	70.58	70.51	68.89	48.35	67.11	68.64
<b>Std. Dev</b>	0.70	1.02	0.19	0.50	3.37	2.74	5.40	4.76	6.95	4.44	3.69

**Table 3**

Comparison of our method with five liver segmentation methods selected from ITK-Snap ([www.itksnap.org](http://www.itksnap.org)) and Seg3D ([www.sci.utah.edu/cibc-software/seg3d.html](http://www.sci.utah.edu/cibc-software/seg3d.html)) in terms of accuracy, running time, easy-of-use and consistency by using 10 liver cases downloaded from MICCAI database. We apply an index (1~5) to indicate easy-of-use: 1 means easiest and 5 means most difficult. Running time is the average of the times spent on 10 cases by 5 participants in each group.

<b>(a) Results of novice group with 5 participants.</b>				
<b>Methods and Description</b>	<b>Average Score</b>	<b>Std. Dev.</b>	<b>Running Time (minutes)</b>	<b>Easy-Of-Use</b>
Manual Contouring (ITK Snap)	80.15	1.21	43	1
Region Competition Snakes (ITK Snap)	79.01	1.64	31	4
Edge Based Snake (ITK Snap)	80.01	2.12	29	4
Manual Paint and Contouring (Seg3D)	79.56	1.87	44	2
Speedline (Seg3D)	79.05	1.74	30	3
Our method	79.72	1.17	5	1
<b>Average</b>	79.60	1.63	30.3	
<b>(b) Results of expert group with 5 participants.</b>				
<b>Methods and Description</b>	<b>Average Score</b>	<b>Std. Dev.</b>	<b>Running Time (minutes)</b>	<b>Easy-Of-Use</b>
Manual Contouring (ITK Snap)	81.25	1.63	32	1
Region Competition Snakes (ITK Snap)	79.85	1.63	20	4
Edge Based Snake (ITK Snap)	80.79	1.60	19	4
Manual Paint and Contouring (Seg3D)	80.26	1.80	31	2
Speedline (Seg3D)	79.67	1.65	20	3
Our method	80.02	1.22	4.7	1
<b>Average</b>	80.30	1.58	21.1	