Liver Workbench: A Tool Suite for Liver and Liver Tumor Segmentation and Modeling

Jiayin Zhou, Wei Xiong, Feng Ding, Weimin Huang, Tian Qi,
Zhimin Wang, Thiha Oo, and Sudhakar Kundapur Venkatesh

Abstract. Robust and efficient liver and tumor segmentation tools from CT images are important for clinical decision-making in liver treatment planning and response evaluation. In this work, we report recent advances in an ongoing project *Liver Workbench* which aims to provide a suite of tools for the segmentation, quantification and modeling of various objects in CT images such as the liver, its vessels and tumors. Firstly, a liver segmentation approach is described. It registers a liver mesh model to actual image features by adopting noise-insensitive flipping-free mesh deformations. Next, a propagation learning approach is incorporated into a semi-automatic classification method for robust segmentation of liver tumors based on liver ROI obtained. Finally, an unbiased probabilistic liver atlas construction technique is adopted to embody the shape and intensity variation to constrain liver segmentation. We also report preliminary experimental results.

1   Introduction

The liver is the largest solid organ in the abdomen and is one of the most frequently involved by tumors. Liver failure due to cirrhosis and liver cancers, both primary and secondary, is prevalent in the Eastern countries and accounts for about one million deaths worldwide annually [1]. The treatment for liver tumors includes

Jiayin Zhou · Wei Xiong · Weimin Huang · Tian Qi · Zhimin · Thiha Oo
Institute for Infocomm Research, A*STAR, Singapore
e-mail: {jzhou,wxiong,wmhuang,tian,}@i2r.a-star.edu.sg
        {zwang,othiha}@i2r.a-star.edu.sg

Feng Ding
Department of Diagnostic Radiology, School of Computing,
National University of Singapore, Singapore

Sudhakar Kundapur Venkatesh
Department of Diagnostic Radiology,
National University of Singapore, Singapore

radiofrequency ablation (RFA), transarterial chemo-embolization (TACE), selective internal radiation therapy (SIRT), resection and even liver transplantation. During treatment, it is important to preserve as much liver tissue as possible for normal liver function especially in those with cirrhotic livers; however inadequate resection or treatment will be ineffective. Hence information about the volume of healthy liver parenchyma and tumor burden is important for clinical decision-making in treatment planning and response evaluation.

The efficacy of volume quantification highly depends on robust and reliable segmentation methods which are able to extract the desired organs/structures from medical images. Besides of the robustness of segmentation algorithms, segmentation results can be improved by incorporating a probabilistic atlas, which encodes probabilities of anatomic variability while retaining both spatial and densitometric variances. In addition, probabilistic atlases provide valuable information for medical image interpretation, registration as well as group variation studies.

A project named *Liver Workbench* is being conducted by our institutions, with the aim to develop an image analysis platform with 3D liver objects (including the whole liver, liver tumors and vessels) segmentation, modeling, visualization and quantification toolkits for liver surgical planning and tumor treatment evaluation. By combining image database with associated ground truth created by radiologists, the Liver Workbench will also work as a performance benchmarking platform for segmentation methods. In this paper, the segmentation and modeling methods developed for the Liver Workbench are presented, including the algorithms, experiments and results.

2   System Overview

Multi-detector computer tomography (MDCT) is the imaging modality of choice for the comprehensive assessment of liver diseases, due to the fast imaging capability and the excellent spacial resolution up to 0.5 mm. However high spacial resolution will cause the decrease of signal-to-noise ratio (SNR), hence some image segmentation algorithms are not able to achieve satisfying performance under a low SNR environment. The contrast of a CT image comes from the differentiation of density, however in a lot of cases, there is no obvious difference in density between liver lesion and surrounding healthy liver parenchyma. In addition, high spacial resolution also leads to the decrease of resolution in density. These factors make the low contrast in lesion and the fuzziness of the lesion contour. The liver is the largest organ in human body, with very large variations in size, shape and hepatic vascular structure. These problems should be addressed in algorithms developed for the Liver Workbench.

This system has four modules: liver segmentation, tumor segmentation, segmentation performance validation, and the construction of probabilistic liver atlas, as shown in Fig. 1. Segmentation ground truth data are included for performance benchmarking, algorithm training and probabilistic atlas construction. The whole liver volume is segmented by a flipping-free mesh deformation model, with the aid

from the liver atlas to be a strong spatial constraint. The segmented liver volume is treated as a region of interest (ROI) and liver tumors are segmented using the support vector machines (SVMs)-based classification from the ROI. After the segmentation, results are fed into the performance validation module, where quantitative evaluation metrics are computed with the comparison to segmentation ground truth. Moreover an iterative landmark-free method based on dense volumes has been developed for the construction of a linear unbiased diffeomorphic probabilistic liver atlas from CT images. The whole system is integrated in 3D Slicer [2] which is an ITK/VTK [3, 4]-based open-source platform.
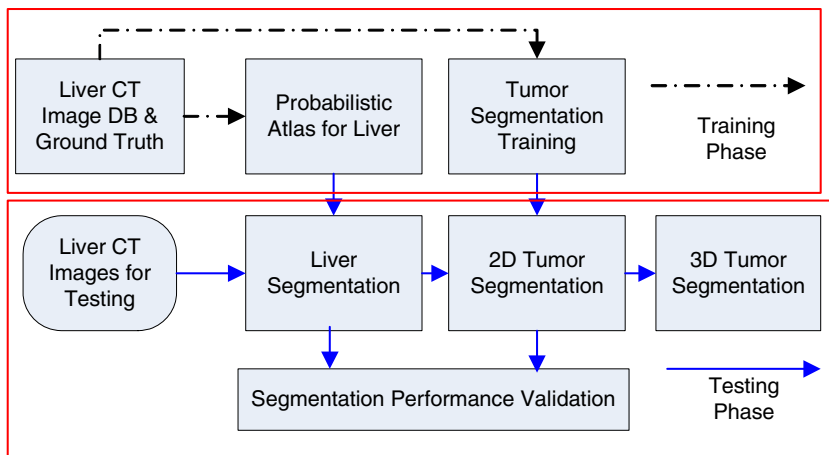

Fig. 1 System overview of the segmentation and modeling part of the Liver Workbench.

3 Segmentation of Liver and Liver Tumors

3.1 *Flipping-Free Mesh Deformation for Liver Segmentation*

Deformable models have been widely used for medical image segmentation [5]. Segmentation methods using implicit models such as the level set method and the fast marching method represent a 3D surface as an implicit function discretized into voxels, resulting in computationally expensive algorithms. The level set method can change the topology of the evolving surface to match highly complex object surface. However, it often leaks out of the object boundaries, producing undesired segmentation. In contrast, segmentation methods using explicit models represent a 3D surface as a mesh, which significantly reduce the space complexity of the algorithms. Deformation is accomplished by displacing the mesh vertices. The problem of mesh-based methods is that the displacements of vertices may cause self-intersections

of the mesh, which can be categorized as flipping or non-flipping. Flipping self-intersection occurs locally if the displacement vectors of neighboring mesh vertices cross in space. As a result, the directions of some surface normals flip after deformation. This problem cannot be solved by simply reducing the deformation step size. Non-flipping self-intersection occurs globally without flipping the surface normals but causes penetration of different parts of the mesh.

In this study, a flipping-free mesh deformation approach [6] has been developed to segment the 3D liver volume from the input CT data volume by iteratively deforming a 3D mesh model to register it to extracted image features. It searches for possible correspondence between mesh vertices and image features over long distances in iterations. The detected correspondence is refined before deformation to avoid flippings. The registration procedure of finding correspondences, flipping detection and avoidance and mesh deformations is iterated until convergence. The major components of the method are detailed as follows.

3.1.1   Image Feature Extraction

We extract the density distribution of all voxels in a volume within a mesh surface, as the liver volume may have heterogeneous voxel densities and some object boundaries are indistinct. Densities of the liver are modeled as a mixture of Gaussians $g(x) = \sum_i a_i f_i(x)$, where x is the voxel density, a_i are coefficients, such that $\sum_i a_i = 1$, and $f_i(x)$ are Gaussian distributions with parameters (μ_i, σ_i). The number of Gaussians is determined by the input images. Parameters a_i, μ_i and σ_i can be estimated by expectation maximization (EM). To smooth out noise, the anisotropic filtering is applied to the input image as a pre-processing step.

3.1.2   Finding Correspondence

A 3D cubical quadrilateral mesh M consists of a set of cubes whose sides are aligned with the axes $x-$, $y-$ and $z-$ axis of a rectangular coordinate system. It is defined by three groups of orthogonal and closed contours that are parallel to the $xy-$, $yz-$ or $zx-$ plane respectively. Each vertex u_i in M is an intersection of two contours from different groups and with exactly four connected neighboring vertices.

To find the correspondence between the mesh model M and the target T (the image feature), for each vertex u_i in M, we search along the projection line $P(u_i)$, which can be defined as the surface normal at u_i, for a possible corresponding point v_i on the surface of T. The point v_i is the intersection of $P(u_i)$ and the face of a feature voxel on the surface of T. Each v_i serves as a target location for u_i. In general, $P(u_i)$ may be defined along other appropriate directions. u_i is labeled as a solitary vertex if its corresponding point cannot be found. Hence corresponding u_j to u_i along $P(u_i)$ is to find the minimum j such that

$$\sum_{i=j}^{j+N} h(u_i) = 0, \qquad h(u_i) = \begin{cases} 0 & g(x_i) < \Gamma \\ 1 & \text{otherwise} \end{cases}, \tag{1}$$

where x_i is the density of u_i, Γ is a pre-defined threshold and $g(x_i)$ is the image feature. The vertex u_j that satisfies Eq. (1) is likely on the boundary of the target object, since $(N+1)$ consecutive voxels along $P(u_i)$ starting from u_j all have low probabilities of belonging to the foreground. In the current study, $N = 3$.

3.1.3 Flipping Detection

The flipping of a mesh cell after mesh deformation occurs at least at one of its edges. Therefore, surface flipping can be identified by detecting edge flipping. Let u_i and u_j denote two non-solitary neighbors on a closed contour, and v_i and v_i denote their respective corresponding points on the target. Then, edge flipping occurs when the orientations of the edges $u_i - u_j$ and $v_i - v_j$ differ significantly such that $(u_i - u_j) \cdot (v_i - v_j) \le \tau \|u_i - u_j\| \|v_i - v_j\|$, where $\tau \in [0, 1]$ is a predefined threshold. The vertices u_i and u_i that form a flipping edge are labeled as flipping vertices; otherwise they are non-flipping vertices. As each vertex u_i is an intersection of two orthogonal closed contours on mesh M, its flipping-free on one contour does not guarantee its flipping-free on the other contour. Therefore, each u_i will undergo the flipping detection along two closed contours when the algorithm iterates.
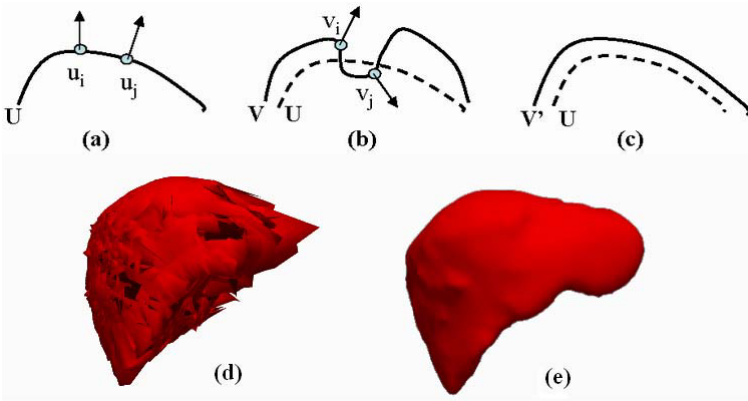

Fig. 2 (a) Two non-solitary neighbors u_i and u_j on a closed curve U; (b) After U deforms to V, u_i and u_j deform to v_i and v_j, the orientation of the edges $u_i - u_j$ and $v_i - v_j$ differ significantly; (c) Flipping avoidance by discarding the point correspondences; (d) A liver segmentation result without flipping detection and avoidance; (e) The corresponding result with flipping detection and avoidance.

3.1.4 Flipping Avoidance

We discard the point correspondences involving flipping. Let $u_i, u_{i+1}, ..., u_n$ denote a consecutive sequence of flipping vertices on a closed contour, excluding solitary vertices, such that u_{i-1} and u_{n+1} are non-flipping. The method identifies the middle flipping vertex u_m of the sequence, labels it as non-flipping, and labels the other flipping vertices as solitary, i.e., discarding their correspondences. After repeating this process for every closed contour, only non-flipping vertices have point correspondences. Thereafter, deforming the mesh according to these correspondences does not result in flipping. The procedure of flipping detection and avoidance is illustrated in Fig. 2.

3.1.5 Mesh Deformation

During mesh deformation, if non-flipping vertices are displaced to their target locations while solitary vertices remain unchanged, as shown in Fig. 3(a), the mesh may fold around solitary vertices, as shown in Fig. 3(b). In an extreme case, it results in non-flipping self-intersections, as shown in Fig. 3(c). To tackle this problem, the displacement vectors of non-flipping vertices are propagated to neighboring solitary vertices, turning them into non-flipping vertices by iterative local averaging of displacement vectors. This process is analogous to the diffusion of gradient vectors [7]. In the meanwhile, it can also smooth the variation of displacement vectors among neighboring non-flipping vertices, thus improving noise resilience. In this study, the Laplacian method [8] is adopted for mesh deformation because it is very efficient, easy to use, and easy to incorporate geometric constraints. During the deformation, non-flipping vertices are displaced towards their target locations, which are regarded as positional constraints. The other mesh vertices are displaced according to geometric constraints including the preservation of Laplacians (i.e., curvature normals) and uniform vertex distribution.

Fig. 3 Folding problem: (a) Displacing non-flipping vertices (dots) around solitary vertices (circle) may cause (b) folding of the mesh, and in an extreme case, (c) non-flipping self-intersection.

3.2 SVMs-Based Voxel Classification and Propagational Learning for Liver Tumor Segmentation

The major idea of this semi-automatic scheme is to utilize the similarities of lesion location, shape and signal intensities among the neighboring slices. Targeted tumor region is first extracted from one single 2D slice in the intermediate part of a tumor using supervised learning-based voxel classification. Then the extracted tumor contour, after some morphological operations, is mapped to its neighboring slices for automated sampling, learning and further voxel classification in the neighboring slices. This scheme is constructed on the important basis that scan is performed on multi-sliced CT and the slice thickness is no more than 3 mm such that there are only slight changes in both structural shapes and density properties among consecutive slices. The proposed segmentation scheme, illustrated in Fig. 4, has three main steps:
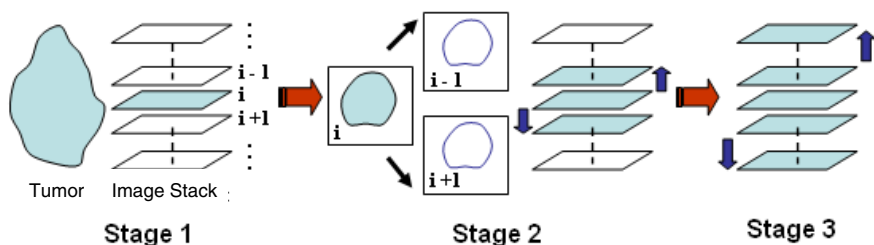

Fig. 4 Three main steps for the liver tumor segmentation scheme.

1. **Step A1** Targeted tumor region in 2D slice i at the intermediate part of the tumor is first segmented out by supervised learning-based voxel classification. In this procedure, tumor samples and non-tumor tissue (healthy liver tissue and other non-liver tissue) samples are manually selected to train a two-class SVMs-based classifier. Then the trained SVM classifier is imposed to a ROI in slice i for voxel classification such that tumor region in the ROI is extracted [9, 10].

2. **Step A2** Let C_i be the contour of extracted tumor region from slice i, C_i^D and C_i^E be the contours after morphological dilation and erosion operations are performed on C_i, respectively. As slice thickness of 1-3 mm was employed for these data, it can be assumed with a high confidence level that the targeted tumor contours and image features vary slightly among neighboring slices. Hence both C_i^D and C_i^E are projected to slices i-1 and i+1. In slices i-1 and i+1, voxels enclosed by C_i^E are used as the new tumor samples for heuristic learning to train the SVM classifier, then the updated SVMs classifier is imposed to the area enclosed by C_i^D for tumor region extraction in the two propagating slices. This step is the first propagation procedure.

3. **Step A3** Similarly the propagation procedure including contour projection, SVMs classifier training and voxel classification is further applied to upper and lower slices for tumor region extraction till all tumor-bearing slices are processed.

3.3 Implementation

In the implementation, the liver segmentation was initialized by a 3D spherical mesh totally inside the target liver (Fig. 5(a)). Voxels inside the sphere are used to build a Gaussian mixture model (GMM) of the density probability distribution of the target liver (foreground). Voxels with low probability are regarded as background feature voxels. Transitions from consecutive foreground voxels to consecutive non-liver voxels along searching directions suggest the presence of boundary points of the target organ. After liver segmentation, the segmented liver contour (Fig. 5(b)) was treated as the ROI for further tumor segmentation. During the SVMs-based voxel classification in tumor segmentation, voxel density and the median of the densities of the voxel's eight-neighbors in the same slice were used as input features. They represent the information for the voxel and its neighborhood, respectively. In addition, a Gaussian radius basis function (RBF) $K(x,y) = \exp\left(\frac{-\|x-y\|^2}{\sigma}\right)$ was adopted as the learning kernel in the SVM classifier used, where σ was set by the variations of tumor samples. In the first step of tumor segmentation, tumor samples were picked by mouse click while non-tumor samples were selected by a rectangular box (Fig. 5(c)). If the amount of non-tumor samples is more than twice of tumor samples, a random re-sampling would be performed on non-tumor samples to balance sample populations, which is important in the training of SVM classifier. After voxel classification for tumor region extraction, the whole propagation procedure including contour projection, SVM classifier training and voxel classification was further applied to upper and lower slices till all tumor-bearing slices were processed. Fig. 5(d) shows the segmented tumor.
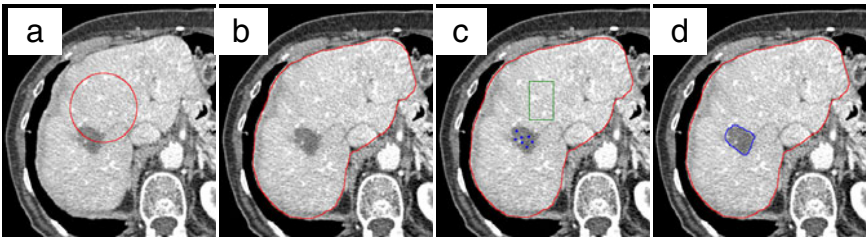

Fig. 5 (a) A 2D slice of a 3D spherical mesh for the initialization of liver segmentation; (b) the contour of the segmented liver in one slice; (c) the selected tumor samples (blue dots) and non-tumor samples (green rectangular box) and (d) the segmented liver tumor (blue contour).

3.4   Experiments and Evaluation Metrics

Twenty sets of multi-detector CT data for abdominal scan (slice thickness, 1-3 mm; image matrix, 512×512 pixels; and in-plane resolution, 0.6-0.9 mm) were used to test the algorithms. The liver boundaries were manually traced out by an experienced body radiologist. For the 20 data sets, 12 are with healthy livers (no focal lesion found) and 11 isolated focal lesions are found in the remaining 8 data sets. The outlines of the 11 tumors identified were also manually traced out by the same radiologist. They will be used as the reference standard (RS) for the comparison with the computerized segmentation results. In this study, three quantitative measures covering both volume and voxel levels were used to evaluate the algorithms' performance, with the comparison to liver and tumor reference segmentation RS [11].

1. Relative absolute volume difference (*RAVD*, %)

$$RAVD = \frac{|Vol_{seg} - Vol_{RS}|}{Vol_{RS}} \times 100\%, \tag{2}$$

   where Vol_{seg} denotes segmented liver/tumor volume, Vol_{RS} denotes the volume of RS. Note that the perfect value of 0 can also be obtained for a non-perfect segmentation, if the volume of that segmentation is equal to the volume of RS.
2. Volumetric overlap error (*VoE*, %)

$$VoE = (1 - \frac{Vol_{seg} \cap Vol_{RS}}{Vol_{seg} \cup Vol_{RS}}) \times 100\%, \tag{3}$$

   where $Vol_{seg} \cap Vol_{RS}$ is the number of voxels in the overlap or intersection of the two volumes, $Vol_{seg} \cup Vol_{RS}$ is the number of voxels in the union. This value is 0 for a perfect segmentation and the value of 100 means that there is no overlap at all between segmentation and RS.
3. Average symmetric surface distance (*ASSD*, mm)

$$ASSD = \frac{\sum_{a \in A}[\min_{b \in B}\{\mathrm{dist}(a,b)\}] + \sum_{b \in B}[\min_{a \in A}\{\mathrm{dist}(b,a)\}]}{N_A + N_B}, \tag{4}$$

   where A and B denote the surfaces of segmented and RS volumes respectively, a and b are mesh points on A and B respectively, $\mathrm{dist}(a, b)$ denotes the distance between a and b. *NA* and *NB* are the number of points on A and B. For each voxel along the border of a surface, the closest voxel along the border of another surface is determined and the distance between the two voxels is computed. All these distances are stored, for border voxels from both RS and segmentation. The average of all these distances gives the *ASSD* which tells us how much on average the two surfaces differ. This value is 0 for a perfect segmentation.

4   Probabilistic Liver Atlas Modeling

Existing liver probabilistic atlases are built based on registrations of landmarks [12, 13, 14], which are sensitive to the choice, the accurate localization and the placements of the landmarks. As the liver shape is highly variable and the texture is almost homogeneous in CT images, automatic localization of the landmarks inside the liver is difficult. Recent developments in constructing statistical atlases of shape variation, and non-rigid registration focus on building models of diffeomorphic deformation fields with a dense voxel correspondence [15, 16]. This can avoid the difficulties of sparse correspondence of landmarks. Hence we build an atlas based on all liver voxels. Registration involves wrapping moving subjects to the space of a *target* image. Bias may be introduced into the registration results towards to the chosen target image if it is arbitrarily chosen [15]. Minimizing or even removing such bias is thus highly desirable. Here we build unbiased atlases by using the classical pairwise deformation approach to construct a linear unbiased diffeomorphic probabilistic atlas of liver. For each pair of registrations, an efficient non-parametric image registration method, called diffeomorphic demons [16] is adopted. A more detailed description of this algorithm was presented in [17] and we outline it here for completeness.

The construction of a probabilistic atlas comprises two phases: find the optimal representative of group of images and statistically summarize the spatial and intensity variations. The first phase is to estimate a template image that is the best representative for the population on the infinite dimensional space of diffeomorphisms. Generally this can be considered as a registration optimization problem involving multiple images. Formally, given N image intensity images $\{x_i\}_{i=1}^{N}$ defined in a very large Euclidian vector space R^q (with q being the maximum number of voxels), we want to find $\hat{\mu}$ such that

$$\hat{\mu} = \arg\min_{x \in R^q} \sum_{i=1}^{N} d(x, x_i)^2, \tag{5}$$

where d is a distance metric. If we choose the simple Euclidean distance, then the unbiased estimation is the *algebraic* mean of the population, i.e.,

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{6}$$

Now considering the possible deformation h_i for each image x_i with respect to the target, we have, at each iteration j,

$$\mu^{(j)} = \frac{1}{N} \sum_{i=1}^{N} I_i^{(j)} = \frac{1}{N} \sum_{i=1}^{N} I_i(h_i^{(j)}(x_i)). \tag{7}$$

Here for the original training images, $j = 0$. Further, we use

$$\omega(h_i^{(j)}(x), \mu^{(j)}) \to h_i^{(j+1)}(x) \tag{8}$$

to denote the ith image x deformed by h_i for each iteration j to the space of $\mu^{(j)}$ and result in a new warped image $h_i^{(j+1)}(x)$, derived from x. The algorithm initializes by setting $j = 0$, followed by the following three steps.

1. **Step B1**: Compute $\mu^{(j)}$ using the Procrustes method, i.e., superposition of all sets of points. If $j = 0$, use Eq. (6), else use Eq. (7).
2. **Step B2**: For each image $I_i^{(j)} = I_i(h_i^{(j)}(x_i))$, find the optimal diffeomorphic deformation $\tilde{h}_i^{(j)}$ to warp $I_i^{(j)}$ to $\mu^{(j)}$: $\omega(\tilde{h}_i^{(j)}(x), \mu^{(j)}) \rightarrow \tilde{h}_i^{(j+1)}(x)$.
3. **Step B3**: Let $j = j + 1$; go to Step B2 until convergence or the maximum number of iterations is achieved.

Having found the optimal representative group mean image, we check whether there is a liver point instance at each voxel of the registered image space. If there is, the number of instances is incremented by one. After scanning all the space, the higher the number of liver instances found, the higher the probability. In this way we construct the liver probabilistic atlas.

5 Experiments and Results

5.1 Liver and Tumor Segmentation Results

The algorithm segmented out 19 liver volumes while it failed in one data set with health liver. Among the 19 segmented liver volumes, some are with minor errors at the border of the liver while some have considerable errors. Fig. 6 shows slices from one data set with good liver segmentation result that minor errors of under-segmentation occur at the marginal part. Fig. 7 shows results from another data set with moderate errors. Besides the under-segmentation errors at the marginal places with high curvatures, the algorithm could not well separate liver from the abdominal wall or the neighboring organs when they have similar density values and are in close contact to each other. Some examples of liver tumor segmentation results are also shown in Figs. 6 and 7. Overall computerized segmentation results are close to the interpretation from radiologist, but large errors occur where blurry transition exists between tumor and normal liver parenchyma. As shown in Table 1, for the 19 livers segmented, the mean $RAVD$, VOE and $ASSD$ are 7.1%, 12.3%, and 2.5 mm, respectively. For the segmentation of 11 liver tumors, as shown in Table 2, the median $RAVD$, VOE and $ASSD$ achieved are 7.3%, 18.4%, and 1.7 mm, respectively.

It is observed that the major errors in liver segmentation come from two types: The under-segmentation at some marginal part of the liver where the shape is with high curvatures and the over-segmentation that the liver contour leaks to the abdominal wall and other neighboring organs. The first type is due to the strong shape constraints from the global 3D mesh to keep the shape smoothness. Therefore it is difficult for the contour to deform to some places with high curvature, though there is strong image feature as the object boundary. On the other hand, the second type comes from the weak image feature as the object boundary. The leakage of the liver
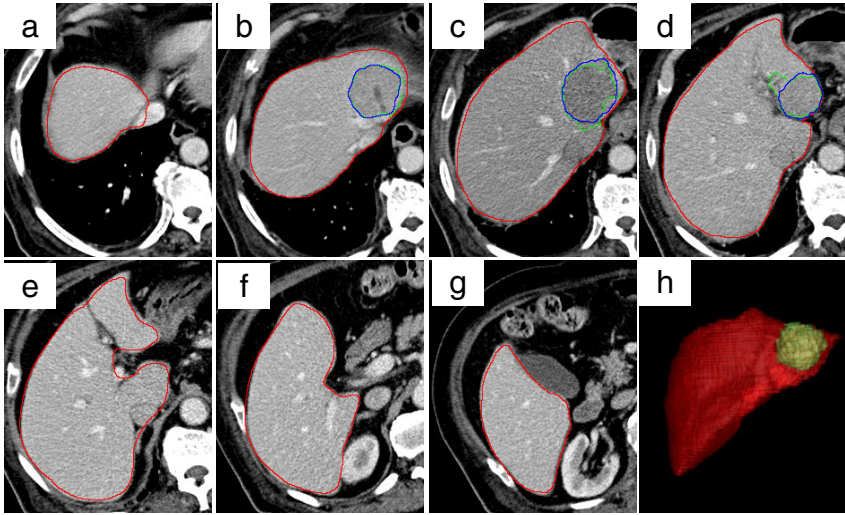
Fig. 6 (a)-(g) Slices from a well segmented liver volume (red contours), with outlines of the segmented liver tumor (green) in the left lobe and the tumor RS (blue); (h) 3D view (coronal) of the segmented liver with the tumor.
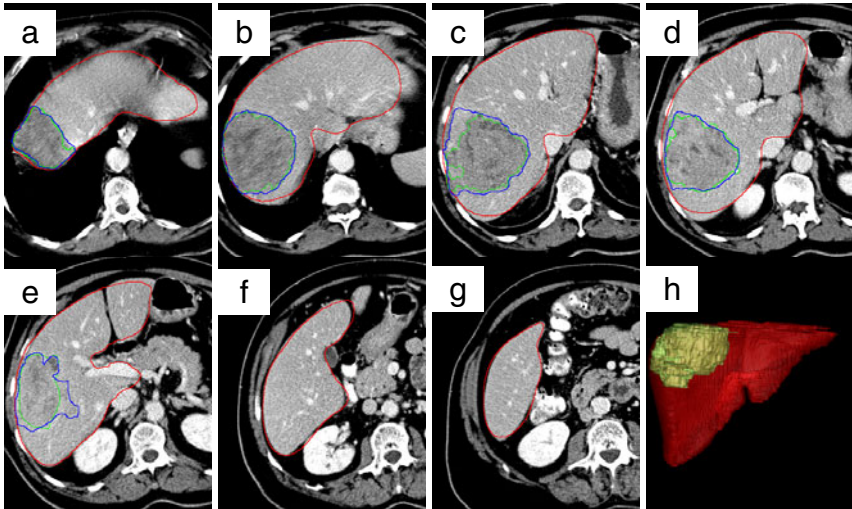

Fig. 7 (a)-(g) Slices from another segmented liver volume (red contours), with outlines of the segmented liver tumor (green) in the right lobe and the tumor RS (blue); (h) 3D view (coronal) of the segmented liver with the tumor.

Table 1 Quantitative evaluation of liver segmentation results.

	Min.	Max.	Mean	STD.	Median
RAVD (%)	0.0	30.8	7.1	8.7	3.5
VOE (%)	6.6	36.3	12.3	7.1	9.9
ASSD (mm)	1.1	10.5	2.5	2.1	1.8

Table 2 Quantitative evaluation of liver tumor segmentation results.

Tumor Subject	1	2	3	4	5	6	7	8	9	10	11	Median
RAVD (%)	2.8	19.6	8.4	6.7	14.2	8.9	6.5	5.9	7.3	5.8	16.1	7.3
VOE (%)	12.8	30.4	20.0	16.3	20.5	24.7	16.7	17.5	11.1	18.4	24.2	18.4
ASSD (mm)	1.0	2.5	1.7	1.3	2.1	2.2	0.8	1.2	0.7	1.1	2.1	1.7

contour to the abdominal wall and other neighboring organ does not cause a significant change of the property of the GMM which was previously learned. Hence it will be very interesting to study the combination of this method with a probabilistic liver atlas, which is able to give a strong shape prior to guide the mesh deformation.

5.2   *Probabilistic Liver Atlas Modeling Results*

We use $N = 25$ CT liver datasets of varying spacings and sizes. They are first normalized to give a common size of $441 \times 441 \times 48$ voxels with voxel size $1 \times 1 \times 5$ mm^3. Registration performance is measured by using the mean square metric and the mutual information between the registered image and the fixed image. The mean square metric between the intensity difference of two images A and B is given by $\mathrm{MSE}(A,B) = \frac{1}{q}\Sigma_{i=1}^{q}(a_i - b_i)^2$, where a_i, b_i is the intensity of the ith pixel of A and B, respectively, and q is the total number of pixels considered. We denote the marginal probability density functions of the intensities of A and B by p_1 and p_2, respectively, and their joint probability density function by p_{AB}. The entropy of an image is defined by $H_i = -\Sigma_k p_i(k) \log_2(p_i(k))$, $i = 1, 2$. Similarly, we can define the joint entropy $H_{1,2}$ by using p_{AB}. Now the mutual information MI can be defined by $\mathrm{MI} = H_1 + H_2 - H_{1,2}$.

We have implemented our method using liver expert-labeled CT image datasets in 10 iterations. For each cycle of iterations j, $j = 1,..,10$, we record the performance indexes (MSE and MI) between each registered image $\tilde{h}_i^{(j)}(x)$ and the mean $\mu^{(j)}$. The respective averages of the MSE and MI measures of all datasets for this iteration are also computed as performance indexes. We have used $N = 5, 10, 15, 20, 25$ datasets to construct 5 respective probabilistic atlases. Of all the datasets involved, their average of MSEs and that of MIs for $N = 10$ are denoted by MSE10 and MI10, respectively. Similarly, we can define the measures for other cases, say, MSE25 and MI25 are for $N = 25$.
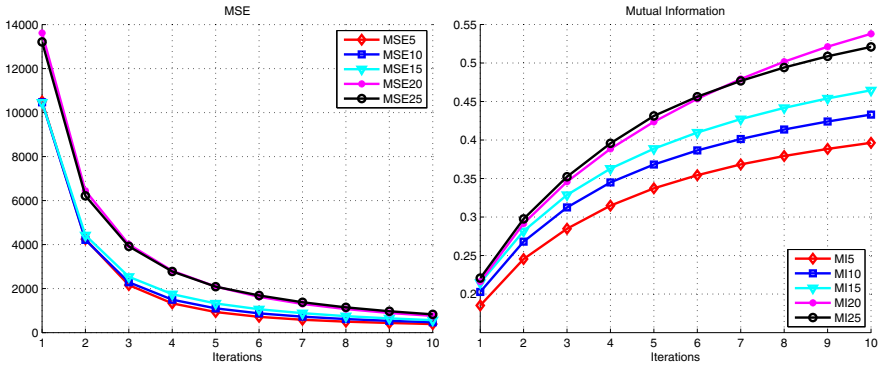
Fig. 8 Performance index curves against the number of iterations: When iterating more times MSEs decrease while MIs increase.

Fig. 8 presents these performance indexes against the number of iterations. As we iterate more times for the optimization, the MSEs decrease monotonically while MIs increase monotonically. Hence, given enough cycles, the iterations will converge. In particular, for MSE5, MSE10 and MSE15, their curves tend to be horizontal for cycles 7 to 10, i.e., already converged. Furthermore, the more datasets we use, the higher their MIs are. Hence with more and more datasets for training, we can obtain more information and more representative liver atlases.

Fig. 9 illustrates the mean images $\mu^{(j)}$, $j = 1, 5, 10$, for $N = 25$, and the corresponding probabilistic atlases in upper, middle and lower rows, respectively. In each of the figures, the left three columns are the mean images and the right three columns are their respective probabilistic atlases in order. Each column is for a
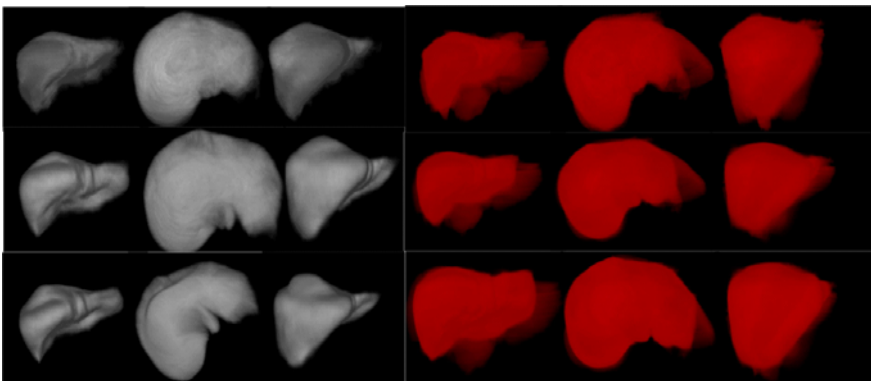

Fig. 9 The mean images (left three columns) and respective probabilistic atlases (right three columns) of the liver using 25 datasets for iteration 1 (upper row), 5 (middle row) and 10 (lower row) iterations. Three views are shown in different columns.

particular view of observations. When visualizing the atlases, we render the image with brightness proportional to the probability of each voxel at which it might be a liver point so that the higher the probability the brighter it appears. We notice that, with more iterations, the fine parts of the liver become more apparent.

6   Conclusions

We have described methods in an ongoing project *Liver Workbench* which aims at providing tools for segmentation and measuring the liver, its tumor and vessels from CT images. They are liver and liver tumor segmentation and unbiased probabilistic liver modeling. Liver segmentation is based on the iterative flipping-free mesh deformations to register the mesh model to image features in 3D volumes. In most situations, it works well, though there are errors in handling liver boundaries with high curvatures. Based on liver ROI obtained, a semi-automatic SVMs-based classification method is introduced to delineate liver tumor contours with an acceptable accuracy. A probabilistic liver atlas is built to confine the shape and intensity variation. The integrations of the techniques are still being investigated.

References

1. Teo, E.K., Fock, K.M.: Hepatocellular carcinoma: an Asian perspective. Dig. Dis. 19, 263–268 (2001)
2. http://www.slicer.org (cited March 15, 2011)
3. ITK: The Insight ToolKit, http://www.itk.org (cited March 15, 2011)
4. VTK: The Visualization ToolKit , http://www.vtk.org (cited March 15, 2011)
5. Suri, J., Farag, A.: Deformable Models II: Theory and Biomaterial Applications. Springer, New York (2007)
6. Ding, F., Yang, W., Leow, W.K., Venkatesh, S.: Segmentation of soft organs by flipping-free mesh deformation. In: Proc. IEEE Workshop Application Comput. Vis. (2009)
7. Xu, C., Prince, J.L.: Snakes, shapes, and gradient vector flow. IEEE Trans. Image Process. 7, 359–369 (1998)
8. Sorkine, O., Lipman, Y., Cohen-Or, D., Alexa, M., Rössl, C., Seidel, H.P.: Laplacian surface editing. In: Proc. 2004 Eurographics, pp. 179–188 (2004)
9. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. University of Cambridge, Cambridge (2000)
10. Zhou, J., Chan, K.L., Xu, P., Chong, V.F.: Nasopharyngeal carcinoma lesion segmentation from MR images by support vector machine. In: Proc. 3rd IEEE Int. Symp. Biomed. Imaging, pp. 1364–1367 (2006)
11. Gerig, G., Jomier, M., Chakos, M.: Valmet: A new validation tool for assessing and improving 3D object segmentation. In: Niessen, W.J., Viergever, M.A. (eds.) MICCAI 2001. LNCS, vol. 2208, p. 516. Springer, Heidelberg (2001)
12. Park, H., Bland, P.H., Meyer, C.R.: Construction of an abdominal probabilistic atlas and its application to segmentation. IEEE Trans. Med. Imaging 22, 483–492 (2003)
13. Park, H., Bland, P.H., Hero III, A.O., Meyer, C.R.: Least biased target selection in probabilistic atlas construction. In: Duncan, J.S., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3750, pp. 419–426. Springer, Heidelberg (2005)

14. Zhou, X., Kitagawa, T., Okuo, K., et al.: Construction of a probabilistic atlas for automated liver segmentation in non-contrast torso CT images. Int. Congress Series, vol. 1281, pp. 1169–1174 (2005)
15. Joshi, S., Davis, B., Jomier, M., Gerig, G.: Unbiased diffeomorphic atlas construction for computational anatomy. NeuroImage 23 (suppl.), S151–S160 (2004)
16. Vercauteren, T., Pennec, X., Perchant, A., Ayache, N.: Non-parametric diffeomorphic image registration with the demons algorithm. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part II. LNCS, vol. 4792, pp. 319–326. Springer, Heidelberg (2007)
17. Xiong, W., Ong, S.H., Xu, G., et al.: Construction of an unbiased diffeomorphic probabilistic liver atlas based on CT images. In: Proc. Int. Conf. Image Process., pp. 1773–1776 (2009)