

# Fast and Robust Semi-automatic Liver Segmentation with Haptic Interaction

Erik Vidholm<sup>1</sup>, Sven Nilsson<sup>2</sup>, and Ingela Nyström<sup>1</sup>

<sup>1</sup> Centre for Image Analysis, Uppsala University, Sweden  
erik@cb.uu.se

<sup>2</sup> Dept. of Radiology and Clinical Immunology, Uppsala University Hospital, Sweden

**Abstract.** We present a method for semi-automatic segmentation of the liver from CT scans. True 3D interaction with haptic feedback is used to facilitate initialization, i.e., seeding of a fast marching algorithm. Four users initialized 52 datasets and the mean interaction time was 40 seconds. The segmentation accuracy was verified by a radiologist. Volume measurements and segmentation precision show that the method has a high reproducibility.

## 1 Introduction

One of the most important steps in medical image analysis is segmentation, i.e., the process of classifying data elements as object or background. Segmentation is needed in diagnostics, therapy monitoring, surgery planning, and several other medical applications. To manually segment the structures of interest in medical datasets is a very tedious and error-prone procedure, while fully automatic segmentation is, despite decades of research, still an unsolved problem. Therefore, many methods are semi-automatic, i.e., the segmentation algorithm is provided with high-level knowledge from the user [1]. A successful semi-automatic method takes advantage of the user's ability to recognize objects and the ability of the computer to delineate objects. A common recognition task in semi-automatic segmentation is initialization by placement of seed points inside the object of interest. The interactive part is highly dependent on the user interface. Interfaces that rely on two-dimensional (2D) interaction have many drawbacks when the data is three-dimensional (3D) since it is not straight-forward how to map 2D interaction into 3D space. It has been shown that by using true 3D interaction with haptic feedback, more efficient semi-automatic methods can be obtained [2].

Liver segmentation is of importance in hepatic surgery planning, where it is a first step in the process of finding vessels and tumours, and the classification of liver segments [3,4]. Liver segmentation may also be useful for monitoring patients with liver metastases, where disease progress is correlated to enlargement of the liver [5, p. 580]. Low contrast between organs and the high shape variability of the liver make automatic segmentation a hard task. In [4], a reference 3D liver model is deformed to fit the liver contour in the image. The method performs well, except for atypical livers where manual interaction is needed. Common for many semi-automatic methods is their slice-based  $2\frac{1}{2}$ D nature [6].

In this work, we demonstrate how a haptic user interface can be used to facilitate initialization of a fast marching algorithm [7] in 3D in order to segment the liver from CT images.

## 2 Fast Marching Segmentation

Fast marching methods [7] are numerical schemes for solving the *Eikonal* equation:

$$|\nabla u| = C, \quad (1)$$

where  $u$  is time of arrival and  $C$  is a cost (“slowness”) function. The common use of fast marching in image segmentation is to design a proper cost image  $C$ , provide a set of seed points  $\mathcal{P}$  with arrival time equal to zero, and then propagate a front from these points until a certain arrival time is reached, i.e., solving Equation (1).

The classical gradient approximations, e.g., centered differences, are not well-suited for discretization of Equation (1). In [7], it is shown how the use of *upwind schemes* make the solution stable. We use the Godunov scheme

$$|\nabla u| \approx \left[ \begin{array}{l} \max(D_{ijk}^{-x}u, -D_{ijk}^{+x}u, 0)^2 + \\ \max(D_{ijk}^{-y}u, -D_{ijk}^{+y}u, 0)^2 + \\ \max(D_{ijk}^{-z}u, -D_{ijk}^{+z}u, 0)^2 \end{array} \right]^{1/2} = C_{ijk}, \quad (2)$$

where  $i, j, k$  denotes the indices in the discretization and  $D_{ijk}^{\pm x}$  are standard forward and backward difference operators. To solve Equation (2) for  $u$ , we use the algorithm in [8].

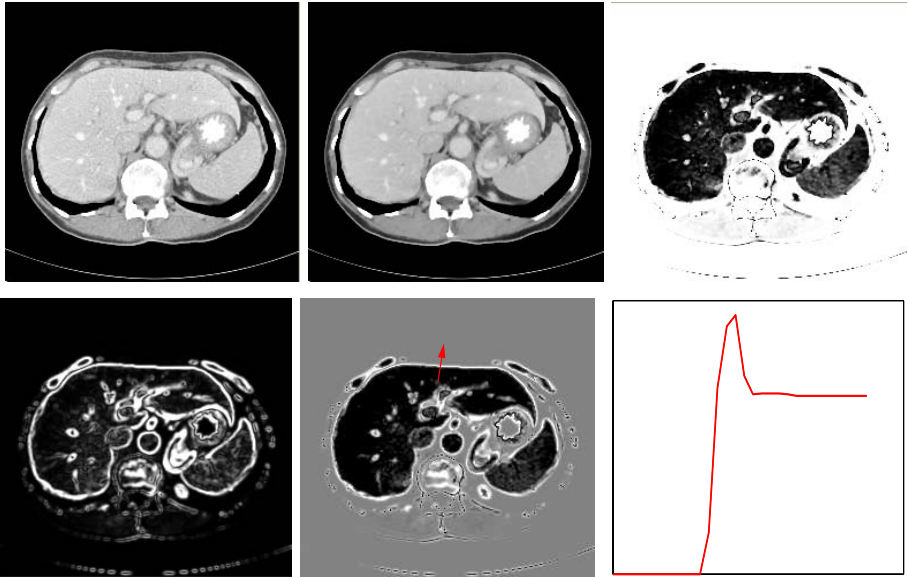
The central idea in fast marching is the observation that information propagates from smaller values of  $u$  to larger values. Using this property, the algorithm builds the solution outwards from the boundary condition. The boundary condition is expressed at the set of seed points:  $u(\mathbf{p}) = 0$ ,  $\mathbf{p} \in \mathcal{P}$ . The algorithm is accelerated by limiting the computational domain to a *narrow band* in the proximity of the front. This narrow band is stored in a fast priority queue based on a minimum heap data structure.

The cost image  $C$  should be designed to achieve low costs in homogeneous parts and high costs at edges and is typically based on the image gradient magnitude. We compute our cost image in four steps. First, we suppress noise in the original image  $I(\mathbf{x})$  by using bilateral filtering [9]. This is an edge preserving smoothing filter that combines domain and range filtering, i.e., it takes into account both spatial closeness and intensity similarity of voxels. We use a filter kernel composed of two Gaussians, one for the domain ( $\sigma_d$ ) and one for the range ( $\sigma_r$ ). The filtered image is computed as

$$I_{BF}(\mathbf{x}) = h^{-1}(\mathbf{x}) \int_{\Omega} I(\boldsymbol{\xi}) e^{-\frac{1}{2}\|\boldsymbol{\xi}-\mathbf{x}\|^2/\sigma_d^2} e^{-\frac{1}{2}(I(\boldsymbol{\xi})-I(\mathbf{x}))^2/\sigma_r^2} d\boldsymbol{\xi},$$

where  $h^{-1}(\mathbf{x})$  normalizes the kernel. As the next step, we apply a voxel-wise Gaussian filter with mean intensity  $\mu_G$  and standard deviation  $\sigma_G$ :

$$I_G(\mathbf{x}) = 1 - e^{-\frac{1}{2}(I_{BF}(\mathbf{x})-\mu_G)^2/\sigma_G^2}.$$



**Fig. 1.** The different steps in generating the cost image. Here, a 2D slice is shown, whereas all computations are in 3D. From upper left: original CT image, result of bilateral filtering, result of voxel-wise Gaussian filter, gradient magnitude, the resulting cost image ( $\alpha = 0.3$ ), and the intensity profile along the red line in the cost image.

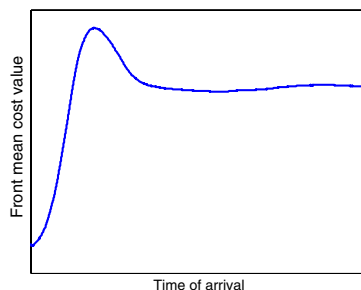
This gives low values for voxels with intensity close to  $\mu_G$ . By weighting  $I_G$  with its gradient magnitude, we obtain our final cost function:

$$C(\mathbf{x}) = \alpha I_G(\mathbf{x}) + (1 - \alpha) |\nabla I_G(\mathbf{x})|, \quad \alpha \in [0, 1].$$

Figure 1 illustrates the steps in the cost image generation and a typical intensity profile of the liver border in the cost image. We have a low cost inside the object, high cost at the edges, and a middle-valued cost outside the object. By this design of  $C$ , we aim to prevent the common problem of leakage. Once we have our cost image, we can specify seed points and run the fast marching algorithm. The problem that remains is to find at which arrival time we put a threshold to obtain our final segmentation. We use a technique similar to the energy measure suggested in [10]. The average cost value of the voxels belonging to the front is used to determine a suitable arrival time threshold. The average cost at time  $u$  is

$$\bar{C}(u) = \frac{1}{N_{Front}} \sum_{\mathbf{x} \in Front} C(\mathbf{x}),$$

where  $N_{Front}$  is the current number of voxels in the narrow band. In the beginning of propagation, the average cost is typically close to zero (assuming that the initialization is inside the object). During propagation, the average cost will increase as more and more voxels approach the object boundary. Eventually,



**Fig. 2.** The average cost of the front voxels at each time instance. The time of arrival at the maximum peak is used as a threshold to obtain the segmentation.

when the front penetrates the object boundary the average cost will decrease again. If we plot the average cost as a function of time, we typically obtain the graph in Figure 2. Note the similarity between the shape of this graph and the cost image edge profile in Figure 1. We locate the maximum peak in the graph and use the corresponding time of arrival as threshold.

### 3 Initialization with Haptic Interaction

The initialization is important for the fast marching algorithm. Most important is that the seeds must be placed inside the object. It is also desired to have them placed centrally in order to reduce the risk of leakage. To represent the shape extent, the seeds should also be placed in protrusions, see Figure 3 (left). To facilitate the initialization, we use a 3D interface with haptic feedback to draw regions of seed points. Our interface consists of a Reachin display [11] that combines a PHANToM desktop haptic device with supporting systems. See Figure 3 (right). The application is developed in the Reachin API 3.2 which is a scene-graph API for haptic and graphic visualization written in C++.

The basic view in the program is an orthogonal tri-planar construction of the dataset. The haptic interaction consists of a spring force that guides the user to hold the cursor in the closest plane. The interface consists of only the most basic functions that are needed for this specific application:

**Draw.** Draw spherical seed regions of radius 5 voxels at the cursor position

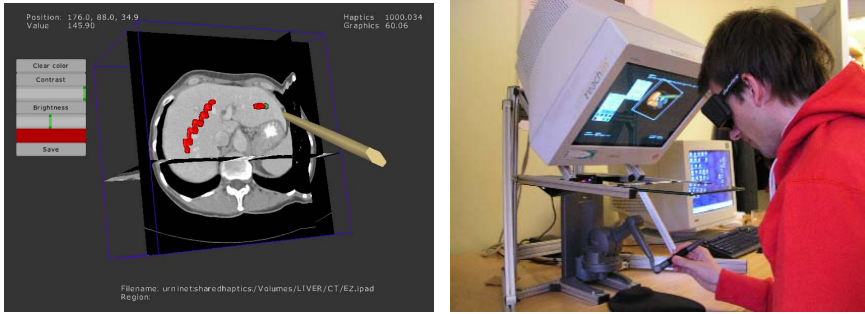
**Clear.** Clear the current seed region and start over

**Browse.** Translate the three planes according to the cursor position

**Contrast/Brightness.** Adjust the graylevel display

**Save.** Save the current seed points and exit

The idea is to use the haptic guidance to put the cursor in one of the planes and then translate this plane through the volume while drawing true 3D seed regions inside the liver. The co-located haptics and graphics guide the user in positioning numerous representative seed-points in a very fast manner.



**Fig. 3.** Left: A screenshot from the application where seed regions are being drawn centrally and representative for the shape extent. Right: A user working with the haptic display. The PHANToM device is positioned beneath a semi-transparent mirror in order to obtain co-location of graphics and haptics.

## 4 Experiment and Results

We used abdominal contrast enhanced venous phase CT images from 26 patients with either carcinoid or endocrine pancreas tumour<sup>1</sup>. Each patient underwent two CT examinations with time interval 3–17 months. The 52 images were acquired with a Siemens Sensation 16 CT scanner. The image dimensions are  $512 \times 512 \times N_z$ , where  $N_z$  ranges from 116 to 194. The resolution in  $xy$  is between 0.54–0.77 mm and the slice thickness is between 2.5–3.0 mm.

Four users performed seeding of the 52 datasets using the haptic interface. One user was an experienced radiologist and the other three users were medical image analysts. The users performed the task independent of each other. Before the real task started, there was a training session on a randomly selected dataset. The ordering of the anonymized datasets were random and the same for all users. All users completed the 52 seeding tasks in less than one hour. The interaction times and the number of seeds are summarized in Table 1.

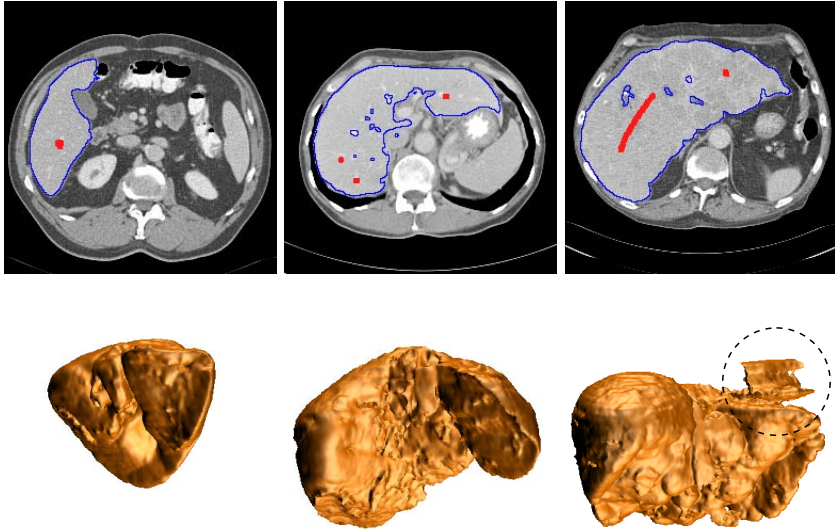
Given the regions of seed points, we computed the mean and variance of seed voxel values,  $\mu_{\mathcal{P}}$  and  $\sigma_{\mathcal{P}}$ . The cost images were then generated with the parameters  $\sigma_d = 1.0$  mm,  $\sigma_r = \sigma_G = 2\sigma_{\mathcal{P}}$ ,  $\mu_G = \mu_{\mathcal{P}}$ , and  $\alpha = 0.3$ . On a standard PC (2.4 GHz, 1 MB RAM), the computation of the cost images was about one minute per image and for the fast marching segmentation about 20 seconds per image and user.

The segmentation results were visually inspected by the radiologist and the majority were considered successful. In a few cases, there were problems with leakage due to low contrast, especially at the heart and the stomach. We found that these problems were easily repaired with manual interaction when necessary. Figure 4 shows slices with seed regions and resulting liver contour and surface renderings of three segmentation results. Note that the leakage problem is emphasized. To obtain a measure of the between-user reproducibility of

<sup>1</sup> Dr Hans Frimmel is acknowledged for providing the datasets.

**Table 1.** Interaction times and number of seeds for all users

(a) Interaction times (s)					(b) Number of seed voxels				
User	Min	Max	Mean	Stdd.	User	Min	Max	Mean	Stdd.
U1	22	173	42	24	U1	2346	8425	4453	1647
U2	20	146	41	26	U2	815	2752	1550	427
U3	24	110	53	19	U3	635	1833	1001	250
U4	13	51	26	9	U4	1434	5975	4047	1049
Total	13	173	40	23	Total	635	8425	2763	1811



**Fig. 4.** Top row: slices with the contour of the segmentation result (blue) and the seed regions (red). Bottom row: surface renderings of the segmentation results in the top row. To the bottom right the problem of leakage at the heart is encircled.

the method, we computed the correlation coefficient for liver volume estimates between the users, see Table 2. The mean liver volume for our datasets, taken as the average of the four segmentations, was  $1731 \text{ cm}^3$  with standard deviation  $669 \text{ cm}^3$ . Comparing volume values alone is not sufficient to determine the precision of the method. As suggested in [12], a measure of the inter-operator precision  $\eta_{ij}$  for two segmentation results  $\mathcal{S}_i$  and  $\mathcal{S}_j$  is the fraction of intersection and union:

$$\eta_{ij} = \frac{\mathcal{S}_i \cap \mathcal{S}_j}{\mathcal{S}_i \cup \mathcal{S}_j}.$$

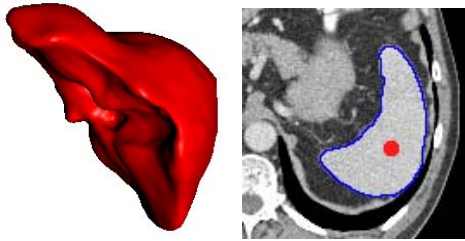
We have computed  $\eta_{ij}$  and the corresponding coefficient of variation (CV) for all possible pairs in our experiment, see Table 3. We also computed the precision for all four users  $\bigcap_{k=1}^4 \mathcal{S}_k / \bigcup_{k=1}^4 \mathcal{S}_k$  and obtained the mean value 0.944 with corresponding CV 5.48%.

**Table 2.** Correlation coefficients for estimated liver volumes between the four users and a 95% confidence interval

	U1	U2	U3	U4
U1	◇ 0.998(0.996–0.999)		0.979(0.963–0.988)	0.987(0.977–0.992)
U2	◇	◇	0.983(0.970–0.990)	0.987(0.978–0.993)
U3	◇	◇	◇	0.994(0.989–0.996)
U4	◇	◇	◇	◇

**Table 3.** Inter-user precision and coefficient of variation (CV) in % for the four users. The precision is averaged between the users for all possible pairs of segmentations and the CV is the corresponding ratio of standard deviation and mean.

	U1	U2	U3	U4
U1	◇ 0.972(2.01%)		0.960(5.36%)	0.969(3.67%)
U2	◇	◇	0.967(4.91%)	0.973(3.59%)
U3	◇	◇	◇	0.973(3.80%)
U4	◇	◇	◇	◇

**Fig. 5.** Left: A deformable surface model fitted to one of our segmentation results. Right: A slice of a spleen segmented with our method showing contour and seed region.

## 5 Conclusions and Future Work

We have demonstrated the use of a haptic 3D interface for fast interactive segmentation by using the fast marching algorithm. The mean interaction time required for the four users was 40 seconds which enables practical use of the method. Comparisons of interaction times with and without haptic feedback will be undertaken in order to quantify the relevance of haptics. Inter-user precision measures show that the method is robust despite the high shape variability of the liver and the users' very different approaches to place seed regions. According to the radiologist, the segmentation results are sufficiently accurate for liver volume quantification. In future work, the segmentation results will be verified with comparison to manually segmented ground-truth data.

We believe that the presented method can be used as a first step on the way to accurate delineation of the liver with methods that involve shape constraints, e.g., deformable surfaces or level-set methods. In Figure 5 (left), we show a

surface model that has been deformed under a distance field generated from one of our segmentations. We have also tried the method on other organs with promising results. Figure 5 (right) shows a segmentation of the spleen.

## References

1. Olabarriaga, S.D., Smeulders, A.W.M.: Interaction in the segmentation of medical images: A survey. *Medical Image Analysis* **5**(2) (2001) 127–142
2. Harders, M., Székely, G.: Enhancing human-computer interaction in medical segmentation. *Proceedings of the IEEE* **9**(91) (2003) 1430–1442
3. Meinzer, H.P., Thorn, M., Cárdenas, C.E.: Computerized planning of liver surgery—an overview. *Computers and Graphics* **26** (2002) 569–576
4. Soler, L., Delingette, H., Malandain, G., et al.: Fully automatic anatomical, pathological, and functional segmentation from CT scans for hepatic surgery. *Computer Aided Surgery* **6**(3) (2001) 131–142
5. Fauci, A.S., Braunwald, E., Isselbacher, K.J., Hauser, D.L., Kasper, D.L., Wilson, J.D., Martin, J.B., Longo, D.L., eds.: *Harrison's Principles of Internal Medicine*. 12th edn. McGraw-Hill, New York (1998)
6. Schenk, A., Prause, G., Peitgen, H.O.: Efficient semi-automatic segmentation of 3D objects in medical images. In: *Proc. MICCAI'00. LNCS 1935* (2000) 186–195
7. Sethian, J.A.: *Level set methods and fast marching methods*. Cambridge University Press (1999)
8. Kimmel, R., Sethian, J.A.: Optimal algorithm for shape from shading and path planning. *Journal of Mathematical Imaging and Vision* **14**(3) (2001) 237–244
9. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *Proc. 6th Int. Conf. on Computer Vision*. (1998) 839–846
10. Yan, J., Zhuang, T.: Applying improved fast marching method to echocardial boundary detection in echocardiographic images. *Pattern Recognition Letters* **24**(15) (2003) 2777–2784
11. Thurfjell, L., McLaughlin, J., et al.: Haptic interaction with virtual objects: The technology and some applications. *Industrial Robot* **29**(3) (2002) 210–215
12. Udupa, J.K., Leblanc, V.R., Schmidt, H., et al.: A methodology for evaluating image segmentation algorithms. In *Sonka, M., Fitzpatrick, J.M., eds.: Proceedings of SPIE Medical Imaging 2002, SPIE* (2002) 266–277