Semi-automatic Segmentation of the Liver and its Evaluation on the MICCAI 2007 Grand Challenge Data Set

Benoit M. Dawant[1], Rui Li[1], Brian Lennon[2], and Senhu Li[2]

[1] Vanderbilt University, Department of Electrical Engineering and Computer Science, Box 1662 Station B, Nashville, TN 37235
[2] Pathfinder Therapeutics, Inc., Nashville, TN
{Benoit.Dawant,Rui.Li}@vanderbilt.edu
{btl,sli}@2pti.com

Abstract. In this paper, we evaluate a semi-automatic liver segmentation method based on a level-set approach and a dynamically adapted speed function. The approach also relies on a-priori anatomic information to reduce leakage at the liver-rib interface. The numerical algorithms have been integrated into a complete system that permits loading DICOM images, segmenting these images, and visualizing the liver surface. Although the normal operation of the system assumes manual contour corrections, such corrections were not allowed for this evaluation. Results show that except for cases with large tumors that extend to the liver surface, the method we propose is comparable to a human rater.

1. Introduction

The method we have used to segment the images provided by the organizers of the challenge is a semi-automatic approach adapted from [1,2] and [3], which has been integrated into a complete system that permits loading the images, displaying them in orthogonal views, delineating initial contours on any of the orthogonal views, segmenting the liver based on these contours, editing computed contours, displaying the segmented liver, and saving segmented surfaces. Figure 1 shows a snapshot of the interface of the system we have developed. In normal operation, the system is used as follows. A contour is delineated manually on one of the slices. This contour is then automatically deformed and the liver is segmented on that slice. If needed, the user can correct the final contours using a number of editing tools that have been incorporated into the environment. This process typically starts in the middle of the liver. A skeleton of the final contour is computed and used to initialize contours on slices above or below the current one and the process is repeated until all the slices are segmented. To speed up the process, contours are not computed on every slice. The contours for the other slices are obtained by interpolation. Because of the format of the challenge, the normal segmentation procedure has been modified and manual editing of the contours has not been allowed. Instead, contours were manually delineated on a number of slices (this number varies across volumes and depends on factors such as

T. Heimann, M. Styner, B. van Ginneken (Eds.): 3D Segmentation in The Clinic: A Grand Challenge, pp. 215-221, 2007.

liver shape and number of slices). The liver was segmented on the slices on which contours were drawn. These contours were then interpolated on the other slices. No manual editing of these contours was allowed.

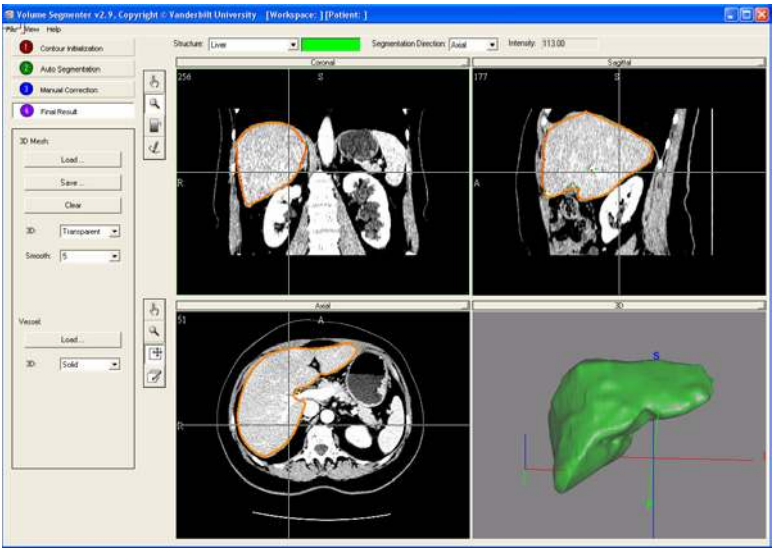

Figure 1: Interface of the liver segmentation system we have developed

2 Method

2.1 Contour Evolution Algorithm

2.1.2 The accumulative speed function

The approach we have used falls into the category of geometric deformable models and we rely on a level-set approach to drive the contour evolution. In this approach, the 2D contour is the zero-level set of a 2D embedding function ϕ, which obeys the following equation:

$$\phi_t + F \mid \nabla\phi \mid = 0 \,, \tag{1}$$

in which F is the speed function, i.e., the function that specifies the speed at which the contour evolves along its normal direction. In traditional speed functions, the terms related to the underlying image are static, i.e., they do not change as time progresses

or as a function of the path followed by the front and a typical expression [4] for the speed term is

$$F = g_1(F_0 - \varepsilon\kappa), \text{ with } g_1 = e^{-\alpha|\nabla G_\sigma * I|},$$

and κ the curvature of the front. To stop the front propagation at weak boundaries, these speed functions have to be designed to decelerate the fronts very quickly. This necessitates the choice of large values for the weights assigned to the gradient terms with the possible drawback of stopping the front at spurious edges. The approach we use is to progressively slow the front down as it passes over the boundary points. In essence, we build memory into the process. If the front passes over one boundary point it slows down some. If it passes over two boundary points in sequence, it slows down more, etc. The rate at which the front slows down as it passes over boundary points can be adjusted and it dictates the type of boundary (i.e., weak or strong, narrow or large) that stops the front. Because the actual speed is a function of the path of the contour, we have called this algorithm the accumulative speed function. The way by which this is implemented is as follows:

1. Define a standard image-dependent static multiplicative term g_{bas}

2. Define g_0 as the actual multiplicative term to be used in the speed function and let $g_0 = g_{bas}$.

3. At each iteration, compute $\{(x, y, z) \mid \Phi(x, y, z) = 0\}$, the zero level set of the embedding function. For every point in this set, retrieve $g_{bas}(x, y, z)$ and extend it to the narrow band in which the embedding function is computed (the notion of extension is defined later). This create an additional multiplicative term which we call g_{ext}

4. Let $g_0 = g_0 . g_{ext}$

5. Use g_0 as the multiplicative term in the speed function that governs the propagation of the front. For example, to include curvature in the final speed function, define it as $F = g_0(1 - \varepsilon\kappa)$

6. Compute the value of the embedding function Φ at t+1, using (1)

7. After a predefined number of iterations or if the zero level set does not move, stop. Otherwise, go to step 3

Step 3 in the algorithm requires the computation of an extended multiplicative term. Here, the notion of extension is the same as the one proposed by Adalsteinsson and Sethian [5] for velocities defined on zero level sets only. Loosely speaking, for

voxels close to the zero level set, we set $g_{ext}(x,y,z) = g_{bas}(c(x,y,z))$ in which $c(x,y,z)$ is the closest point in the zero level set. Following Adalsteinsson's approach, this can be achieved by computing an extended multiplicative term that satisfies the following equation:

$$\nabla g_{ext} \cdot \nabla \theta = 0$$

More details on this approach and comparisons between a traditional speed function and the one we use can be found in [1].

2.1.2 Definition of the static speed function

Our approach still requires the definition of an underlying static multiplicative term. Following the approach proposed by Cao [3] the evolution equation we use is

$$\phi_t = Sign(g_{bas}) \cdot ((|\, g_0 \,| - \varepsilon k) \cdot |\, \nabla \phi \,|) \tag{2}$$

In which $Sign(g_{bas})$ is the sign of the base (temporally invariant) speed function and g_0 is the value of the accumulative speed function. g_{bas} includes both an intensity and a gradient term and is defined as follows:

$$g_{bas} = g_{grad} \cdot g_{gray} \tag{3}$$

in which $g_{grad} = e^{-\alpha|\nabla G_\sigma * I|}$ and $|\nabla G_\sigma * I|$ is the magnitude of the intensity gradient of the image after convolution with a Gaussian smoothing filter with standard deviation σ, and

$$g_{gray} = \begin{cases} 2(e^{-\gamma|I - I_{\min}|}) - 1 & I \le I_{\min} \\ 1 & I_{\min} < I \le I_{\max} \\ 2(e^{-\gamma|I - I_{\max}|}) - 1 & I > I_{\max} \end{cases} \tag{4}$$

The term $Sign(g_{bas})$ is required because of the propagation scheme included in the accumulative speed function algorithm. Without this term, one would propagate a negative term when the base speed function is negative. This would lead to a term g_0 whose sign would alternate from iteration to iteration.

The algorithm we propose requires six parameters: I_{min}, I_{max}, α (the coefficient in the gradient exponential term), γ (the coefficient in the intensity exponential term),

and ε (the multiplicative term in front of the curvature). The sixth parameter is the width of the narrow band. These parameters are non-linearly related, which makes their manual selection non-intuitive. To address this problem, we have devised schemes by which the first five coefficients can be automatically estimated based on ROIs drawn on the structures of interest in the images [3]. The width of the narrow-band is a user-selected parameter but the performance of the algorithm is not sensitive to this choice.

2.2 A priori anatomic information and its use in the design of the speed function.

One area in which segmentation is particularly difficult in liver CT images is the interface between the liver and the muscles between the ribs when the liver is in close proximity of the muscles. The approach we propose to address this problem is to constrain the propagation of the contour using a-priori anatomic information. In every slice, the skin and the ribs are automatically segmented. This is done through a sequence of operations including thresholding, morphology, and region labeling. The distance between the side of the ribs toward the chest cavity and the skin is computed and used as the distance between the skin and the liver. This distance is interpolated between rib segments to constrain the propagation of the liver. This is done by adding an additional exponential term in the speed function.

3. Results

The method described above has been applied to all the test volumes. As discussed in the introduction, the software has not been used the way it is intended to be used to conform with the rules of the challenge. For each of the volumes, the following steps were followed: (1) the image volume was loaded, (2) the ribs constrain was automatically computed, (3) initial contours were drawn on selected slices, (4) the initial contours were automatically deformed , (5) contours on the other slices were interpolated from the existing contours. These contours were saved and sent to the organizers of the challenge. Total segmentation time averages 20 minutes on a machine with a Pentium D, 3.2GHz processor and 2GB of memory (about 10 minutes to delineate contours on 20 to 30 slices, and 10 minutes to compute the final contours and perform the interpolation). Figure 2 compares our contours to the manual ones for the three cases selected by the organizers. Table 1 lists the results sent back to us by the organizers.

The strengths and weaknesses of the method we propose are apparent from the figure and the table. Cases 3 and 4, which are shown in Figure 2, contain very large tumors that extend to the surface of the liver; the other cases do not have this characteristic. In our method, the values of the parameters used by the algorithm are automatically estimated from the original contours. We typically draw these original contours over normal regions. If we did also include abnormal regions, the variance in intensity values would be such that the estimated parameters would allow the contours to leak in adjacent structures, which may have intensity values in the same range as the abnormal tissue region. If the tumor does not extend to the surface of the liver, there is no major problem. The contours evolve around the tumor and stop at the edge of the liver; this is the case for dataset 10. The figure also illustrates the efficacy of our rib

constraint. Very little leakage occurs at the interface between the rib cage muscles and the liver. Without cases 3 and 4, the average total score we have obtained is 71.9, which is close to the score obtained by the human rater. Another observation is that our method underestimates the total volume of the liver. Because our current objective is to produce liver surfaces to be used for surface-based registration in an image-guided surgical system and because we are mainly interested in the outside surface, which is smooth, we smooth the surface we compute. This helps with the convergence of the registration algorithm but may reduce accuracy in certain areas of the liver.

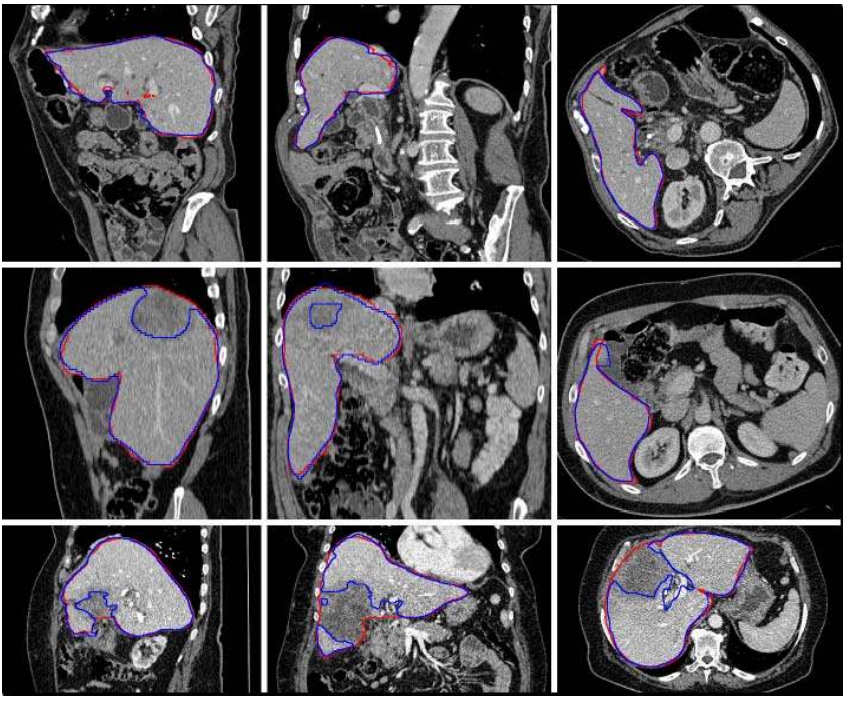

Figure 2: From left to right, a sagittal, coronal and transversal slice from a relatively easy case (1, top), an average case (4, middle), and a relatively difficult case (3, bottom). The outline of the reference standard segmentation is in red, the outline of the segmentation of the method described in this paper is in blue. Slices are displayed with a window of 400 and a level of 70

4. DISCUSSION

A relatively easy solution to the difficulty we have encountered with this data set is to define two structures: one for the normal liver region, one for the abnormal region. Initial contours can be defined for both these structures, contours can then evolve independently, or even be coupled, and the final regions merged. At the time of writing,

this approach has not been implemented but it will be at the time of the workshop. An alternative is to simply edit the contours that are incorrect. For instance, correction of the contour shown on the left panel of the middle row would only require a few mouse clicks. As usual tradeoffs between automaticity and practicality will have to be assessed to determine which approach leads to a system of clinical value. Results we have obtained suggests that, when using the editing tools implemented in our system, end users will be able to generate segmentation accurate enough fast enough to make this system useable clinically. Clinical evaluation of this system as the pre-operative component of an image-guided surgical system is currently on-going.

Dataset	Overlap Error		Volume Diff.		Avg. Dist.		RMS Dist.		Max. Dist.		Total
	[%]	Score	[%]	Score	[mm]	Score	[mm]	Score	[mm]	Score	Score
1	7.3	71	0.2	99	1.1	73	1.8	76	13.6	82	80
2	7.3	72	-1.5	92	1.0	75	1.6	77	19.4	74	78
3	16.4	36	-13.4	29	3.1	22	7.3	0	51.6	32	24
4	16.0	38	-12.4	34	2.9	26	6.7	6	45.1	41	29
5	8.4	67	-5.1	73	1.5	62	2.7	62	27.9	63	66
6	9.1	65	-5.6	70	1.4	65	2.6	64	21.0	72	67
7	7.6	70	-6.7	64	1.1	72	2.0	73	21.7	71	70
8	8.7	66	-7.8	59	1.4	65	2.3	67	21.6	72	66
9	6.6	74	-4.2	78	0.8	80	1.2	83	13.3	82	80
10	10.3	60	-5.3	72	1.5	62	2.3	69	14.6	81	68
Average	9.8	62	-6.2	67	1.6	60	3.1	58	25.0	67	63

Table 1: Results of the comparison metrics and corresponding scores for all ten test cases

Acknowledgements

This work has been funded, in parts, by a grant from Pathfinder, Inc. to Vanderbilt. B.M. Dawant has a financial interest in this company.

References

1. Pan S., Dawant B.M., “Automatic 3D segmentation of the liver from abdominal CT images: a level-set approach”, SPIE Proceedings vol. 4322, Medical Imaging 2001: Image Processing, Milan Sonka; Kenneth M. Hanson, Editors, pp.128-138
2. B.M. Dawant, S. Pan, and R. Li, “Robust Segmentation of Medical Images Using Geometric Deformable Models and a Dynamic Speed Function”, LNCS 2208/2001, Proceedings of the 2001 MICCAI conference, pp. 1040-1047.
3. Z. Cao, “Segmentation of medical images using level set based methods”, Ph.D. Dissertation, Department of Electrical Engineering and Computer Science, Vanderbilt University, 2004
4. R. Malladi, J.A. Sethian, and B.C. Vemuri, “Shape modeling with front propagation: A level set approach”, *IEEE Trans. PAMI*, **17**(2):158-175, 1995
5. Adalsteinsoon, D. and Sethian, J. A., "The fast construction of extension velocities in level set methods," Journal of Computational Physics, vol. 148, no. 1, pp. 2-22, 1999.