

An Interactive Framework for Boundary Delineation for Medical CBIR

Christina Pavlopoulou, Avi Kak, Carla Brodley
{pavlo, kak, brodley}@ecn.purdue.edu
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907

Abstract

CBIR systems designed for medical applications often require that a human in the loop demarcate the pathology bearing regions in the image, since fully automatic extraction of such regions is still not possible. In our CBIR system for the domain of HRCT images of the lung, physicians did not find this interaction too onerous since the boundary surrounding a pathology bearing region did not have to be precise. But in our new domain, the liver image database, that is unfortunately not the case. The boundaries supplied by the physician must correspond precisely to the outline of the liver or to the boundary of a pathology bearing region inside the liver. To meet this demand, we have developed a new user interaction framework for semi-automatic boundary extraction. All that a physician has to do is to click on a couple of pixels on the boundary to be extracted. The system then tries its best to extend the boundary as far as possible, sometimes even extracting the entire contour correctly. When errors occur, all that the physician is called upon to do is to click on where a correction to the boundary needs to take place. In this manner, an entire boundary can be specified with very little input from the human, which is a most important consideration with physicians as he/she can hardly be expected to click on every boundary point.

Keywords: Medical CBIR, interactive segmentation, deformable shapes.

1. Introduction

Based on our previous work [17] and the work of others [4, 8] it is now clear that CBIR for medical imagery requires a human in the loop for localizing the pathology bearing regions. Our system for the CBIR of HRCT (High Resolution Computed Tomography) images of the lung calls upon a physician to circumscribe the pathology bearing regions (PBR) in the cross-sectional images of the lung at the time the image is entered into the database. The images are in-

dexed for storage and subsequent retrieval on the basis of the attributes of the pathology bearing regions.

In the system presented in [17], the physician can very quickly circumscribe a PBR because, as we showed in [16], it is not critical for the delineated boundary pixels to correspond exactly to the outline of the PBR. Our results in [16] show that even when the area marked by a physician is up to 50% larger than the actual area of a PBR, the retrieval precision is not much affected. But, unfortunately, this laxity with which a local feature can be delineated in an image does not extend to all medical domains. Consider, for example, the HRCT images for the liver domain, examples of which are shown in figure 1. Characterization of liver pathology requires that the extracted boundaries of the liver correspond more or less exactly to the actual boundary of the liver. By the same token, it is essential that the boundaries of any PBR's found inside the liver or in its vicinity correspond as closely as possible to the true boundaries.

This has necessitated that we modify the physician's interaction with the images. Of course, if the physician's time and patience were not important issues, we could ask the physician to simply click on a dense set of points on the actual boundary of the anatomical feature of interest. But that obviously is not the case. We have therefore devised a new interactive strategy for contour delineation that seeks to do the most (of boundary extraction) with the least (of physician's input). Initially, the physician only has to click on a couple of points. Using these boundary points, the system tries to set up an optimum discriminator for distinguishing between the inside of the true boundary and the outside. On the basis of this discriminator, the system tries to draw out the entire boundary (usually in a fraction of a second). For example, figure 2 shows the bounding contour as constructed by the system using the two pixels shown (the first is marked by an arrow) that were supplied by a human. If the bounding contour is incomplete, as is the case in the example shown, or if the physician believes that the computed boundary does not correspond sufficiently closely to the actual boundary, he/she can click on one more point. The sys-

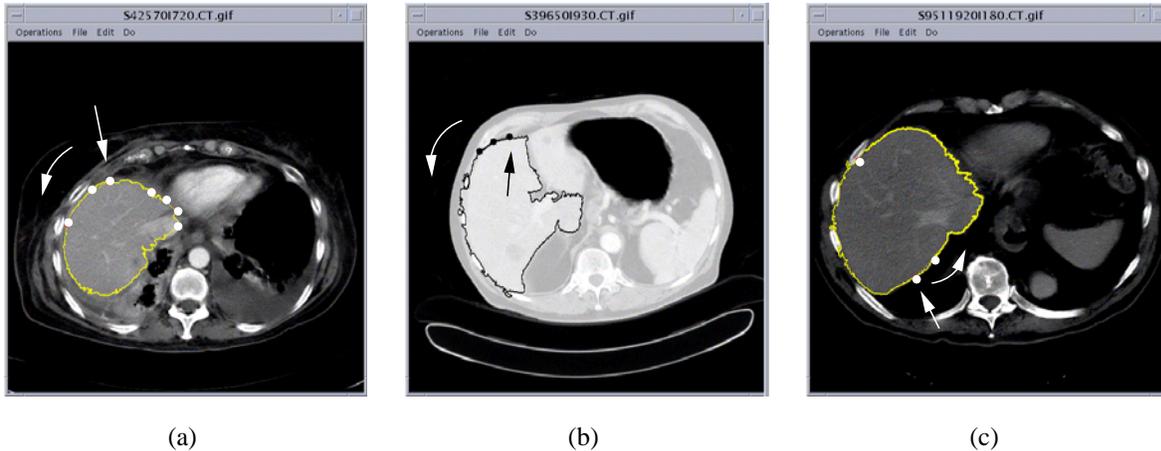


Figure 1. Three different images from our liver database. As can be seen here, the images exhibit considerable patient-to-patient variation with regard to shape, grayscale, contrast, etc., of the anatomical features.

tem takes this additional evidence into account to modify the discriminant function and presents a new boundary contour to the physician. Figure 2 (b) shows the modification of the bounding contour of figure 2 (a) using one more pixel (shown by another arrow) supplied by the human. Although this process can be continued with an arbitrary number of points, usually it takes a couple of mouse clicks to extract the entire bounding contour. In the example shown, the entire boundary was extracted with just three clicks from the user.

The paper starts with a review of the previous work on interactive approaches to boundary extraction. Subsequently, we describe our approach in detail, *highlighting the research questions that have been raised by our new approach to user interaction*. Finally, we will show some experimental results on HRCT images of the liver.

2. Human Assisted Extraction of Boundary Contours – Past

Ours is, of course, not the first attempt at semi-automatic extraction of boundary contours in images. But the approaches presented previously by others are not appropriate for CBIR for a variety of reasons, some because they are too demanding in terms of the a priori specification of some critical parameters needed by the computations, and others because they are too computationally expensive for physician interaction on the workstations of the day. When there is as much image to image variation as we show in the sample images of figure 1, it is simply not possible to use the same set of parameters and decision thresholds for all the images in the database.

One of the most well-known paradigms for interactive extraction of bounding contours is that of snakes [9]. The main idea is that the user first roughly delineates the boundary of the object and then image derived terms, which typically depend on the intensity of edges, are used to attract these markings to the real boundary. The initial curve is deformed until its internal energy becomes approximately equal to the strength of the image terms. This formulation requires an a priori set of parameters that regulate the extent to which the curve is allowed to follow gray level discontinuities. This limits the applicability of the method to domains where these parameters can be set reliably for a large set of images. Of course one can always request the user to change these parameters but this makes the interaction more difficult and time consuming.

Other semi-automatic approaches that extend the original snakes formulation to make it more user-friendly include growing snakes [3] and ziplock snakes [13]. In ziplock snakes, the user initially is required to select two points on the boundary of the object. The line connecting these two points starts to deform sequentially (parts close to the user-selected points are deformed first) until approximate balance of image and curve energies is achieved. If the user does not like the result, he/she can provide more intermediate contour points and the above procedure is repeated. Although the goals of this method are the same as ours there are several differences. Again the amount of deformation allowed depends on the edge extractor used. Furthermore, the optimization technique used for a given set of points is iterative and more computationally intensive than ours.

Another extension of the snake interaction paradigm can be found in [2, 11, 12]. The image is represented as a

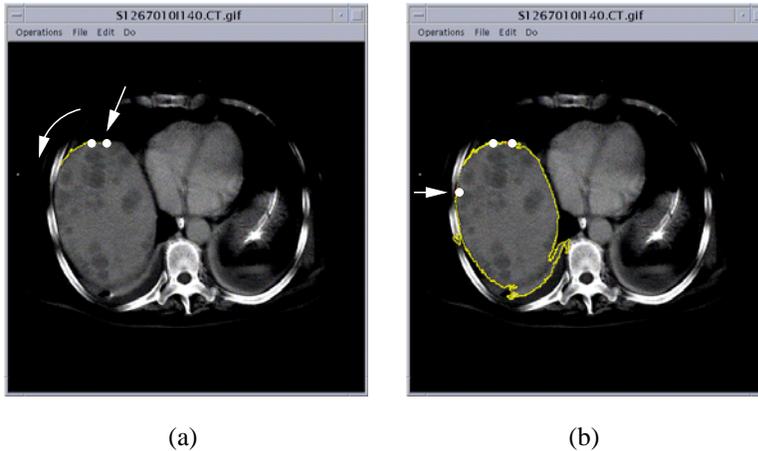


Figure 2. In (a) the straight arrow points at the first pixel marked by the user as a boundary point and the arrow with the curved tail shows the direction in which the boundary is extracted. In (b) the arrow points to the additional point clicked by the user in order to continue the boundary extraction process. So, the extraction of the complete bounding contour shown in (b) took only three clicks from the user in this case.

weighted graph, whose each node is a pixel or a small region of the image and each edge is a weighted average of terms like intensity of grayscale, local smoothness, etc. The user again is required to click on boundary points. Every time this happens, the shortest paths between the node corresponding to the point selected and all the other nodes in the graph are computed and the user must select one of them using the movement of the mouse. This makes user input expensive. Furthermore, the range of applicability of the method depends on the edge detector selected and the relative weights of the features determining the weights of the edges. The user can also allow the determination of the feature weights based on his/her response (training).

Other schemes for interaction are the ones described in [14] and [5]. In [14], the idea is to identify the causes for failure of a segmentation method and then define operations that try to fix them. This however requires more complicated user input as well as knowledge of how the particular operations were implemented and how they exactly work. In [5], the authors introduce a system which aims at providing high level image editing operations; this goal is totally different than ours.

3. Framework

Figure 3 is a diagrammatic illustration of our approach. As mentioned in the Introduction, the user interaction is iterative. The user provides initial input, which the system uses to build a local model according to which the pixels in the neighborhood are classified as belonging to the two

sides of a bounding contour. If the user does not like the resulting bounding contour, he/she can modify it by providing additional input which is used by the system to train a different local model. This procedure is repeated as long as desired by the user. Our figure 2 illustrated this procedure graphically. The initial input there consisted of the two marked pixels, the first of which is pointed to by an arrow in the frame on the left. Using this data, the system made a model (as a distribution function) of the gray levels on the two sides of the contour. Subsequently, the user had to supply one additional point, as shown by the third white dot on the contour in the frame on the right, for the entire contour to be extracted correctly.

This approach raises a host of questions:

- What model to use for representing the gray levels on the two sides of the bounding contour? This question becomes even more complex for color imagery and when it is desired to also use texture discontinuities for bounding contours.
- In general, no single model will describe all the exterior pixels in the vicinity of a bounding contour. In general, that will also be the case for the interior pixels. That raises the question of how to alter the model as the boundary grows either because of the additional inputs from the user or because the system has extended the previously supplied human input.
- Should there be any constraints on the placement of the initial boundary points by a human? The constraints

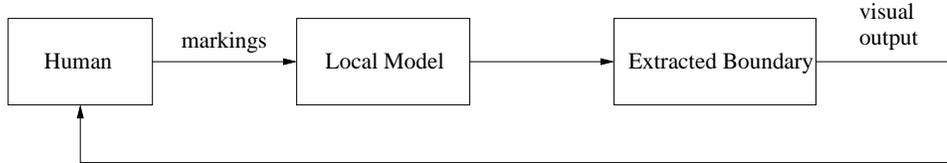


Figure 3. A diagrammatic representation of the human interaction in our approach.

could be on the spacing between the points, the number of points, etc.

- When the human supplies a new data point to correct the errors seen in the contour extracted so far, how to decide which part of the previously extracted contour to discard and how to join that contour with the new datum?
- What algorithms to use to smooth out the noise in the extracted bounding contours?
- For a given set of human supplied boundary points, how far to extend the boundary contour automatically? In other words, what decision thresholds to invoke to stop the automatic extension of the boundary contour using the currently available human input?

With regard to modeling, the model used must possess sufficient discriminatory power to distinguish between the regions that are on the two sides of the human supplied boundary points. For our first CBIR demonstration project based on this new mode of interaction, we have confined our attention to images with smoothly varying gray levels, as is the case with the images in the liver database. These images generally have low contrast between adjoining anatomical features, although the overall gray level variation in an image can be quite large. (There is also large patient-to-patient variation for the same anatomical features in these images – a fact that speaks against the use of global models and global signatures for indexing.)

We believe that for such images it suffices to use normal distributions to describe the pixels on the two sides of a small part of the bounding contour. Thus, the two sides of such a contour would be distinguishable on the basis of their means and variances. But obviously, a single value for the mean and a single value for the variance would not describe all the exterior pixels adjacent to a boundary contour.

Initially, these parameters are estimated from the pixels in a small window on either side of human selected points. The usual formulas are used for these estimates:

$$m_c = \frac{1}{n_c} \sum_{j=1}^{n_c} x_j \text{ and } v_c^2 = \frac{1}{n_c - 1} \sum_{j=1}^{n_c} (x_j - m_c)^2$$

m_i and v_i^2 are the mean and variance of the c side, namely left (L) or right (R).

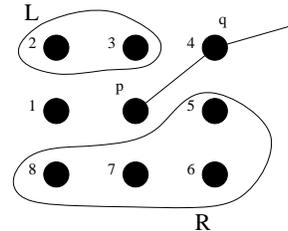


Figure 4. 8-neighborhood of pixel p

The size of the window is selected, so that the model constructed achieves the maximum separability between the left and the right sides of the contour. As a measure of separability we have used the ratio of the between class scatter with the within class scatter ([6]), which in this case is given by

$$J(m_L, v_L^2, m_R, v_R^2) = \frac{(m_L - m_0)^2 + (m_R - m_0)^2}{v_L^2 + v_R^2}$$

m_0 is the expected value of m_L and m_R .

Subsequently, a naive Bayesian approach [6] is used to classify the pixels as boundary or not. Very similar formulations include [7] and more recently [15]. Let p be the last pixel classified as boundary and q the one before it. We consider the 8-neighborhood of p and let p_i be the i -th pixel in this neighborhood numbered as in figure 4. From Bayes rule we have:

$$P(c|p_i) = \frac{P(c) P(p_i|c)}{P(p_i)}$$

where c is L or R . The class c of pixel p_i is given by the maximum of the a posteriori probabilities $P(R|p_i)$ and $P(L|p_i)$. This gives the well known decision rule:

$$h(p_i) = \frac{P(L) P(p_i|L)}{P(R) P(p_i|R)} \geq 1$$

In what follows we assume that the classes L and R are equiprobable.

Assume for a moment the scenario of figure 4 where the contour has so far been extended to the pixel p shown in the center via the pixel marked p_q . Now we must decide which pixel to declare as the next boundary pixel. To explain, the left-neighbor pixel p_1 belongs on the boundary if the pixels

p_2 and p_3 belong to class L and the pixels p_5 , p_6 and p_7 belong to class R . Given that we have already classified pixels p and q , then the best pixel to continue with is the one for which the probability of the event pixels p_q to p_1 belong to L and pixels p_2 to p_q belong to R is maximized. This probability can be easily computed by assuming independence of the events {pixel p_i belongs to class c }, and using the above assumptions.

As mentioned earlier, a single value of the mean and a single value for the variance will, of course, not describe all of the exterior or the interior pixels adjacent to the boundary. To get around this problem, our approach is based on the tracking paradigm, in the sense that the system tries to track the boundary using the data supplied by the human. Each time the human supplies new input, we update the means and the variances to reflect the properties of the pixels in the vicinity of the new point supplied by the user. The system then tries to extract as much of the boundary as it can using the new values of the means and the variances.

As the reader would expect, despite the fresh computation of the means and the variances for each new human-supplied boundary point, the extension of the boundary can certainly be expected to go haywire when there is a significant change in the properties of the anatomical features exterior to the liver. For example, when the exterior anatomical feature changes from a soft-tissue structure to a hard-tissue structure, automatic extension of the boundary on the basis of previously computed means and variances cannot work. This difficulty is illustrated with the help of Figure 5 (a). While the system automatically extends the boundary contour up to the point marked A, but then the process breaks down because the contour starts to go around the hard-tissue structure.

How to elicit further human input to fix such errors and how to decide what portions of the previously computed boundary to retain are research issues unto themselves, as we already mentioned at the beginning of this section. For the domain of liver images, the following heuristic, illustrated with the help of Figure 5 (b), has worked well: The user is expected to click at a point that is on the correct boundary and that is as close as possible to the latest correct pixel on the computed boundary. In the figure, the pixel marked A is the last correct pixel on the computed boundary. After examining this output, the human clicks on point B which is on the correct boundary. The system then retains the previously computed boundary up to point A, joins A with B, and continues on beyond B as before.

We have so far left unaddressed the issue of how to represent the boundary itself, especially the boundary segment that falls between the initially marked points supplied by the human. Since in our current project the human supplies only two boundary pixels initially, we have no choice but to assume that the boundary segment between those two points

is straight. For this reason, it is important that those two pixels be fairly close to each other so that this assumption is not violated. When the system then adds a boundary pixel to the two already provided, it again makes the assumption that the boundary between the last point and new point is straight. So while the boundary pixels are represented simply as a list of points, there is the underlying assumption that the boundary segment between adjacent points in the list is a straight edge. In the future, we hope to experiment with spline based representations to see what tradeoffs we may be able to achieve between the computational effort involved and the smoothness of the contours.

Regarding the question of when to stop the automatic growing of a boundary contour, the system stops when we return to a point close to the one we started from, or the number of points we tracked is larger than a previously specified large number.

4. Results

Efficient human interaction is central to the success of a medical CBIR system. Physicians are often short on time and any input elicited from them has to be small and the feedback instantaneous. An important contribution we have made in this paper is to point this out with regard to the delineation of pathology bearing regions. We demonstrated an interactive boundary extraction scheme that, at least for the domain of liver images, has just the right flavor of interaction.

With regard to the correctness of the extracted contours, by definition they will always be correct because the human is there to steer the contour away from erroneous extensions. So in order to judge whether or not our approach is technically sound, the issue is not whether the extract contour is correct, but how many user clicks does it take to produce a contour that completely circumscribes an anatomical feature. A second issue is how the number of clicks needed depends on the choice of the initial points supplied by the human. We refer to the second issue as *stability*.

To answer the above issues we interactively segmented 110 of the 414 images our database currently contains. Many of these images are very similar because they are slices belonging to the same patient. So by restricting the number of images we process, we do not lose much from the validity of the results. The results of this experiment are reported in table 1, where we display the mean number of clicks and its standard deviation required to extract a complete contour for four different starting positions. We segmented all images clockwise (this direction was arbitrarily chosen), and for each image we started the delineation from 4 different locations, namely the top of the liver, the right side, the bottom and the left side. For each starting point we computed the mean and the standard deviation of

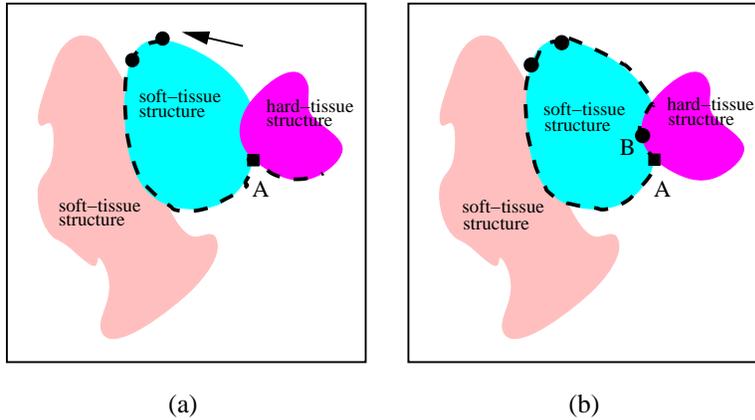


Figure 5. The means and the variances computed for the pixels in the vicinity of the initial points supplied by a human (these points are marked by an arrow in the left frame) will result in errors in boundary extraction when the pixel parameters change significantly, such as when the exterior adjoining structure changes from a soft-tissue structure to a hard-tissue structure. The point A marked in the frame on the left shows such a pixel. Such errors can be fixed by the user clicking on the correct boundary pixel, such as the pixel marked B, in the vicinity of point A.

	top	right	bottom	left
mean	7.58	7.68	7.85	6.87
std. dev.	3.67	3.58	3.77	3.51

Table 1. A study of the performance of our interactive segmentation approach on 110 HRCT liver images starting from four different locations: top of liver, right side, bottom and left side. The mean and the standard deviation of the number of clicks required for each starting position was computed.

the number of points needed to extract the boundary. As the table shows, on the average 7 points were needed to extract the liver from a variety of images. Furthermore, this number of points does not seem to depend upon the starting point. Therefore, the method is stable. Stability is also illustrated in figure 6. For the same liver image in each of the six frames shown, the arrow points to the initial boundary pixel supplied by the user. The number of white dots is the total number of boundary pixels that the user had to supply in order for the complete contour to be extracted. So, for the start point marked by the arrow in frame (a), the user had to supply a total of three points, as shown by the white dots. For frame (b), it was only two points. For frame (c), it was again only two points, and so on.

All of the extracted boundary contours shown so far have been for the entire liver. That the technique works just as

well for PBR's is illustrated by our figure 7. There we show the extracted boundaries of low-attenuation liver lesions that can be very difficult to segment otherwise.

We have also employed the liver contours extracted with our method for indexing and retrieval based on shape similarity, which is an important cue for liver diseases. Currently, our database consists of 414 HRCT slices belonging to 13 different patients. Ideally, one would like to retrieve slices belonging to different patients than the one of the query image, however 13 is a small number for such an experiment. For this reason, we included 2 to 3 slices from each patient with the rationale that the shape of the liver for slices placed apart varies hugely and thus we do not introduce dependencies that could affect significantly the retrieval output. In total we used 30 images. Each image served as a query image and the 4 most similar images were retrieved. Because in this case we do not have ground truth, we evaluated the output by counting the number of retrieved images we considered similar with the corresponding query image. Similarity of the liver contours was computed by representing them as sparse polygons and subsequently comparing their turning functions using the method of [1]. To find the sparse representation we used the method of discrete evolution introduced in [10]. The total retrieved images were 120 and from them we considered "correct" 82, yielding a precision accuracy of %68. We believe this percentage will increase as we expand our database.

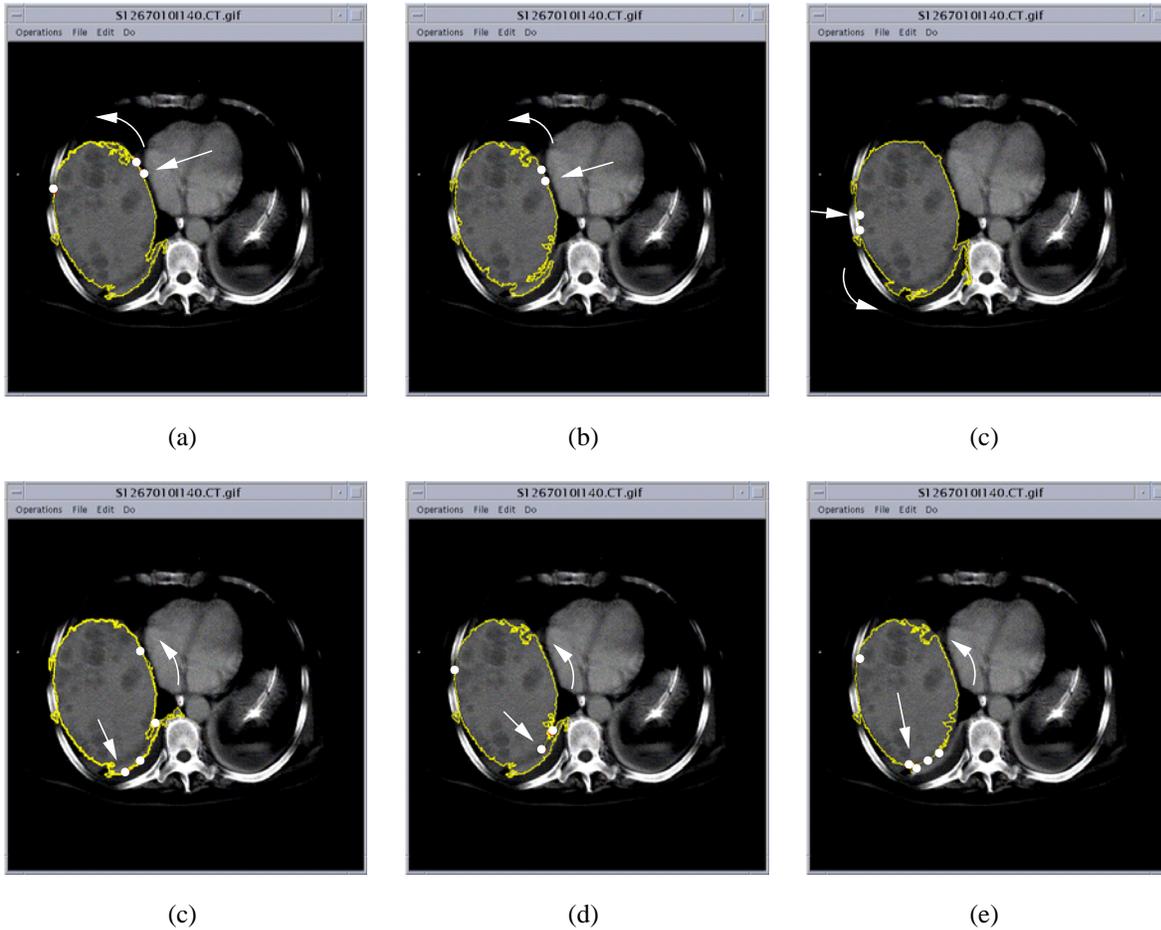


Figure 6. A study of the stability of the boundary delineation process. The arrow in each frame points to the first boundary pixel marked by the human. All of the human supplied points are shown by white dots. The frame in (a) takes the three human-supplied points shown as the white dots, the frame in (b) the two points shown, etc.

5. Conclusions and Future Work

We have argued that some medical image domains for CBIR require a precise delineation of the boundaries of the anatomical features and of the pathology bearing regions. Since at present such boundaries must be supplied interactively, an important measure of the suitability of a medical CBIR system is how demanding this interaction is on the physician's time. In this paper, we presented an interactive approach that was designed specifically for the domain of liver images. This approach is capable of yielding an entire bounding contour with just a few judiciously placed mouse clicks by the physician. Future work includes carrying out a quantitative evaluation of this procedure in collaboration with the physicians as well as extending the approach to color images and to images where boundaries consist of tex-

ture discontinuities.

Acknowledgments

This work was supported by the National Institute of Health under grant No. 1 R01 LM06543-01A1. We would like to thank our domain expert Dr. Alex Aisen of the Indiana University Medical Center for providing us with the data and for many helpful discussions.

References

- [1] E. Arkin, P. Chew, D. Huttenlocher, K. Kedem, and S. Mitchell. An Efficiently Computable Metric for Comparing Polygonal Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3), 1991.



Figure 7. Our technique works equally well for the hard-to-segment low-attenuation liver lesions shown here.

- [2] W. Barrett and E. Mortensen. Interactive Live-Wire Boundary Extraction. *Medical Image Analysis*, 1(5):331–341, 1997.
- [3] M. Berger and R. Mohr. Towards Autonomy in Active Contour Models. In *Proc. International Conference on Pattern Recognition*, 1990.
- [4] W. W. Chu, C. C. Hsu, A. F. Cardenas, and R. K. Taira. A knowledge-based image retrieval with spatial and temporal constructs. *IEEE Transactions on Knowledge and Data Engineering*, 10(6):872–888, 1998.
- [5] J. Elder and R. Goldberg. Image Editing in the Contour Domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3), 2001.
- [6] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.
- [7] F. Ghelli. A Sequential Learning Method for Boundary Detection. *Pattern Recognition*, 21(2):131–139, 1988.
- [8] C. C. Hsu, W. W. Chu, and R. K. Taira. A knowledge-based approach for retrieving images by content. *IEEE Transactions on Knowledge and Data Engineering*, 8(4):522–532, 1996.
- [9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. In *Int'l Conference on Computer Vision*. IEEE, 1987.
- [10] L. Latecki and R. Lakaemper. Polygon Evolution By Vertex Deletion. In *Scale-space Theories in Computer Vision. Proc. Int'l Conf. Scale-Space*. M. Nielsen and P. Johansen and O. Olsen and J. Weickert eds, 1999.
- [11] E. Mortensen and W. Barrett. Interactive Segmentation with Intelligent Scissors. *Graphical Models and Image Processing*, 60(5), 1998.
- [12] E. Mortensen and W. Barrett. Toboggan-Based Intelligent Scissors with a Four-Parameter Edge Model. In *Proc. Computer Vision and Pattern Recognition*. IEEE, 1999.
- [13] W. Neuenschwander, P. Fua, L. Iverson, G. Szekely, and O. Kubler. Ziplock Snakes. *Int'l Journal of Computer Vision*, 25(3):191–201, 1997.
- [14] S. Olabarriaga and A. Smeulders. Setting the Mind for Intelligent Interactive Segmentation: Overview, Requirements, Framework. In *Int'l Conference on Information Processing in Medical Imaging*, 1997.
- [15] N. Paragios and R. Deriche. Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:266–280, 2000.
- [16] C. Shyu, C. Brodley, A. Kak, A. Kosaka, A. Aisen, and L. Broderick. Local versus global features for content-based image retrieval. In *Workshop of Content-Based Access of Image and Video Databases*, pages 30–34. IEEE, 1998.
- [17] C. Shyu, C. Brodley, A. Kak, A. Kosaka, A. Aisen, and L. Broderick. ASSERT: A physician-in-the-loop content-based image retrieval system for HRCT image databases. *Computer Vision and Image Understanding*, 75(1/2):111–132, 1999.