Applications of Semisupervised and Active Learning to Interactive Contour Delineation

Christina Pavlopoulou PAVLO@ECN.PURDUE.EDU
Avi Kak KAK@ECN.PURDUE.EDU
Carla Brodley BRODLEY@ECN.PURDUE.EDU
Purdue University, School of Electrical and Computer Engineering, West Lafayette, IN 47907 USA

Abstract

We describe a novel application domain for semi-supervised and active learning algorithms, namely that of intelligent interactive contour extraction. It is well-known that object delineation is an ill-posed problem unless guided by the human or by apriori constraints and models. We focus on user-steered extraction, which has been the focus of investigation in a large volume of work in computer vision. We will discuss how this problem can be naturally translated to a semi-supervised and active learning problem and we will describe our work so far towards investigating the issues involved.

1. Introduction

Contour extraction is one of the important problems in computer vision, however in the general case it is ill-posed unless guided by a human or by apriori set constraints and models. We will concentrate here on the human-guided case. Of course, the human could manually delineate the entire contour, however this is very tedious and error prone. The goal is therefore to delineate the desired contour with the least human intervention possible. The human provides steering guidance to the growing contour when it takes off in a wrong direction. By casting the problem in a learning/statistical framework the goal is similar to the goal of active learning and semisupervised learning.

Human-steered methods have been the focus of investigation of a large number of papers in computer vision, although they are not usually combined with learning. Typically, these methods are iterative and alternate between two steps: human input and boundary estimation. More precisely, the human provides, for example a rough approximation of the desired boundary or some points that belong to the boundary. This input is subsequently used to estimate as much of the true boundary of the object as possible. The result of this optimization is displayed and if the user is not satisfied with the output, then he/she can provide additional input and the process is repeated.

Characteristics of the methods proposed so far include:

- They require prior estimation of edges in the image, which is also an ill-posed problem.
- The information inferred from human input - typically mouse clicks on the border of the contour to be delineated - is only the coordinates of the clicks which affects the optimization. However, additional information can be inferred if we take into account the feature values of this region. For example, there could also exist *some discriminatory information* that the human perceives and thus identifies the boundary.
- Finally, the learning and statistical modeling usually attempted for this problem are typically *supervised* or *unsupervised*. In the supervised case a set of training images or human markings is used to estimate the statistics of the object to be extracted and/or the background expected. Using other images for estimation is pretty restrictive given the immense variability of objects and backgrounds. In the unsupervised case, the contour is deformed until the best discrimination between the statistics of the object and the background is achieved. This is a reasonable goal only if the object boundaries correspond to strong edges and there are no other strong edges in the vicinity of the object, which is not the case most of the times.

The above issues are resolved if one uses a semisupervised model of learning. The human can provide

Proceedings of the ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data, Washington DC, 2003.

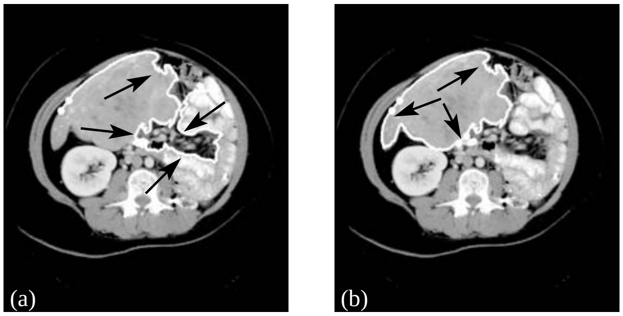

Figure 1. Sample steps of interaction with an HRCT liver image using our new method. (a) Initial user input shown with white dots and the estimated contour shown in white. (b) Subsequent user input and contour estimation. Black arrows point to the estimated contour.

some initial labeled data which can be used to build a classifier to distinguish between the two sides (inside/outside) of the object to delineate. This classifier can be used to estimate as much of the true boundary as possible. Given that the labeled data will be sufficient to identify small parts of the boundary only, incorporation of unlabeled data in the construction of the classifier could allow adaptation to changes in the grayscale and more reliable estimates given the small amount of data the user has entered.

It is an issue of course how training data will be elicited from the user. Given that the user is unlikely to know what parts of the boundary are important for training an active learning type of interaction is preferable. That is, the user provides some small part of the boundary which is used as labeled data to train a classifier. The boundary is extended and is displayed to the user for inspection. If the user disagrees with the output, he/she provides additional input which corrects and extends the boundary and modifies the classifier constructed. An interaction of this type is illustrated in Figure 1.

Some of the characteristics that this problem exhibits are the following. It is a 2-class problem since we are interested in discriminating between the two sides of the contour. Spatial constraints are important: the pixels found on each side of the contour should belong to the same class. Since we are aiming for an interactive system whatever learning algorithms we employ should be on-line. Furthermore, the human input applies to a small region of the image only and the statistics of the image might change drastically even in neighboring regions. For this reason an incremental approach to boundary estimation and classifier updating are preferable. Finally, the initial data used for

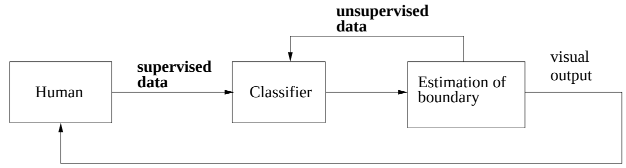

Figure 2. A diagrammatic representation of our framework for interactive contour delineation.

training are not uniformly sampled from the whole image which is what is usually assumed in semisupervised techniques.

Our approach so far consists of a piecewise extension of the boundary and an incremental updating of the classifier. These steps are illustrated in Figure 2. Given the human input on a local region of the image we build a Bayesian model suitable to characterize that local region. We extend the boundary in a small neighborhood in the maximum likelihood sense given the model previously constructed. Subsequently, we update the Bayesian model with the pixels that maximize the local separability. The process continues until a maximum number of pixels is estimated and the output is displayed to the user for inspection. If the user does not like the results and decides to provide additional input the model is modified accordingly.

We start the description of our method by first reviewing related previous work in Section 2. In Section 3 we provide the details of our interaction protocol with the human. Sections 4 and 5 contain the details of our statistical modeling and optimization involved. We provide experimental results in Section 6 and we conclude with Section 7.

2. Previous Work

One of the most well-known paradigms for interactive extraction of bounding contours is that of snakes (Kass et al., 1987). The main idea is that the user initially provides a rough delineation of the boundary of the object which is subsequently deformed until its internal energy becomes approximately equal to the strength of image derived terms. The computation of these terms usually depends on edge detection. In typical applications of this framework learning is not involved. However, there have been some approaches that employ statistical modeling, though not with user-interactivity in mind and the learning involved is either supervised or unsupervised.

Some of these approaches include (Zhu & Yuille, 1996; Figueiredo et al., 2000; Paragios & Deriche, 2002).

Formulations that appear in (Zhu & Yuille, 1996; Figueiredo et al., 2000) aim to segment the entire image so that the discrimination between the statistics of different regions is maximized subject to smoothness constraints. This criterion is similar to unsupervised clustering. In (Paragios & Deriche, 2002) a model of the interior of the object and/or the background is constructed based solely on human markings provided initially; thus the learning is supervised. No alteration of the model is attempted during the segmentation and the human "must know" what parts of the image are necessary for the construction of the statistical model, which is not the case most of the times.

Another interesting and well-known interaction paradigm is the live wire or intelligent scissors method (Falcao et al., 1998), (Mortensen & Barrett, 1998). In this approach, a weighted graph representation of the image is assumed. Each image pixel is a node and the weight assigned to an arc connecting any two adjacent nodes is determined by the image edge intensity, grayscale value, local smoothness, etc., at the pixel corresponding to the node. Given two user supplied points the boundary is estimated using a shortest path algorithm between the user points. The critical part of this approach is the construction of the weight function which heavily depends on the edge extraction method and the weights of the features used. Definition of these weights based on the human input has been attempted but it involves parameters arbitrarily defined.

More recently graph-cut algorithms have been used for optimizing energy-based criteria for interactive segmentation (Boykov & Jolly, 2001), (Xu et al., 2003). The work reported in (Boykov & Jolly, 2001) could be considered semi-supervised however the emphasis is on the graph formulation part and no results are reported with respect to how much user input is needed.

Furthermore, there have been some methods (Bensaid et al., 1996; Velthuizen et al., 1994) that employ semisupervised clustering in order to segment the entire image. Semisupervised in this case essentially means initialization of k-means by the human. That is, the human specifies the number of clusters and a few pixels in each cluster from which the initial means are computed. However, the method is applicable for very homogeneous regions only since each cluster corresponds to one region.

In machine learning semisupervised learning has been investigated both in a generative (Nigam et al., 2000; Ghahramani & Jordan, 1994) and in a discriminative context (Bennett & Demiriz, 1998). We have opted to follow the generative model because it can be implemented efficiently and the densities of the classes are computed at the same time. The latter allows the extension of the boundary in the maximum likelihood sense. However, unlike the approaches tried so far which use EM to fill in the missing data, we follow the two step approach mentioned in the introduction, i.e. we estimate some part of the boundary and then we update the Bayesian model. We do that for two main reasons. First this technique allows us to incorporate the spatial constraint that if one pixel belongs to one side of the boundary then it must have the same label with all the pixels of the same side. The second reason follows from the problem statement itself. We are not interested in labeling the pixels of the entire image; we want to delineate a particular object and to this end only a few pixels in the neighborhood of the boundary are relevant. Thus, we have to select what data we will use for the construction of the classifier and by incrementally extending the boundary we include first the most likely ones.

3. Interaction Protocol

The interpretation of the human input is very important for this problem since the supervised data computed will steer the estimation of the contour towards the desired direction. The initial input provided by the user is two points such that the line connecting them is on the desired boundary as illustrated in Figure 3. Using features extracted from small regions on the two sides of this line a Bayesian classifier is built. We currently use grayscale as a feature, however the methodology can be extended to color, texture, or a combination of them. For this type of images, it is sufficient to describe the pixels on the two sides of a small bounding part with normal distributions. The size of the regions, from which the model parameters will be computed, is selected, so that the model constructed achieves the maximum separability between the two sides of the contour. As a measure of separability we have used the ratio of the between class scatter with the within class scatter (Fukunaga, 1990), which in this case is given by

$$J(m_L, v_L^2, m_R, v_R^2) = \frac{(m_L - m_0)^2 + (m_R - m_0)^2}{v_L^2 + v_R^2} \tag{1}$$

m_L, m_R, v_L^2,v_R^2 are the means and the variances of the distributions of the two sides of the boundary, namely L and R. m_0 is the expected value of the pixels on both sides.

After the initial estimation the contour is extended and and the classifier is updated. We defer detailed description of these steps to Section 4. The estimated

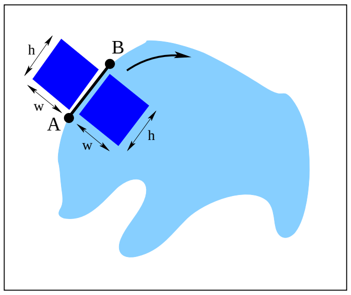

Figure 3. Initial input provided by the user.

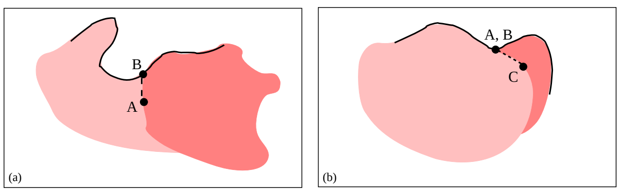

Figure 4. Interpretation of subsequent user input.

boundary is displayed so that the user determines whether or not it is the correct one. If the user does not agree with it, he/she can provide additional input to steer the estimation towards the desired result. We have made the following conventions to avoid ambiguities (Figure 4). Let A be the point entered by the user and B the closest point of the estimated boundary to A.

- If the Euclidean distance between A and B is larger than a pre-specified threshold, then A and B are connected with a line, a model is constructed using this segment, and the process starts over (Figure 4(a)).
- If the distance is smaller than the threshold, then the system waits for another click, say C. Then B and C are connected with a linear segment and the process start over (Figure 4(b)).

In both cases the model and the boundary estimated after point B are discarded.

4. Modeling

The objective goals are boundary extension and incorporation of unlabeled data. We can extend the boundary in the maximum likelihood sense given the statistical model constructed. Incorporation of unlabeled data can be achieved by fitting statistical models to the pixels on the two sides of the estimated boundary.

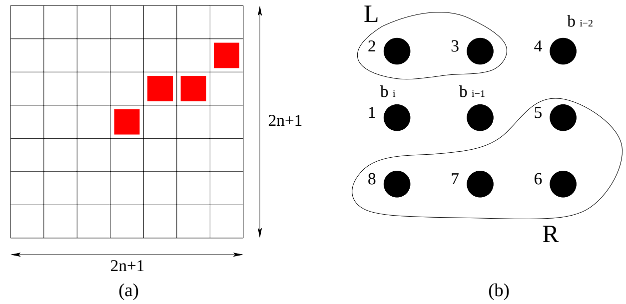

Figure 5. (a) Estimation of the n best pixels that can follow p, the last estimated boundary point. Gray cells indicate previously estimated boundary points. (b) 8-neighborhood of pixel x_{i-1}

Note that boundary estimation depends on the model estimation and vice versa.

We proceed with the notation required to formalize the above objectives. We will use capital letters for random vectors and small for random variables. Let $\mathbf{B}$ be a random vector denoting the boundary we seek to estimate. $\mathbf{B} = (b_1, \ldots, b_N)$ where b_i is a discrete random variable denoting the i-th pixel of the boundary. Let L and R denote the two sides of the boundary, $M = (M_L, M_R)$ denote the model parameters of the two sides and $X = (X_L, X_R)$ the set of feature values of the pixels of the two sides of the boundary B. In this paper, we focus on the grayscale values of the pixels, although the formulation is the same for other types of features like color and texture. Finally, we denote the grayscale values of the 8-neighborhood of pixel b_i with X_i, and we use x for the grayscale value of a single pixel.

4.1. Boundary Estimation

The optimal boundary B^* in the maximum likelihood sense given a model M, a starting pixel b_1 supplied by the user and the grayscale values X of the pixels in the neighborhood of the boundary is:

$$B^* = \arg\max_B P(b_2, \ldots, b_N | b_1, X, M) \tag{2}$$

Because the boundary is one-dimensional and connected, we can model it as a Markov chain. For computational efficiency, we assume first order Markov dependencies between the boundary points which allows us to write the joint probability in the following form:

$$P(b_2, \ldots, b_N | b_1, X, M) = \prod_{i=1}^{N-1} P(b_{i+1} | b_i, X, M) \tag{3}$$

To calculate the conditional probabilities we use the

8-neighborhood of each pixel as illustrated in Figure 5(b). Then we have

$$P(b_i|b_{i-1}, b_{i-2}, X_{i-1}) = \frac{P(b_i, b_{i-1}, b_{i-2}|X_{i-1})}{P(b_{i-1}, b_{i-2}|X_{i-1})} \quad (4)$$

where X_{i-1} denotes the grayscale values of the pixels in the neighborhood of b_{i-1}. The denominator of the above equation can be computed as a marginal probability:

$$P(b_{i-1}, b_{i-2}|X_{i-1}) = \sum_{b_i, i \in I(i)} P(b_i, b_{i-1}, b_{i-2}|X_{i-1}) \quad (5)$$

where $I(i)$ are the pixels in the 8-neighborhood of b_i.

Using Bayes rule the numerator of Equation 4 is given by (we use b_i^{i-2} as a shorthand notation for (b_i, b_{i-1}, b_{i-2})):

$$P(b_i^{i-2}|X_{i-1}) = \frac{p(X_{i-1}|b_i^{i-2})P(b_i^{i-2})}{p(X_{i-1})}$$

We assume that all grayscale values X_{i-1} and all boundary segments (b_i, b_{i-1}, b_{i-2}) are equiprobable. The second assumption could be relaxed to prefer smoother boundaries although we have not pursuedthis here. To compute $P(X_{i-1}|b_i, b_{i-1}, b_{i-2})$ we assume that the grayscales on the two sides of the boundary in the neighborhood of b_{i-1}, X_{i-1}^L and X_{i-1}^R, are independent. Thus:

$$p(X_{i-1}|b_i, b_{i-1}, b_{i-2}) = p(X_{i-1}^L|M_L)p(X_{i-1}^R|M_R)$$

Furthermore, assuming independence of the grayscale of the pixels for each side we have:

$$p(X_L|L) = \prod_{j \in L} p(x_j|M_L)$$

The above equations are similar for the R side.

Using the 8-neighborhood to compute the conditional probabilities implies second order Markov dependencies and not first as we have assumed in Equation 3. For computational efficiency, we approximate Equation 4, so that the probability of a pixel to be on the boundary depends only on the previous one. This is incorporated in the optimization method used to maximize Equation 3.

4.2. Model Estimation

The statistical model we employ to describe the grayscale of the pixels in the vicinity of the boundary is that of the mixture of Gaussians. A Gaussian density is a good model to describe an image locally, so a small part of the boundary can be modelled well by a mixture of two Gaussians (one Gaussian for each side). However, we need to estimate a large part of the boundary and the grayscale of the pixels typically is not homogeneous. Thus, it is necessary to represent each side of the boundary with a mixture of Gaussians. The density of a grayscale value x is given by:

$$p(x|i) = \sum_{j=1}^{c_i} p(x|j, i)P(j|i) \quad (6)$$

where

$$p(x|j, i) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{\frac{-(x-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

i denotes the side of the boundary L or R, j indexes the components of the mixture and c_i is the number of components for each side i. This number of components is not known apriori and needs to be estimated during the construction of the model.

We use the maximum likelihood criterion penalized by MDL to estimate the model parameters. The likelihood of the grayscale is given by

$$P(X|M) = P(X_L|M_L) \cdot P(X_R|M_R)$$

The penalized log likelihood criterion using MDL is given by:

$$\log P(X_L|M_L) + \log P(X_R|M_R) - \frac{m_C}{2} \log M \quad (7)$$

where M is the total number of pixels of both sides of the boundary and m_C is the number of model parameters for a mixture of $C = C_L + C_R$ components. For one dimensional data $m_C = (C - 2) + 2C$.

5. Optimization

Boundary estimation requires the statistical model of the grayscale so that the probabilities 3 can be evaluated. On the other hand the boundary must be estimated so that the model parameters can be estimated, since in Equation 7 X_L and X_R must be known. Furthermore, N in Equation 3 is usually large (several hundrends) so exhaustive exploration of possible paths is impossible. For this reason we have used a greedy algorithm in which we alternate between boundary estimation given the model estimated so far and model estimation given the boundary estimated so far. The boundary is extended by the n 'best' pixels in the maximum likelihood sense, where $n << N$.

For the estimation of the n best boundary pixels we observe that the log of Equation 3 is a summation

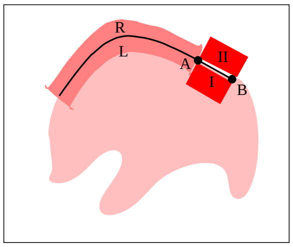

Figure 6. Mixtures have been estimated up to point A. The new boundary found is the segment AB. Our method proceeds by determining whether regions I and II should become new clusters, or they should be merged to the existing ones.

of terms each of which depends only on the previous one. We can take this observation into account and use a shortest path algorithm to optimize Equation 3. We proceed as follows. Let b be the last point of the estimated boundary as illustrated in Figure 5(a). We consider a square window with center b and side length $2n+1$. Note that all paths of length n will be contained in this window. We compute the maximum probability paths originating from b and ending to each pixel in the window. This can be done using Dijkstra's shortest path algorithm (Cormen et al., 1990), only in this case we are constructing maximum cost paths [1]. At the same time we can maintain the length of the paths constructed. At the end we choose the path with n pixels that has the highest probability.

This method does not guarantee the global optimum; it will find the best *shortest* path of n points, which will provide good solutions in this case. However, we could devise a dynamic programming approach that would find the global optimum but it would be computationally expensive.

The standard method used to estimate the parameter values of a mixture is the Expectation Maximization (EM) algorithm (Dempster et al., 1977). For computational efficiency we have used k-means (Duda et al., 2001). After k-means converges we compute each $P(j|i)$ equal to the percentage of the points that the corresponding cluster contains.

An important issue with clustering is the initialization. k-means requires an initial estimate of the means of the clusters as well as the number of clusters. These estimates are provided as we discover more boundary. The process is illustrated in Figure 6.

More specifically, suppose that we have estimated the statistical model for each side up to point A and the new boundary found is the segment AB. Let k_L and k_R be the number of components of the mixtures of the L and R sides respectively. For each side, we consider whether the number of components should be increased by one or not. This involves invoking k-means for four cases: side L should be modeled with k_L+1 components and side R with k_R; side R should be modeled with k_R+1 components and side L with k_L; the number of components should increase by one for both L and R; the number of components should stay the same for both L and R. In the cases where the number of components increases, the initial estimates of the means of the new components are the means of the regions I and II as it is illustrated in Figure 6. Furthermore, the most similar components of the two sides are tried to be merged until no more changes take place.

We have not discussed two issues so far. The first is related to what data we use to update the classifier and the second is related to what happens when the user provides additional input. Regarding the first issue, after we estimate a small part of the boundary we select the regions on its two sides that maximize the separability criterion 1 (similarly to the initialization of the method). The rationale for that is that we should include in our model data that are as well separated as possible. Regarding the additional user input we have considered two choices. Either we continue incrementing the existing model or we start a new model. The rationale for the first idea is that keeping all the data would lead to more reliable estimates. The rationale for the second is that the user clicked on some part of the boundary whose statistics are totally different from the statistics of the boundary estimated so far (that's why it was not estimated). We decide which alternative to follow by extending the boundary for a number of pixels using both strategies and choosing the one that estimated a boundary with greater separability between its two sides. (We have also tried weighting the most recent human input data more but that yielded inferior results).

6. Experimental Evaluation

We believe that two measures can sufficiently evaluate an interactive method like ours. The first is the number of mouse clicks required to delineate a boundary. This measure demonstrates the effectiveness of a method from the standpoint of the burden placed on the human user. Clearly, the fewer the number of clicks the more effective the method. The second measure is the difference of estimated contours for the same object starting from different image locations.

[1] This does not cause problem with cycles since the log likelihoods are negative.

This measure quantifies the consistency and stability of a method in delineating a boundary. It is a moot issue to evaluate the correctness of an interactive approach; since the user is an integral part of the system, the contour produced will always delineate the correct object.

Using these two measures we evaluate the need for semi-supervised modeling by comparing two methods. The first - supervised - uses only the human supplied data to extract the boundary. These data are modeled by a mixture of two Gaussian densities. The second - semi-supervised - uses the unlabeled data as well using a mixture of mixtures model described earlier. In both methods the boundary was incremented by 10 pixels at a time. We have experimentally determined that this was a reasonable value for both methods. Our results were not significantly different if the boundary was incremented by 20 pixels at a time. For less than 10 pixels the performance of the methods degraded.

The results of our study are shown in Table 1. Our data set consists of HRCT liver images, 25 slices obtained from 14 different patients. The criteria we used to select the slices were: the slices were not be consecutive, there wasn't any ambiguity with respect to the liver boundary, and the slices were neither the top nor the bottom slices of a scan (these are too easy and they do not shed any light as to the usefulness of the method). In each of these slices the liver boundary was extracted starting from 3 different boundary locations that were the same in all three methods. The consistency of the method was measured by computing the average shortest Euclidean distance between the points of the boundaries that delineate the same object and subsequently averaging these means over all the images in the data set. These means are measured in pixels and are shown in the first column of Table 1. The average number of mouse clicks needed to delineate the liver is shown on the second column.

Clearly, the most effective method in terms of both human input and consistency is the semisupervised one. It is worth mentioning that in images with large homogeneous regions both supervised and semi-supervised yielded comparable results. In more complex images semisupervised did better.

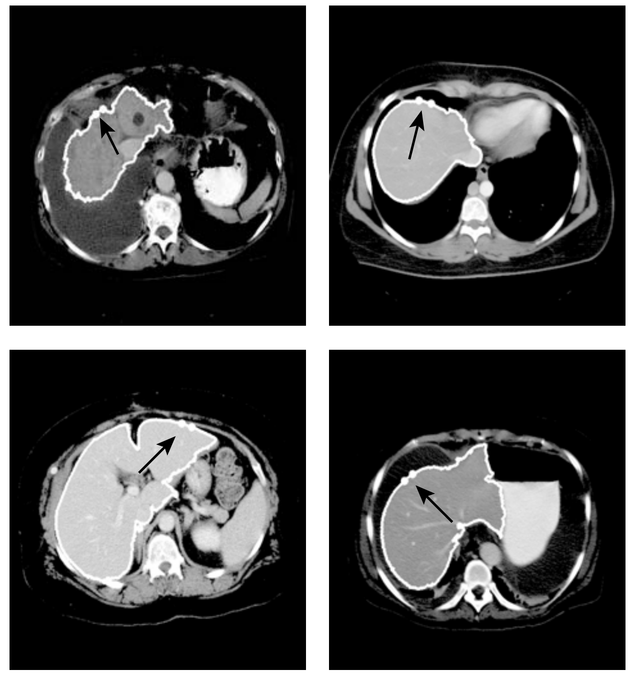

Figure 7. Boundaries obtained with our method. White circles pointed to by arrows indicate user clicked points. For visualization reasons we display results requiring only two clicks to delineate the entire boundary.

7. Conclusions and Future Work

We have presented a novel application domain for semisupervised and active learning techniques, that of interactive contour delineation. Our approach consisted of alternating between the steps of boundary extraction and model updating. We experimentally demonstrated that this is superior to supervised learning based on a small part of the boundary.

There are several research avenues for the future. It would be interesting to investigate a local approach instead of the mixture one. This could capture more spatial dependencies and could handle additional user input more naturally. Although we used an active learning type of interaction for eliciting training data we have not made use of any of the criteria employed in the active learning literature. These criteria could provide more disciplined ways to determine when to stop the estimation of the boundary. Finally, we extend the boundary in fixed-length increments. Is adaptive estimation of this increment feasible and useful?

Acknowledgements

This work was supported by the National Institute of Health under grant No. 1 R01 LM06543-01A1. We would like to thank the anonymous reviewers for their comments that improved the presentation of this paper.

Table 1. Consistency measured in pixels and average number of clicks required to delineate the outline of the liver in 25 HRCT slices.

Method	Consistency (in pixels)	Mean number of clicks
Semi-supervised	1.39	7.17
Supervised	1.52	10.32

References

Bennett, K., & Demiriz, A. (1998). Semi-Supervised Support Vector Machines. *Advances in Neural Information Processing Systems.*

Bensaid, A., Hall, L., Bezdec, J., & Clarke, L. (1996). Partially Supervised Clustering for Image Segmentation. *Pattern Recognition*, *29*, 859–871.

Boykov, Y., & Jolly, M. (2001). Interactive Graph Cuts for Optimal Boundary and Region Segmentation of objects in N-D images. *Proc. Int'l Conference on Computer Vision.*

Cormen, T., Leiserson, C., & Rivest, R. (1990). *Introduction to Algorithms.* McGraw Hill.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal Royal Statistical Society, Series B*, *39*, 1–38.

Duda, R., Hart, P., & Stork, D. (2001). *Pattern Classification.* Wiley-Interscience.

Falcao, A., Udupa, J., Samarasekera, S., & Sharma, S. (1998). User-Steered Image Segmentation Paradigms: Live Wire and Live Lane. *Graphical Models and Image Processing*, *60*, 233–260.

Figueiredo, M., Leitao, J., & Jain, A. (2000). Unsupervised Contour Representation and Estimation Using B-Splines and a Minimum Description Length Criterion. *IEEE Transactions on Image Processing*, *9*, 1075–1087.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition.* Academic Press.

Ghahramani, Z., & Jordan, M. (1994). Supervised Learning from Incomplete Data via and EM Approach. *Advances in Neural Information Processing Systems 6.*

Kass, M., Witkin, A., & Terzopoulos, D. (1987). Snakes: Active Contour Models. *Int'l Conference on Computer Vision.*

Mortensen, E., & Barrett, W. (1998). Interactive Segmentation with Intelligent Scissors. *Graphical Models and Image Processing*, *60.*

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text Classification from Labeled and Unlabeled Documents Using EM. *Machine Learning*, *39*, 103–134.

Paragios, N., & Deriche, R. (2002). Geodesic Active Contours and Level Set Methods for Supervised Texture Segmentation. *International Journal On Computer Vision*, *46*, 223–247.

Velthuizen, M. V. R., Venugopal, P., Clarke, L., & Hall, L. (1994). Tumor Volume Measurements Using Supervised and Semisupervised MRI Segmentation. *Artificial Neural Networks in Engineering.*

Xu, N., Bansal, R., & Ahuja, N. (2003). Object Segmentation Using Graph Cuts Based Active Contours. *Proc. Computer Vision and Pattern Recognition.*

Zhu, S. C., & Yuille, A. (1996). Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*, 884–900.