Shape-based models for interactive segmentation of medical images

Kevin P. Hinshaw*, Russ B. Altman†, and James F. Brinkley‡

*Dept. of Computer Science and Engineering, FR–35, University of Washington, Seattle, WA 98195
†Stanford University School of Medicine, MSOB X265, Stanford, CA 94305–5479
‡Dept. of Biological Structure, SM–20, University of Washington, Seattle, WA 98195

ABSTRACT

Accurate image segmentation is one of the key problems in computer vision. In domains such as radiation treatment planning, dosimetrists must manually trace the outlines of a few critical structures on large numbers of images. Considerable similarity can be seen in the shape of these regions, both between adjacent slices in a particular patient and across the spectrum of patients. Consequently we should be able to model this similarity and use it to assist in the process of segmentation. Previous work has demonstrated that a constraint-based 2D radial model can capture generic shape information for certain shape classes, and can reduce user interaction by a factor of three over purely manual segmentation. Additional simulation studies have shown that a probabilistic version of the model has the potential to further reduce user interaction. This paper describes an implementation of both models in a general-purpose imaging and graphics framework and compares the usefulness of the models on several shape classes.

Keywords: interactive 2D image segmentation, probabilistic models, constraint-based models, knowledge-based medical imaging, 2D shape analysis

1. INTRODUCTION

1.1 Motivation

Accurate image segmentation is one of the key problems in computer vision. Before high-level reasoning can be applied to an image, it must be broken down into its major structural components. For example, in radiation treatment planning, radiologists need to compute the best path for applying radiation to a tumor while avoiding critical structures such as the eye and the kidney. To accomplish this task, they acquire a set of CT or MR images of the patient, manually trace the outlines of the critical structures on each image, and use the resulting contours to build 3D models. If the image segmentation phase could be partially automated, the time required for radiation treatment planning would be reduced dramatically.

As a first step towards segmentation, images are commonly processed to enhance edges. However, standard gradient-based edge detection techniques work well only when images contain clear intensity differences at boundaries. Unfortunately, organ boundaries in CT and MR images are often difficult to detect with such techniques because adjacent soft tissues with similar densities get mapped to nearly equal intensity values. In these types of images, boundaries may be clearly defined in certain regions, but there are often large sections where they are not.

A method for identifying region boundaries which does not depend solely on intensity gradient is needed. People are able to do this task; they can "fill in the blanks" mentally in low-contrast regions by using visual cues from nearby edges and knowledge of an object's shape. If an image segmentation system is provided with the right kind of information, it may be able to do the same. Using a shape model to assist in segmentation may make it possible to guide the search for edges which are clear and infer the location of the boundary in regions where the image contrast is too low. Good shape models could enable the system to identify structures from just a few strong edge points.

Geometric constraint networks have been proposed as a flexible, generic model for representing shapes of biological structures.[4] Such a network consists of a set of physical objects, a set of possible locations in space for each of those objects, and a set of constraints specifying which object locations are compatible with each other. Later work presented a specialized, 2D version of geometric constraint networks called a *radial contour model*, where the "physical objects" are discrete points along the boundary of a contour.[5] That paper presented one approach to constraining points (which we shall call the *min/max model*) that was shown to be useful for guiding segmentation. More recently we proposed an alternative constraint representation

called the *probabilistic model*.[2] Experiments showed that this approach also has potential, but the results were based on simulations rather than actual test data.

This paper presents *Scanner*, a system which uses shape models to assist in image segmentation. Since fully automatic segmentation techniques are still unreliable, this system has been designed to work interactively. Scanner uses shape models to guide an edge detector in searching for structures. The resulting segmentations can be edited by the user to correct mistakes. Currently, both the min/max and probabilistic models are available in the system. This paper will describe the results of using these models to segment a large number of CT images, thus giving the first real data on the merits of the probabilistic model as a segmentation tool.

1.2 Related work

Many approaches to the image segmentation problem have been proposed. For example, Kobashi and Shapiro describe a dynamic thresholding approach to CT image segmentation that is driven by anatomical knowledge.[8] The system uses organ features such as expected location, size, and gray tone level to decide which of a series of thresholded versions of the image gives the best segmentation of the organ. Progressive landmarking is also incorporated so that organs found with high certainty can be used to help locate others.

Kass *et al*. first introduced a popular deformable model known as a *snake*, which is a 2D contour that reacts to applied forces.[7] The equations that determine the shape of a snake are taken from the literature on dynamics of physical systems. In its simplest form, a snake will tend toward a shape that minimizes bending or stretching energy, but it will also be attracted to strong edges in an image. Computing a snake's shape becomes an optimization problem — a contour must be found which follows regions of high intensity gradient but stays as "smooth" as possible. This type of model is very good at segmenting regions when large portions of their boundaries have clear edges. However, it depends heavily on being initialized with a contour which is close to the actual boundary, because the contour will usually stick to the first strong edge it finds

Since their introduction, physically-based models such as snakes have been widely used. Szeliski and Terzopoulos present a probabilistic model which uses physically-based contours as its source of prior information.[9] They also use the Kalman filter to improve the accuracy of the model as the contour evolves over time. This approach is well-suited for finding contours in time-varying image sequences, but it still depends on getting a good starting point.

1.3 Paper overview

This paper examines the usefulness of the min/max and probabilistic models as methods for performing semiautomatic image segmentation. Section 2 explains the models and describes Scanner, an application which implements the models within a general purpose imaging and graphics framework. Section 3 outlines the approach that was used to evaluate the performance of these two models. Section 4 presents the results of the evaluation and discusses the usefulness of these shape models as a means of semiautomatic image segmentation. Finally, Section 5 summarizes our findings and suggests improvements and additions to the current models.

2. METHODS

2.1 Radial contours and models

The particular contour representation used in this work is called a *radial contour.* Each contour is a closed polygon stored in polar coordinates with the parameterization $r(\theta)$. In other words, the points on the contour can be thought of as the tips of spokes, or *radials*, emanating from a point at the center of the contour. A local coordinate system is associated with each radial contour. (See Figure 1.) Whenever a contour is created, it is initialized with two points which define its *long axis*. These two points can be chosen arbitrarily, but they usually correspond to extremal points of the shape. The midpoint of the axis specifies the contour's center point and is used as the origin of the local coordinate system. The axis itself determines the orientation of the local vertical dimension.

In the current implementation, we have made some simplifying assumptions. Radials are restricted to be evenly spaced, so that the angle between any two adjacent radials in a contour is the same. All of the testing reported here uses 24 radials spaced 15° apart. We also limit a contour to have exactly one length r_i associated with each radial R_i. As a result, the connectivity of the polygon is implicit, with its edges formed by proceeding sequentially along the endpoints of the radials. These simplifications make it easier to measure aspects of a contour's shape at the cost of imposing limits on the shapes the model can repre-

sent. However, the resulting representation is still flexible enough to handle most of the organs that need to be identified in a domain like radiation treatment planning.

By gathering data from a collection of similarly-shaped radial contours, we can build a *radial contour model* that describes them. The intent is that this model will be able to guide the search for structures which share this shape. During segmentation, data acquired from the image (*e.g.* edge information) can be used to focus attention on the range of shapes in the generic model which are also consistent with the contour in the image.

2.2 Min/max model

The min/max model contains two types of shape information. First, it associates an uncertainty interval $[I_i,O_i]$ with each radial R_i, such that $I_i \le r_i \le O_i$. Taken together, the I_i's define an inner uncertainty contour, and likewise the O_i's define an outer uncertainty contour; any contour modeled by a particular shape model is expected to lie in the area between these two contours. A best guess at the contour's actual location is chosen as the contour with vertices $B_i = \frac{1}{2}(I_i + O_i)$. (See Figure 2.) Second, the model stores an n-by-n matrix of shape constraints, where n is the number of radials in the model. Let $s_{ij} = r_i/r_j$, the ratio of the lengths of R_i and R_j. The (i,j) entry of the matrix specifies a lower bound, L_{ij}, and an upper bound, U_{ij}, for s_{ij}.

Given a set of training contours, computation of the model's shape constraint matrix is straightforward. In each contour, we compute s_{ij} for each pair of radials R_i and R_j. We then calculate the (i,j) entry of the matrix by setting L_{ij} to be the minimum value of s_{ij} seen in the training set and setting U_{ij} to the maximum value.

Once a model has been created, it can be instantiated as a radial contour by specifying the endpoints of a long axis. When a contour is created in this way, its long axis provides the exact lengths for the radials at 90° and 270°. These two values are used in conjunction with the model's constraint matrix to compute the uncertainty intervals for all of the other radials in the contour. Given a radial length r_j and the restriction $L_{ij} \le s_{ij} \le U_{ij}$, we can infer that $L_{ij} r_j \le r_i \le U_{ij} r_j$ for every other radial R_i. If these bounds are more restrictive than R_i's previous uncertainty interval $[I_i,O_i]$ then the interval is tightened. This reduction may in turn shrink the uncertainty for other radials, so the process repeats until no more uncertainty intervals change. During segmentation, this process of constraint propagation is reapplied each time a new radial length is found.

2.3 Probabilistic model

As its name implies, the second type of model under consideration uses probabilistic techniques to quantify shape. For each radial R_i, the model stores a continuous probability distribution with mean $E(r_i)$ and variance σ_i^2. These two values provide the edge detector with a "best guess" starting point when looking for an edge, along with an indication of how far from this guess the detector may need to look to find the correct edge. As with the min/max model, geometric constraints are based on relations of radial lengths. But instead of using length ratios to constrain r_i and r_j, the probabilistic model stores the covariation between them.

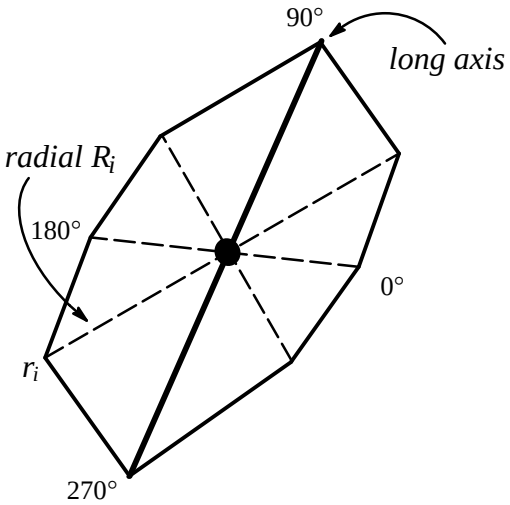

Figure 1. An 8-Radial Contour

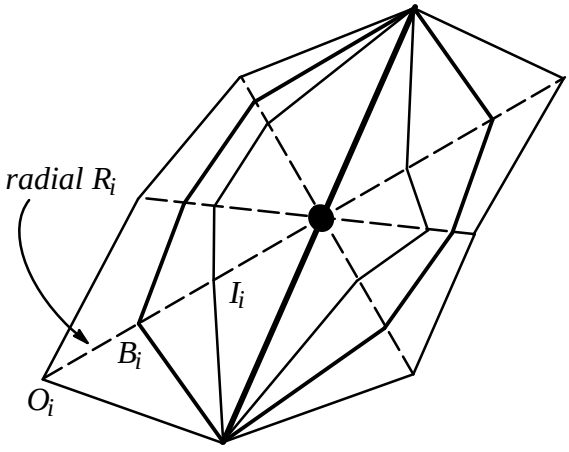

Figure 2. An 8-Radial Contour with Uncertainty Intervals

Given a training set of n-radial contours, we can think of the radial lengths as a vector of random variables, $x = [r_1\ r_2\ \cdots\ r_n]$. We then construct two components for the probabilistic model as follows:

1. Build a state vector storing the mean values for each radial's distribution:

$$M(x) = [E(r_1)\ E(r_2)\ \ldots\ E(r_n)] \tag{1}$$

2. Build a variance/covariance matrix:

$$C(x) = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2 & \ldots & \sigma_1\sigma_n \\ \sigma_2\sigma_1 & \sigma_2^2 & \ldots & \sigma_2\sigma_n \\ \vdots & \vdots & . & \vdots \\ \sigma_n\sigma_1 & \sigma_n\sigma_2 & \ldots & \sigma_n^2 \end{bmatrix} \tag{2}$$

where:

$\sigma_i^2 = E(r_i^2) - E(r_i)^2$ defines the variance for r_i and

$\sigma_i\sigma_j = E(r_i r_j) - E(r_i)E(r_j)$ defines the covariance for r_i and r_j.

One nice feature of the ratio constraints used in the min/max model is that they are scale invariant — they depend on relative lengths of radials rather than absolute lengths. To maintain this property when building a probabilistic model, each contour in the training set is normalized relative to its long axis. When a probabilistic model is instantiated as a contour, the length of the long axis determines the proper scale and is used to compute the proper mean, variance, and covariance values.

Once a contour has been created from a probabilistic model, laws of conditional probability are used to update the individual radial distributions as data about radial lengths becomes available. Specifically, we use a Bayesian technique known as the Kalman filter[6] to combine prior and measured distributions to get an updated distribution. In this case, the mean and variance for r_i constitute the prior distribution. The measured distribution consists of the radial length, which is specified by the user or an edge detector, and a variance value which estimates the accuracy of that selection. If the length is user-specified, it is assumed to be exact and the variance is set to be almost zero; if it was selected by an edge detector, the variance value is chosen based on the noise inherent in the edge detector.

2.4 Scanner

The min/max and probabilistic models have been incorporated into an interactive segmentation system called Scanner. Originally developed for NeXT computers,[5] the application was redesigned to run on Silicon Graphics machines. This change of hosts was motivated mainly by two factors: insufficient support on the NeXT for intensive image and graphics manipulation, and the inability to move applications written on the NeXT to other platforms. The first problem was solved by the SGI hardware, which is geared toward high performance graphics. To solve the second problem, the new version of Scanner was implemented as a module in Skandha, a 3D graphics server written by Jeff Prothero of the University of Washington's Department of Biological Structure. Skandha is built on top of Slisp, a hybrid Lisp–C programming toolkit we are developing. Slisp is based on Xlisp, a Lisp interpreter written in portable C code.[3] Primarily, Skandha extends Slisp by adding primitives which facilitate graphics operations. With the exception of one machine-dependent module responsible for generating graphics display calls, Skandha is written in a combination of Lisp and C with the intent that it can be ported to other machines.

Several members of the Department of Biological Structure currently use Skandha to build 3D reconstructions from stacks of segmented images. At the moment though, all segmentation is done by hand. By building Scanner within the Skandha framework, we now have the ability to assist in the production of these animations by providing some model-based segmentation tools. Another advantage of using Skandha is that its hybrid nature makes it easy to construct prototypes quickly, but still allows for speed when necessary. With the graphics primitives available, it does not take long to write a Skandha application at the Lisp level, complete with graphical user interface, for doing 3D graphics manipulation. Once an initial design is completed, computationally intensive routines can be optimized by rewriting them in C and calling them as primitives from the existing Lisp code.

Figure 3 shows the interface for Scanner. Menus at the far right of the window allow the user to manipulate models, contours, and images. The main graphics window shows an image and a radial contour of a kidney. Toggle switches are provided to control the display of the radials, best guess contour, and inner and outer uncertainty contours. In addition, the user can set edge detector parameters, such as which edge should be used when several are found along a radial and the order in which radials should be searched.

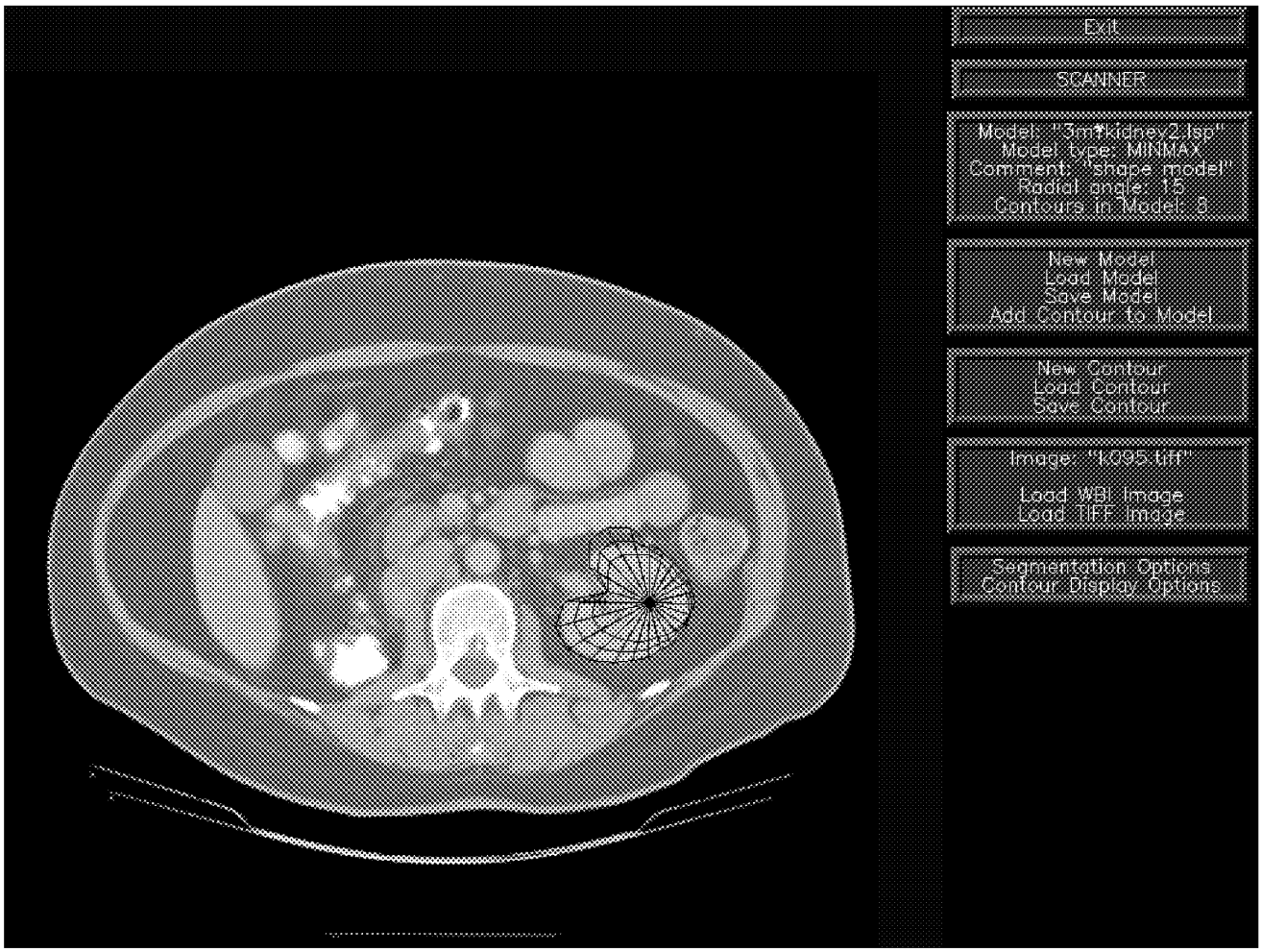

Figure 3. Scanner

To segment an image with an existing shape model, the user loads the appropriate image and a model into Scanner. The user then draws the long axis of the structure on the image to specify the contour's location and local coordinate system. Once the contour has been placed, Scanner uses a 1D edge detector to scan the uncertainty interval of each radial. Each time an edge is found, it is incorporated into the model by applying constraint propagation (for min/max models) or the Kalman filter (for probabilistic models). When the system finishes the segmentation, the user can correct errors by clicking on a radial and dragging it to the proper length. Contours can then be saved so that they may be processed by higher-level vision routines.

Scanner can also be used to build contour models for later use. By turning off constraint propagation and edge detection, the user can draw radial contours on an image manually. A new model (min/max or probabilistic) can be created, and contours can be added to it as they are drawn. The resulting model can then be saved so that it may be used to segment similar contours in the future.

3. EVALUATION

Fifteen biological shape classes were used for testing the two types of radial contour models. The classes were cross-sections of the following: eye, rib, lung, spinal cord, two portions of the spleen, three liver shapes, three levels of kidney, and three shapes of vertebrae. We chose shapes that (1) can be represented by the radial contour model, (2) are critical structures that are frequently segmented for radiation treatment planning, and/or (3) display interesting geometric properties. Since the models

are two-dimensional, they can be used to describe only cross-sections of three-dimensional structures. Consequently, for several organs multiple shape classes were used to capture shape information at particular locations along the object's axis.

We have hypothesized that a useful segmentation system must allow the user to make corrections interactively. To make Scanner as useful as possible, the amount of work that needs to be done by hand should be minimized. Consequently, the "usefulness" of these two models is measured by counting the number of radials the user needs to specify per contour (the lower this value, the higher the utility). This count includes the two endpoints of the contour's long axis and any corrections the user would make to the results returned by the model-based segmentation system. We assume that in a segmented contour, radial R_i would be corrected if $|r_i - r|$ exceeds some arbitrary threshold, where r is the distance along R_i to the actual boundary location. The tests described in this paper used a threshold of 4 pixels.

For each of the 15 shape classes, 16 CT images were selected from an image archive.* The 240 images were partitioned into two trial sets, each containing 8 images per shape class. The structure on each image was segmented by hand using a 24-radial contour, and the results were stored in a database.† This set of hand-drawn contours provided both data from which to build test models and *ground truth contours* which could be used to measure the accuracy of the semiautomatic segmentations achieved with Scanner.

For each of the 15 shape classes, the corresponding contours from the first trial set were used to construct a min/max model and a probabilistic model. Next, each image in the second trial set was segmented with the appropriate model from the first trial set. Segmentation was initialized with the long axis from the ground truth contour for that image. After each segmentation, the result was compared to the ground truth contour and the number of radials needing correction was counted. For comparison purposes, each image was also segmented a third time, using a *control model* which contained no shape information. This provided data on how well the edge detector could do when it had no knowledge about the structure being segmented. This entire process was then repeated by switching the trial sets — models were built from contours in the second trial set and used to segment the images in the first trial set.

For all of the tests reported here, radials were processed in order of increasing uncertainty interval size. If multiple edges were found for a single radial, the edge detector returned the one closest to the center of the contour.

4. RESULTS AND DISCUSSION

4.1 Comparison of models

Two features make the probabilistic method potentially more powerful than the min/max model. The first difference involves how a radial's length is set. In the min/max model, a radial's length stays fixed once a value has been assigned to it. But in the probabilistic model, setting a radial's "length" actually corresponds to specifying a distribution (generally a very narrow one) where the edge is expected to lie. Unless that distribution has no variation, this means that future iterations of the Kalman filter can still affect the radial's length. When the system is using a 1D edge detector to generate edge locations, it inevitably will make mistakes. If the shape model being used is a good one, this additional flexibility in the probabilistic model may make it possible to overcome these errors in edge detection.

Second, the probabilistic model does not have a fixed region along each radial in which an edge is expected to be found. Consequently, any point along a radial can be used to tighten the distributions for other radials. This is not the case with the min/max model, which is forced to stop all constraint propagation if r_i is ever set outside of the interval $[I_i,O_i]$. (The current implementation does not take advantage of this feature of the probabilistic model. Instead of searching along the entire length of each radial R_i, it computes a search region $[I_i,O_i]$, choosing $I_i = r_i - k\sigma_i$ and $O_i = r_i + k\sigma_i$ for some constant k. This region can then be searched for edges as with the min/max model.)

One drawback of the probabilistic model is that its order of complexity is higher than that of the min/max model. Previous work has demonstrated that in the min/max model, each constraint propagation operation has complexity $O(n)$, where n is the number of radials.[5] With the probabilistic model, updating the variance/covariance matrix requires at least one dense matrix-vector multiplication each time an edge is found, yielding a complexity of $O(n^2)$. This means that as the number of radials used in a model increases, the probabilistic model will become increasingly slower relative to the min/max model.

---

* This archive was created by the University of Washington's Department of Electrical Engineering in collaboration with the Department of Radiology.

† These contours were drawn by one of the authors to evaluate the performance of the NeXT version of Scanner.

4.2 Test results

Figure 4 summarizes the performance of the control, min/max, and probabilistic models for the complete set of trials. This data corresponds to running the experiments with an edge threshold of 10%,* assuming that detected edges are specified with a variance of 1.0 square pixels, and searching within 2 standard deviations of the mean for each radial in the probabilistic model. For each model type measurements were computed from 240 trials. The overall results suggest that shape knowledge does indeed lead to better segmentation, since the control model required more user-defined radials than either the min/max or probabilistic model. The difference in user-defined radials for the min/max and probabilistic models is not statistically significant, but the difference between these and the control model is significant at $p < 0.01$.

	User-defined Radials		Time	
Model Type	mean	σ	mean (sec)	σ
Manual†	24.0	0.0	27.9	7.4
Control	11.7	6.6	9.9	1.1
Min/Max	8.7	6.4	20.8	5.8
Probabilistic	7.9	6.3	166.2	47.4

Figure 4. Overall Performance

A breakdown of how the models performed on each shape is shown in Figure 5. This table illustrates that shape knowledge is much more effective on some structures than others. For the eye, spinal cord, ribs, and vertebra, the model performs extremely well, making fewer than two errors on average. The models were less effective on the kidney and spleen, although they still did a reasonably good job. However, they had a difficult time segmenting the liver and lungs. The poor results on the liver stem from the high variability in the liver models. When radial uncertainty intervals are too large, the chances of the edge detector selecting the wrong radial length are much greater. With the lung, the opposite problem occurred. The lung models had so little variation that during testing, the boundaries in the images fell outside the uncertainty intervals. Consequently, the edge detector had no chance of finding the proper boundary.

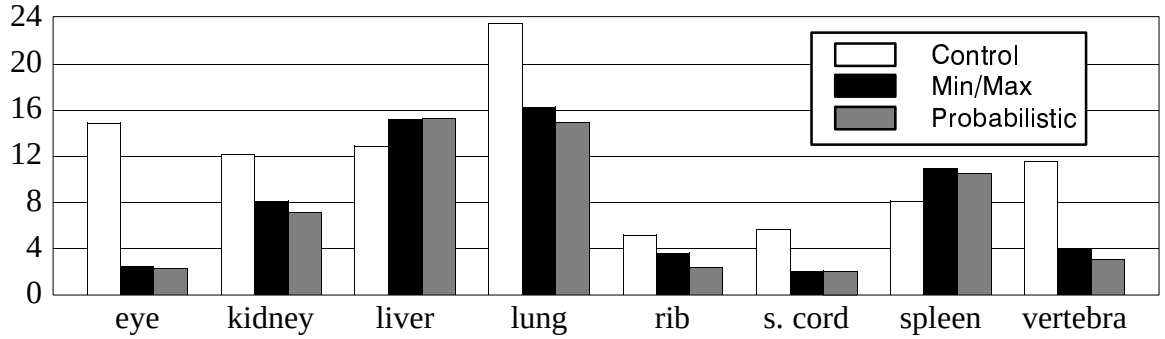

Figure 5. User-Defined Radials by Shape

While the timings shown in Figure 4 for the control and min/max models are comparable to manual segmentation time, the speed of the probabilistic model is clearly inadequate for an interactive segmentation system. It should be noted that the slow performance results primarily from the decision to implement most of both models at the Lisp level — including the computationally intensive portions of the constraint propagation and Kalman filter routines. Rewriting these portions of the models in C would improve significantly the performance of both the probabilistic and min/max models. In addition, the code implementing the probabilistic model could be improved further by applying sparse matrix methods.

---

* This threshold was chosen because it gave the best overall results for all three models. It is important to note that the threshold had only minor impact on the performance of min/max and probabilistic models, but it dramatically affected the results for the control model. Consequently, the data in Figure 4 shows less improvement over the control model than was reported in earlier work, which used a threshold of 5% rather than 10%.[5]

† The time for drawing 40 of the 240 contours was used to compute the timing results for manual segmentation.

To test the benefit gained from starting with just a rough guess at a contour's shape, the complete set of experiments was repeated. This time, however, edge data was used only to set radial lengths during segmentation; it was *not* used to incrementally update the models. In other words, the initial uncertainty intervals in the models were used but were not tightened as edges were found. Figure 6 shows the rather surprising results: these segmentations were as good as or better than those achieved previously, and they took significantly less time to compute. Figure 7 shows the performance on a shape-by-shape basis. These results demonstrate that simple shape knowledge can greatly improve manual segmentation, in terms of both speed and the required amount of user interaction.

	User-defined Radials		Time	
Model Type	mean	σ	mean (sec)	σ
Min/Max	7.6	5.5	8.3	0.9
Probabilistic	6.9	5.0	5.4	0.7

Figure 6. Overall Performance Without Sequential Updates

To see why better results were not reached when edge data was added to the model during segmentation, we took a closer look at the experiments while they were running. In many cases, uncertainty intervals were nearly eliminated after a couple of edges were found, but the intervals were converging on non-boundary points. Consequently, the edge detector had no chance of finding the correct edges for radials which were considered late in the segmentation process. Two possible causes for this behavior are (1) insufficient variation in the test models, and (2) poor choices by the edge detector early in the segmentation process.

To test the first hypothesis, the testing strategy was modified. Rather than testing the models on an unseen dataset, they were used to segment images from their own training set. Since the data used to build the models came from these images, this experiment eliminated lack of model variation as a possible cause for poor segmentation. Figure 8 shows the performance of the models on the training set data as compared to the unseen data. The training set segmentations required slightly fewer user-defined radials, but the experiments that did not incorporate edge data still performed as well as those that did. We can conclude that lack of variation in our test models was not the cause of poor model convergence in the first set of experiments.

The number of errors made when testing the models on their own training data offers additional support for the hypothesis that there are problems with the edge detection strategy. Further testing is needed to determine whether premature commitment to erroneous edges causes the shape models to diverge from the correct boundaries. The results seen here suggest that even with constrained search regions, errors will be made during edge detection, and folding this incorrect data back into the shape models causes them to diverge from the correct shape. It is likely that performance could be improved by using more sophisticated edge analysis. For example, considerable advantage could be gained if all possible edges for each radial were considered

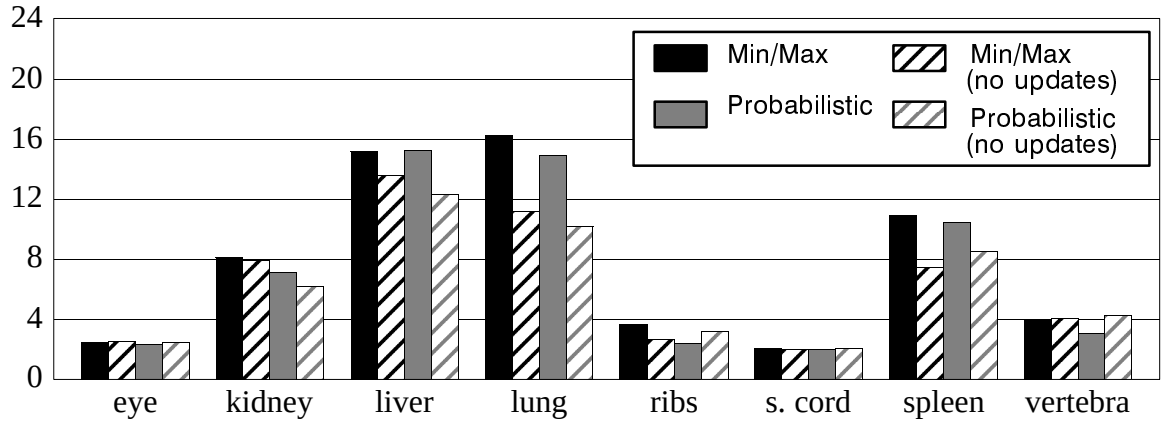

Figure 7. User-Defined Radials by Shape
(with and without sequential updates)

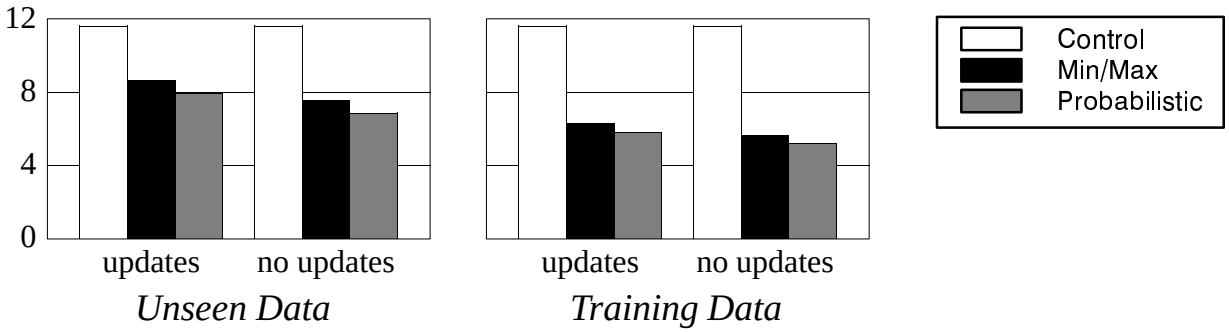

Figure 8. User-Defined Radials for Unseen Data vs. Training Set Data

simultaneously. This could be accomplished by incorporating a recent extension to the Kalman filter algorithm which uses constraints that are weighted mixtures of Gaussian distributions.[1] Each potential edge could be modeled as a Gaussian with a weight determined by the edge's strength in the image, and the algorithm could be used to choose the most likely set of edges.

5. CONCLUSION

We have discussed the benefits of using shape knowledge to facilitate image segmentation and compared the performance of two shape models. The results show that both models are effective methods for reducing segmentation time. When the min/max and probabilistic models were used without incorporating edge data, they required less than one-third the number of user-defined radials needed for manual segmentation, and they ran up to five times faster than manual methods. For some structures, the models were also significantly more effective than automatic methods which lacked shape knowledge.

The experiments also showed that for many shapes, erroneous data from the edge detector may be causing the shape models to diverge from the correct solution. By collecting other information, it may be possible to reach a better estimate of which edges are correct. These estimates could be used to determine the order in which radials should be added to the model, and in the case of the probabilistic model, the variance which should be used when the Kalman filter is applied.

In addition to the potential improvements mentioned already, there are a number of ways in which these models could be strengthened. As mentioned earlier, only the most likely portion of the radial is searched by the edge detector when the probabilistic model is used. Ideally, some method which examines the entire radial should be used so that the correct edge is never excluded from consideration. One possible approach that needs to be tested would use the Kalman filter to combine the radial's length distribution with a distribution describing the intensity gradient along the radial. Our hypothesis is that the mean value for the resulting distribution would give the best guess for the edge's location, and would provide more accurate information than the current technique. Another possibility would be to combine either the min/max or probabilistic model with snakes. The primary difficulty with snakes is finding a good starting point; perhaps a shape model could be used to get an estimate of a boundary's location and used to initialize a snake.

There are many paths that still need to be explored with these models. Hopefully the framework provided by Scanner will make it easy to implement new ideas quickly and decide which ones will be the most useful for performing interactive image segmentation.

6. ACKNOWLEDGMENTS

This work was sponsored by National Cancer Institute Grant R29CA59070, National Library of Medicine Grant RO1LM04925, and Achievement Rewards for College Scientists (ARCS). The research is part of the Digital Anatomist Program at the University of Washington, which is directed by Cornelius Rosse. Russ B. Altman is a Culpeper Medical Scholar, supported by LM–05652. Helpful suggestions were offered by Greg Heil, Loyd Myers, Jeff Prothero, and Ken Thornton. Thanks to T.J. Goan and Lara Lewis for their useful feedback on early drafts of this paper.

7. REFERENCES

1. Altman, R. B., C. C. Chen, W. B. Poland, J. P. Singh, "Probabilistic constraint satisfaction with non-gaussian constraint noise," *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 15–22, August 1994.

2. Altman, R. B. and J. F. Brinkley, “Probabilistic Constraint Satisfaction with Structural Models: Application to Organ Modeling by Radial Contours,” *Proceedings of the Annual Symposium on Computer Applications in Medical Care*, pp. 492–6, 1993.

3. Betz, D., “Xlisp: An object-oriented Lisp,” unpublished reference manual for version 2.1, available via FTP from `ftp.biostr.washington.edu` and other Xlisp FTP sites, 1989.

4. Brinkley, J. F., “Hierarchical geometric constraint networks as a representation for spatial structural knowledge,” *Proceedings, 16th Annual Symposium on Computer Applications in Medical Care*, pp. 140–4, November 8–12, 1992.

5. Brinkley, J. F., “A Flexible, Generic Model for Anatomic Shape: Application to Interactive Two–Dimensional Medical Image Segmentation and Matching,” *Computers and Biomedical Research*. vol. 26, no. 2., pp. 121–42, April 1993.

6. Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, Mass., 1984.

7. Kass, M., A. Witkin, and D. Terzopoulos, “Snakes: Active Contour Models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, January 1988.

8. Kobashi, M. and L. Shapiro, “Knowledge-Based Matching for 3D Radiotherapy Planning,” *Proceedings. 11th IAPR International Conference on Pattern Recognition. Vol.1. Conference A: Computer Vision and Applications*, pp. 293–6, 1992.

9. Szeliski, R. and D. Terzopoulos, “Physically–Based and Probabilistic Models for Computer Vision,” *SPIE Geometric Methods in Computer Vision*, Vol. 1570, pp. 140–52, 1991.