COMPUTERS AND BIOMEDICAL RESEARCH 26, 121–142 (1993)

A Flexible, Generic Model for Anatomic Shape: Application to Interactive Two-Dimensional Medical Image Segmentation and Matching

JAMES F. BRINKLEY

Department of Biological Structure, University of Washington, Seattle, Washington 98195

Received April 16, 1992

A representation called a radial contour model (RCM) is described for two-dimensional anatomic shapes. The model, which is a type of a geometric constraint network (GCN), is both flexible, in that it can deform to fit a particular instance of an anatomic shape, and generic, in that it captures all examples of a particular anatomic shape class. The model is implemented in a program, called SCANNER (version 0.7), for interactive model-based two-dimensional image segmentation and matching. Use of the model allows the segmenter to direct the search for edges in the image, and to fill in edges where none are present. Evaluations were done using models of 15 cross-sectional shapes appearing on CT images from 16 patients. Results from 480 trials show that the model-based approach reduces segmentation time by nearly a factor of 3 over manual methods, and correctly classifies 72.9% of the contours. The results not only suggest that the RCM will be useful for several current medical image segmentation tasks, but also support the hypothesis that geometric constraint networks are a viable approach to anatomic shape representation.

1. INTRODUCTION

One of the most ubiquitous problems in medical image analysis is segmentation of important biological structures from the background. This problem arises frequently, because segmentation is a necessary step for any sort of processing other than enhancement of images for visualization. In the case of 3-D or 4-D (time-varying) data the segmentation problem is even more acute, because the amount of image data is much larger. Even 3-D display for human visualization may require segmentation, so that obscuring structures can be removed from the display (*1*).

Examples of medical problems that require segmentation are estimation of fetal weight from ultrasound (*2*), quantitation of left ventricular size and shape from ultrasound (*3*), classification of cells by shape (*4*), and reconstruction from serial sections for biomedical education (*5, 6*) or for radiation treatment planning (*7*). In most of these cases the segmentation problem is a major bottleneck to the wider use of automated image analysis procedures.

0010-4809/93 $5.00

The most common segmentation methods are simple edge following or region growing (*8*). Although these techniques work well for high-contrast structures such as bone, that are well separated from the background, they fail for many soft tissue objects where the contrast is less. The failure of these low-level techniques has led to the development of knowledge-based methods which attempt to compensate for image ambiguities by giving the computer some knowledge of the problem domain (*9–12*). One of the more important kinds of knowledge is spatial knowledge about anatomic shapes and range of variation, as well as the geometric relationships among anatomic objects (*13*). This type of knowledge has been used in several model-based medical image segmentation systems to direct low-level image processing operations as they search for anatomic objects in expected regions of the image (*14–18*). Such an approach is appealing for medical imaging because, unlike the general vision problem (*19*), the "world" that needs to be modelled is "limited" to human anatomy and pathology. However, the difficulty for biological, as opposed to the man-made objects in industrial vision, is how to represent biological variation.

The approach to segmentation advocated in this paper is based on two premises: (1) segmentation requires spatial knowledge about the shape as well as the variation among anatomic objects and (2) the segmentation problem in general is not likely to be completely solved in the near future, so any useful segmentation system must be interactive.

In a previous report a representation for anatomic objects was introduced which attempted to capture the expected shape *and* the range of variation for anatomic objects in a flexible, generic model (*17*). This representation is a special case of a more general representation called geometric constraint networks (GCNs) (*20, 21*), and was subsequently used as the basis for a system for protein structure determination from nuclear magnetic resonance data (*22*). Although the earlier anatomic shape representation was tested on balloon models imaged with ultrasound, it was never evaluated on real medical images, nor was it implemented in an interactive segmentation system.

The current paper describes a two-dimensional version of this representation (Section 2), as well as its implementation in an interactive computer system for two-dimensional medical image segmentation and matching (Section 3). The advantage of this approach to segmentation is that the variability in the model allows the segmenter to direct the search for edges in the image, and to fill in missing edges where none are present. Evaluations of this system (Section 4) support the hypotheses that (1) the representation is able to model the shape and range of variation for a variety of cross-sectional shapes (2) the interactive system is useful for image segmentation of structures that may not be well separated from the background, and (3) the representation is useful for shape matching. The results show that the current system will be useful for several 2-D image segmentation tasks such as radiation treatment planning or cardiac size and shape determination. They also provide justification for re-implementing the full 3-D model in an interactive system, and for further developing the GCN approach to spatial knowledge representation.

2. Anatomic Shape Representation

2.1. Requirements

For interactive knowledge-based medical image segmentation an ideal shape model would not only be *flexible* enough to accurately fit the data, but would also be *generic* in that it would capture the range of variation for all expected instances of a given shape class. Knowledge of variation would allow the model to predict regions on the image within which to search for the desired object. The model should allow the shape of an unknown object or image region to be classified by matching the object against a set of generic shape models, and the model should adjust itself quickly to image data, so it could be used interactively. Although many of the currently popular deformable models are accurate (*23–26*), they do not encode variation, so cannot easily be used to direct the search for edges. On the other hand, the generic models developed for man-made objects are not flexible enough to accurately fit the data (*27*).

2.2. The Radial Contour Model

The radial contour model (RCM) is a flexible, generic model that implicitly captures shape and range of variation for a class of two-dimensional shapes, in the form of geometric constraints. Although the model cannot capture all possible shapes, it demonstrates the utility of the GCN approach to spatial knowledge representation by showing that it is useful for interactive two-dimensional medical image segmentation and matching. Because the model adjusts quickly to new data, it should be more suitable for interactive use than similar but more complex models (*18*).

The RCM is a specialization of a GCN, which in turn is a specialization of a constraint network. A constraint network consists of a set of variables, a set of possible values for each variable, and a set of constraints that determine which of the possible values of the variables are compatible with each other (*28, 29*). A geometric constraint network (GCN) is a constraint network in which the variables represent physical objects, the possible values of the variables represent possible locations of those objects in space (also called the *accessible volumes*), and the constraints represent required geometric relationships between the objects (*20, 21*). In the case of the RCM, the variables represent points on the contour boundary, the possible values of the variables define the accessible volumes of the contour points with respect to a local contour coordinate system, and the constraints are essentially allowed slope ranges for the lines between neighboring radials.

Figure 1 shows the structure of an RCM. A set of n fixed radials R are shown emanating from the origin of a local contour coordinate system ($n = 8$ in the figure). The local coordinate system is defined by a *long axis*, specified by the user, which allows the contour to be arbitrarily oriented with respect to the image. The length of the long axis determines the overall size of a particular instantiation of the model, but the model shape is invariant with respect to scale.

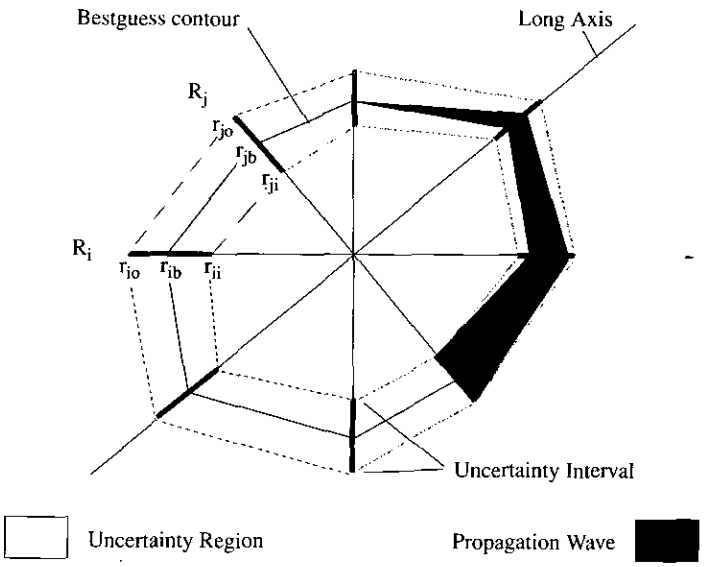

FIG. 1. A radial contour model. A contour boundary is defined by the distance r_i along each of a set of fixed radials R_i, emanating from a local coordinate system. The coordinate system is defined by a long-axis input by the user. Constraints between neighboring radials are learned from a training set. Edge information obtained at one radial propagates through the constraints to generate an uncertainty region that can be used to limit the search for additional edges.

The radials are spaced a fixed angle apart (45° in this example). The position of the contour boundary along each radial R_i is specified by a single scalar r_i representing the distance from the origin to the contour boundary along the radial. The set of possible distances along each radial is specified by a one-dimensional *uncertainty interval* $[r_{ii}\ r_{io}]$ along the corresponding radial. Straight lines $r_{ii}r_{ji}$ connecting the minimum values for each interval define an inner uncertainty contour, and straight lines $r_{io}r_{jo}$ connecting the maximum values define an outer uncertainty contour. The area between these two contours defines the *uncertainty region* (similar to a confidence region), within which the contour is always assumed to lie. The lines between the midpoints r_{ib} of each interval define a *bestguess contour,* which is taken to be the bestguess at any time as to the actual location of the contour in the image. The uncertainty interval $[r_{ii}\ r_{io}]$ for each radial R_i is the RCM equivalent of the *accessible volume* of an object in the general GCN model.

Figure 1 shows a *local* RCM, in which constraints are only specified between neighboring pairs of radials. That is, R_i is only constrained by R_{i-1} or R_{i+1} (modulo the number of radials). This is a 2-D version of the model used in the previous work (*17*); another type is a *maximal* radial contour model, in which each radial is constrained by every other. In general the constraint network represented by this model should require exponential time to solve because of

the greater number of constraints, but the constraining power should also be greater. However, for this application the maximal model is actually solved in linear time (that is, in time proportional to the number of radials rather than the square of the number of radials). Preliminary studies of the usefulness of these two model types for segmentation and matching showed that they are roughly similar, although the maximal model is slightly better. Therefore, all further evaluation will be with respect to the maximal model.

Each pairwise constraint in a binary constraint network specifies which of the possible values for the two variables are compatible. In the case of the RCM, each constraint specifies which values for each radial vertex pair are self-consistent. Constraints between two radials are unidirectional, although the constraint in the opposite direction can be inferred from a given constraint. Constraints are "learned" from a training set of similarly shaped radial contours (for example, a set of normal kidney cross-sections). For each member of the training set, and for each pair of radials R_iR_j connected by a constraint, let r_i be the measured distance from the origin to the contour boundary along radial R_i, let r_j be the measured distance from the origin to the contour boundary along radial R_j, and let s_{ij} be the ratio r_i/r_j. The constraint C_{ij} between radials R_i and R_j is expressed by an interval $C_{ij} = [L_{ij}\ U_{ij}]$, where L_{ij} is the minimum value of s_{ij} observed over all members of the training set, and U_{ij} is the maximum observed value for s_{ij}. For the local contour model, the range $[L_{ij}\ U_{ij}]$ may be thought of as a range of allowable slopes for the contour line between the connected radials. For the maximal model this analogy breaks down, unless each radial is thought of as being "near" every other radial, in a higher dimensional space.

3. Using the RCM for Interactive Segmentation and Matching

The RCM is implemented in a program, called SCANNER (version 0.7), that is designed to demonstrate and evaluate the utility of the model for interactive 2-D image segmentation and matching. SCANNER is implemented in Objective-C on the NeXT computer, and makes extensive use of the interface development tools available on the NeXT. The program is part of a general distributed framework that we are developing for medical image analysis. Details of this framework will be published elsewhere.

SCANNER consists of a series of Objective-C objects which communicate with each other via message passing. The user communicates with the program via the interface show in Fig. 2. The interface allows the program to be run in many ways in order to experiment with different segmentation techniques. As such, SCANNER is useful as a framework for the development of interactive medical image processing algorithms. For model-based segmentation the program is used in three main ways: (1) creation of an RCM from a training set, (2) using the RCM for interactive segmentation and (3) using the RCM for matching.

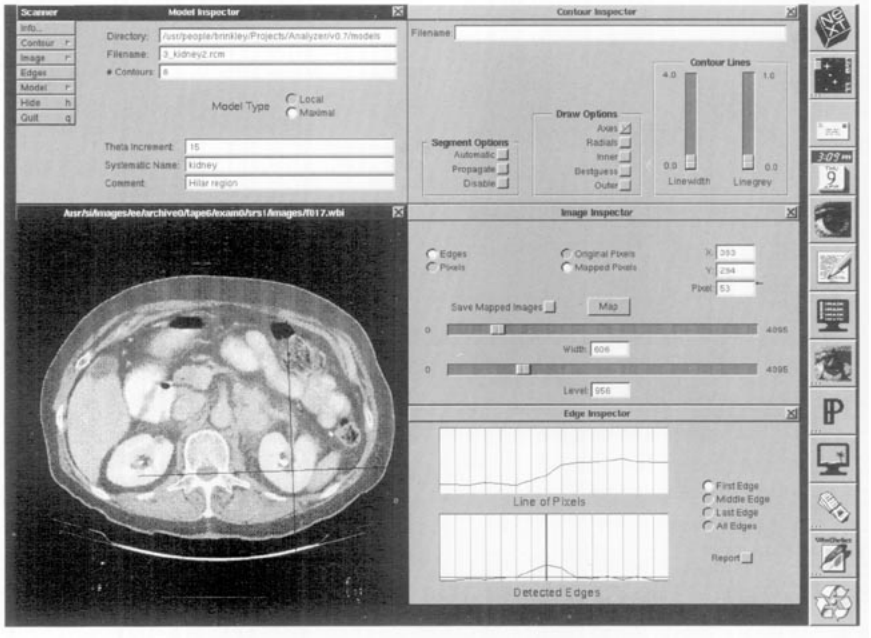

FIG. 2. SCANNER user interface. The user controls the operation of the program through menus (not shown) and panels corresponding to the major data structures: models, contours, images, and edges. The image at the lower left is an abdominal CT image through a kidney. The crossed lines are a user-defined local coordinate system for the radial contour model 3_kidney2. rcm that will be used to find the kidney in Fig. 3.

3.1. Creation of a Radial Contour Model

A single RCM represents the range of variation for a series of similarly shaped contours obtained from a training set. For example, in Fig. 2 the Model Inspector (upper left panel) shows a radial contour model called "3_kidney-2.rcm," which was created from eight 2-D training contours. A new model is created by specifying a filename, and an angle θ defining the increment between fixed radials. In Fig. 2 and in all subsequent discussion theta was set to 15°, giving 24 fixed radials.

Each training contour is interactively added to the model by retrieving the corresponding image, manually defining the radial contour corresponding to the shape in the image, and then adding the contour to the model.

A single radial contour has the same structure as the RCM shown in Fig. 1, except that for each radial R_i, the distance r_i has been defined, so that $r_i = r_{ii} = r_{ib} = r_{io}$ and the uncertainty interval at R_i reduces to the single point r_i. In addition each radial contour has associated with it a transform defining the position and orientation of its local coordinate system with respect to the image.

Manual contour definition is specified by turning on the "Disable" switch in the Contour Inspector (upper right panel in Fig. 2). The position and orientation of the local coordinate system is specified by clicking the mouse on one end of the contour long axis in the image, and dragging a line to the other end. This operation has three effects (1) to align the radial contour with the image, (2) to specify the overall size of the radial contour, and (3) to specify the distances r_i along the two long-axis radials in the radial contour model (Fig. 1).

Once the local coordinate system has been defined the distances r_i along each of the other 22 radials are defined by moving the mouse to a radial on the image and dragging the contour along that radial until it matches the contour in the image.

Once the distances have been defined along each of the 24 radials the radial contour is added as a training instance to the current RCM. The model is updated by extending each constraint interval $[L_{ij}\ U_{ij}]$ between radials R_i and R_j such that it includes the observed ratio $s_{ij} = r_i/r_j$ in the current contour. As more contours are added to the model, the model begins to represent the expected range of variation.

3.2. Using the Radial Contour Model for Model-Based Segmentation

A previously created RCM may be used for interactive model-based segmentation by retrieving the model, turning off the "Disable" switch in the Contour Inspector (Fig. 2, upper right panel), turning on the "Propagate" switch, and retrieving an image containing a shape to be segmented.

The segmentation process is initiated by specifying a long axis, in the same manner as that used for manual contour definition. Definition of the long axis initiates the model-based segmentation procedure, which runs automatically if the "Automatic" switch is set in the Contour Inspector. Initially, the uncertainty intervals $[r_{ii}\ r_{io}]$ for all radials R_i except the long-axis radials are set to very large values ($r_{ii} = 0$, r_{io} = infinity), so all distances for each radial are initially possible.

The long axis radials initiate a constraint propagation algorithm called *arc consistency*, or *relaxation labeling* (*28–30*), which reduces the uncertainty interval for each radial by propagating the known information at the long axis radials throughout the constraint network defined by the RCM. The area labeled "Propagation Wave" in Fig. 1 shows the effect of this algorithm for a local RCM. For the maximal model the "wave" of information propagates to all parts of the model simultaneously. In either case, when the algorithm converges the radial uncertainty intervals are reduced from their initially large values, and the uncertainty region defined by these intervals can be used to limit the search for additional radial edges. Details of this algorithm are given in a previous report (*17*).

The CT image in Fig. 2 shows the local contour coordinate system for a kidney cross-section, after the long axis has been defined. Figure 3A is a close-up of the same kidney cross-section, except that the uncertainty intervals for

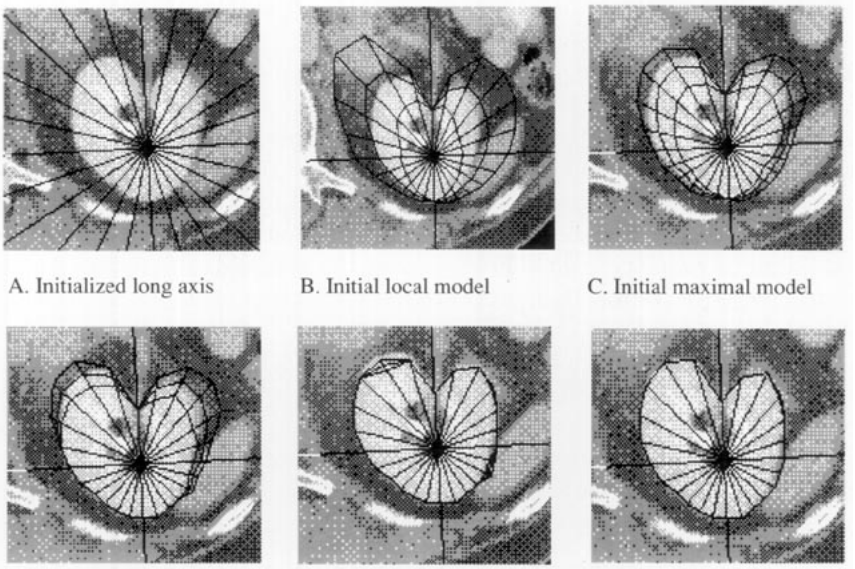

FIG. 3. Model-based segmentation of the kidney. (A) Initialized model after the user has defined the long axis. The two vertical radials are the long axis. The distances to the contour boundary along all other radials have not yet been defined, and their uncertainty intervals are infinite. (B) Local RCM after constraint propagation from the initialized long axis radials. The uncertainty intervals for all radials have been reduced from their initial infinite values, but only the long-axis radials have been defined precisely. (C) Maximal RCM after initial constraint propagation. (D) The RCM after several of the radials have been defined. (E) The RCM after all radials have been defined. (F) The RCM after the user has corrected two errors with the mouse.

the radials are also shown. At this point the uncertainty intervals for all radials, except those defining the long axis, are at their initially large values. Figure 3B and C show the bestguess contour (middle line) and uncertainty region for the *local* and the *maximal* models after the network has achieved arc consistency. Note that the bestguess contour is a reasonable approximation to the actual kidney contour, even though the computer has only "seen" the contour boundary along the two long axis radials. The rest of the contour was generated solely by constraint propagation. The bestguess contour and uncertainty region therefore represent a manifestation of the implicit generic shape knowledge contained within the constraints. The uncertainty region in Fig. 3B is larger than that in Fig. 3C because fewer constraints are present in the local model. The remaining parts of Fig. 3 are generated using the maximal model.

Given the initial uncertainty region the computer chooses a radial along which to search for a contour edge (currently, the radial with the smallest uncertainty interval which has not been examined). Once a radial is chosen, the line of pixels in the image corresponding to the selected radial uncertainty

interval is sent to a one-dimensional edge finder, which displays the line in the upper view of the Edge Inspector shown in Fig. 2 (lower right panel, "Line of Pixels"). The edge finder performs a one-dimensional gradient convolution on the line, and displays it in the lower view ("Detected Edges"). A peak in the gradient is accepted as an edge if it is above a user-setable threshold (the horizontal line in the "Detected Edges" view).

The detected edge defines the distance along the selected radial, thereby reducing its uncertainty interval $[r_{ii}\ r_{io}]$ to the single distance r_i. The constraint propagation algorithm is reapplied, so that the uncertainty region becomes smaller, and the bestguess contour becomes a better approximation to the actual contour.

This process of selecting radials and updating the model is allowed to proceed until all radials have been determined. Figure 3D is an intermediate stage in the process, showing that part of the kidney contour has been found and that the uncertainty as to the location of the remaining portion is smaller than it was initially. Figure 3E shows the final stage, where all radials have been found and the bestguess contour closely matches the contour in the image.

In its current stage of development the program may choose an incorrect edge along a radial. In this, as in all cases, the user is allowed to manually correct the error with the mouse. Figure 3F shows the corrected contour, which then becomes the output of the program. The output contour can be added as a training instance to the model, if desired.

The advantages of this approach to segmentation are: (1) the uncertainty region allows a low-level edge finder to search for edges in small areas of the image, rather than the entire image, so that incorrect edges are found less frequently, and (2) if an edge is not present, as is often the case in soft tissue objects, the constraints allow a reasonable guess to be made. Thus, the system is able to use edge information where it is available, but is able to substitute shape constraints to fill in any gaps, in a manner that is perhaps not to different from the way trained radiologists substitute anatomic knowledge for defects in images.

The main disadvantage of this approach is that it is not completely automatic, in that the user must specify the initial coordinate system as well as correct any errors. However, the program was designed to be interactive on the assumption that segmentation is a very difficult problem that is not likely to be completely solved in the near future. The user is required to perform those tasks which are easy for him or her but difficult for the computer, while the computer is given the bulk of the work. As further research improves the model-based approach, more of the tasks can be transferred to the computer, thereby gradually progressing towards a completely automated system, while in the meantime generating solutions that reduce segmentation time over a completely manual method. If the user is explicitly recognized as an integral part of the process, then a semi-automatic system such as this will be more acceptable than an automatic one that makes noncorrectable mistakes.

3.3. Using the Radial Contour Model for Matching

The RCM can also be used to classify a radial contour by comparing it against a set of stored RCMs representing different contour shape classes. This capability would be useful, for example, for determining whether a given shape is abnormal or not. For example, one previously created RCM may represent cross-sectional shapes of normal kidneys, and another may represent abnormal kidneys. In order to determine which of these shape classes the unknown contour is most similar to, it can be matched against each of the two RCMs, and the model with lowest match score taken to be the most similar shape.

In the SCANNER program, a radial contour is classified by retrieving a series of radial contour models and computing the match score for each one. The match score is computed by determining the ratio $s_{ij} = r_i/r_j$ for each pair of radials R_iR_j in the unknown contour that is connected by a constraint in the model, where r_i and r_j are measured distances along the corresponding radials to the contour boundary. If s_{ij} is within the interval $[L_{ij}\ U_{ij}]$ then 0 is added to the match score (i.e., the constraint is satisfied). Otherwise the squared differences $(s_{ij} - L_{ij})^2$ and $(s_{ij} - \mathrm{U_{ij}})^2$ are computed, and the smaller of these numbers is added to the match score. Thus, the degree of mismatch depends on the amount by which the constraints are unsatisfied. An implementation of this procedure for clinical use would perform these matches automatically, without requiring the user to manually retrieve each possible matching model.

4. Evaluation

4.1. Goals

The goals of the evaluation were to test the major hypotheses implied by the radial contour model, and by extension, the geometric constraint network approach to shape representation. These hypotheses are (1) a network of binary constraints captures both the essential or average shape of various contour shape classes, as well as the range of variation, (2) these shape models, when implemented as part of an interactive segmentation system, can significantly speed up the segmentation process, and (3) the shape models are useful for classifying an unknown contour in a matching system.

4.2. Methods

In order to test these hypotheses, two trialsets comprising routine computed tomography (CT) patient imagesets were obtained. Different patients were used for each trialset. Manually segmented contours from trialset 1 were used as a training set to create radial contour models for several shape classes. These models were used in model-based segmentation of the same shapes observed in trialset 2, and the resulting automatically segmented contours were compared against manually segmented contours of the same shape. The process was then reversed: manually segmented contours from trialset 2 were used as a training set to create models, which were used in model-based segmenta-

tion of the shapes in trialset 1. Manually segmented contours in trialset 1 were also matched against the models created in trialset 2, and vice versa. In each case the model with lowest match score was taken as the best match.

The evaluation was carried out automatically once the radial contours has been manually outlined. Evaluation was carried out by ancillary programs that called SCANNER using remote procedure calls, and which stored results in a relational database running on a separate machine. These programs are components of our image processing framework that will be described elsewhere. The following sections provide more details about the evaluation methods.

4.2.1. Images, structures, and shapes. The CT images are part of an online archive created at the University of Washington (UW) by the Department of Electrical Engineering, in collaboration with the Department of Radiology. For this evaluation, each trialset contained images from eight patients, four of whom had head examinations, and four of whom had abdominal examinations.

Table 1 describes the 15 cross-sectional shapes that were represented with the radial contour model. These shapes were chosen from among those appearing in the abdominal and head CT images in the archive, based on the following criteria: (1) the radial contour model could adequately represent the shape, which must be a *single-valued distortion of a circle;* (2) the shapes are cross-sections through "critical" structures that are routinely manually segmented in clinical situations such as radiation treatment planning; and/or (3) the structures exhibit interesting shapes that demonstrate the utility of the radial con-

TABLE 1

RADIAL CONTOUR MODELS OF CROSS-SECTIONAL SHAPES[a]

Model	Shape description or region of transverse section
Eye1	Middle
Kidney1	Inferior pole
Kidney2	Hilum (middle region)
Kidney3	Superior pole
Liver1	Inferior
Liver2	Middle
Liver3	Superior
Lung2	Hilum (middle region)
Ribs1	Vertically oriented ovoid shape
Spinal Cord1	Circular shape
Spleen1	Inferior pole
Spleen2	Superior pole
Vertebra1	Horizontally oriented ovoid shape
Vertebra2	Indented ovoid shape
Vertebra3	Arched shape

[a] All cross-sections are transverse. Each model was created from eight training contours, two contours from each of four patients Sections from the ribs, spinal cord, and vertebra do not correspond to specific regions, but rather to similar-appearing shapes.

tour model for shape representation. Several of the structures were subdivided into distinct shape classes (for example, Kidney1, Kidney2 and Kidney3). For the eye, kidney, liver, and spleen the shape classes represented cross-sections taken from specified regions along the axis of the organ. For the ribs, spinal cord, and vertebra the shape classes were simply equivalence classes that visually appeared similar.

4.2.2. Radial contour models. Two types of RCM were created for each trialset: a NO-SHAPE model and a SHAPE model.

The NO-SHAPE model contained equal and very wide constraint limits $[L_{ij}\ U_{ij}] = [0.1\ 999]$. These limits generated radial uncertainty intervals that included the entire visible image, from the origin of the local contour coordinate system to the limits of the image on the screen. These wide search intervals were equivalent to adding no shape knowledge at all to the edge finder, so the edge finder searched along the entire radial.

The SHAPE models were maximal radial contour models in which every radial was constrained by every other radial. For each trialset one model was created for each of the 15 shapes, using the 8 manually segmented contours corresponding to that shape as a training set.

4.2.3. Trials. The models from trialset 1 were used in model-based segmentation of the shapes in trialset 2, and vice versa. For example, for each of the 120 contours in trialset 1 (8 contours from each of the 15 shapes) model-based segmentation was performed using each of the two models created from the corresponding contours in trialset 2: the NO-SHAPE model and the SHAPE model. A total of 480 trials were run (2 trialsets $\times$ 120 contours/trialset $\times$ 2 models/contour).

For each trial the corresponding contour and model files were loaded into SCANNER. The two axis endpoints used to establish the local coordinate system for the manually segmented contour were retained in the model-based segmentation to simulate the effect of the user manually defining the initial long axis. All other radial intervals for the contour were reset to their initial wide intervals, after saving a copy of the manually defined radials. The result was two contours whose coordinate systems were exactly superimposed: a gold standard contour with all *radials defined manually by the author,* and a trial contour with only the long axis radials defined. This trial contour corresponded to the initial radial contour that would appear after the user had indicated the long axis, but before any model-based segmentation had taken place (as in Fig. 3A).

Given the initial configuration of the trial contour, model-based segmentation was allowed to proceed as described in Section 3. The two long-axis radials initiated constraint propagation, after which radials were searched for edges and the constraint propagation procedure reapplied.

Once the model-based segmentation procedure had run to completion and all the radials had been searched, the trial contour was compared against the gold standard contour. For each pair of corresponding radials R_i the two radial distances r_i were compared. If the difference between these two distances was

greater than a global parameter EPSILON, set by the user, then the trial contour radial distance was changed to match the gold standard radial distance, and the radial was recorded as a user-defined radial in the database. This procedure was designed to simulate interactive corrections that would be made by a user after observing the results of a segmentation. The greater the number of user-defined radials, the less useful the model-based segmentation would be.

4.2.4. Measurements and global parameters. The measurements recorded for each trial were the number of user-defined radials, the execution clock time starting from initial constraint propagation to just prior to comparison with the gold standard contour, and the number of radial visits (discussed in Section 4.3.2).

The value of EPSILON remained constant for all trials at 4 pixels, a number that was determined by visually examining the appearance of corrected contours from several trials. All other global parameters in the system were also maintained at constant values for all the trials: the image window width and level used to establish the mapping from original 12-bit CT numbers to 8-bit display pixels remained constant at 606 and 956, respectively; pixel values examined by the edge finder were always 8-bit mapped values; the edge threshold was set to 5% of the maximum mapped gray value of 255; and the edge finder was set to always choose the first edge it found that was over the threshold.

The fact that all these global parameters remained constant, together with the exact superposition of trial and gold standard coordinate systems, meant that the only variables in the evaluation were the two different model types, so all differences in usability must be due to the different sized search regions generated by these models; that is, to the application of varying amounts of shape knowledge.

4.3. Results and Implications

4.3.1. Shape representation. Figure 4 shows maximal radial contour shape models that were created for one of the trialsets. Models created for the complementary trialset appeared very similar. Each figure was generated in SCANNER from an initial contour with long axis defined in the vertical direction, and the constraint propagation procedure was allowed to converge. No other radials except the long-axis radials were input manually. Thus, the figures represent the initial bestguess and uncertainty for each shape after seeing just the two long-axis radials, similar to Fig. 3C. All models are shown in this standard orientation and size in order to facilitate shape comparison. However, as described in Section 3, the models are invariant with respect to size, and can be arbitrarily oriented on the image according to the long axis.

4.3.2. Overall usefulness for interactive segmentation. Table 2 gives an overall indication of the expected usefulness of the radial contour model, both in terms of actual execution time and in terms of user-defined radials. The execution time numbers are compared against execution times for manual segmenta-

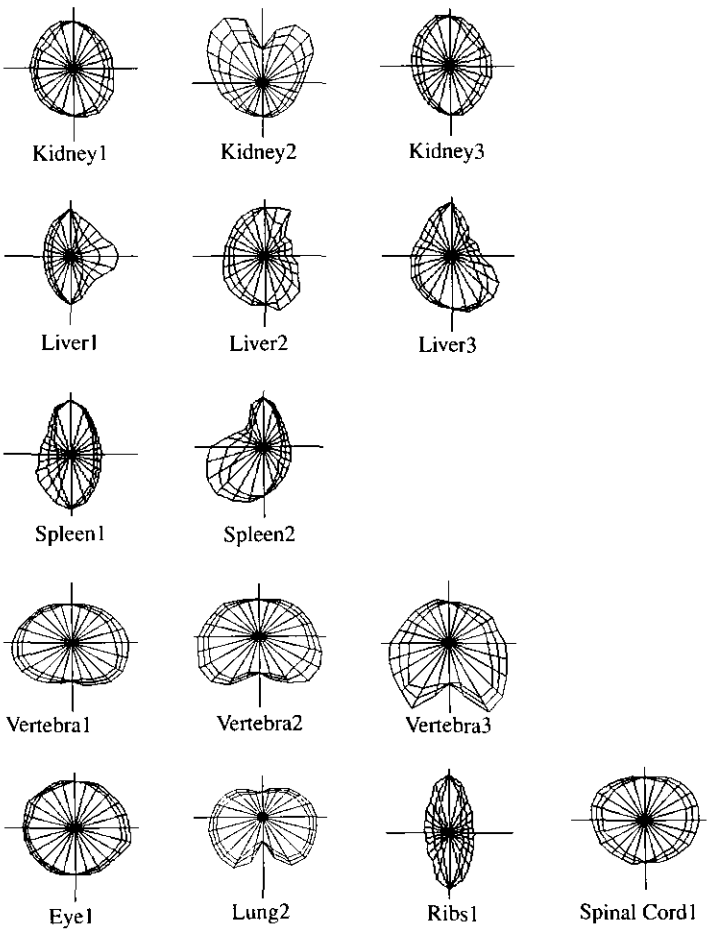

FIG. 4. Radial contour models of several cross-sectional shapes. Each RCM is a maximal model generated by constraint propagation from the initial long-axis radials, as in Fig. 3C. No other radials have been defined.

tion during creation of the training contours. For 40 arbitrarily selected contours the mean clock time for manual segmentation, as measured by a stop watch, was 27.9 sec., with a standard deviation of 7.4 sec. For the 240 trials with the NO-SHAPE model (120 trials for each trialset) the mean execution time was 25.3 sec., and for trials using the SHAPE model the mean execution time was 16.9 sec. All these differences are significant at $p < .01$. Thus, if no mistakes are made, the SHAPE model should speed up segmentation by about a factor of 2 over the manual method.

Of course these times are highly dependent on the particular hardware, which in this case was a NeXT using a Motorola 68040 CPU. In addition, no attempt was made to optimize SCANNER for rapid execution. As hardware

TABLE 2
OVERALL USEFULNESS OF THE RADIAL CONTOUR MODEL[a]

Model type	Trials N	User-defined radials		Execution time[b]		Radial visits	
		Mean	SD	Mean	SD	Mean	SD
None (manual)	40	24.0	0.0	27.9	7.4		
NO-SHAPE	240	18.1	6.8	25.3	1.5		
SHAPE	240	9.0	6.4	16.9	4.2	108.8	37.9

[a] All differences between means for user-defined radials and execution times are significant at $p < .01$.
[b] Seconds.

improves and as the program is optimized for speed it is to be expected that these execution times will dramatically decrease, to the point that execution times will cease to be a major factor in the usefulness of this method.

A more important indicator of expected execution time that is independent of hardware is the mean number of radial visits during each trial, where a radial visit is defined as the examination of a single radial during one iteration of the constraint propagation procedure. This number can be used as an indication of best case execution time, since execution time will be some hardware dependent constant times the number of radial visits. Constraint satisfaction procedures are known in general to take time proportional to the square of the number of nodes in the network, $O(n^2)$ (*29*), in this case the number of radials, which would mean that for larger numbers of radials the execution time would become unacceptably slow, even with order of magnitude speedups in hardware.

However, my previous report showed that, at least for the local model, expected execution time is a linear function of the number of radials in the network, that is $O(n)$ (*17*). It can be shown that the expected execution time of the maximal model is also $O(n)$.

The empirical results confirm that the expected time of the maximal model is $O(n)$. The number of radials in each model was 24. If the constraint propagation procedure visits each radial only one time before it converges, then the number of visits for each instantiation of the procedure should be 24 minus the number of radials that have already been searched and are therefore effectively no longer in the network. Thus, the overall number of radial visits should be 22 for the first application (the total radials minus the long axis radials), 21 for the second, 20 for the third, etc., for a total of $22 + 21 + \ldots = 253$ radial visits. If the procedure were $O(n^2)$ then the number of radial visits should be $22^2 + 21^2 + \ldots = 3795$ radial visits.

The last two columns of Table 2 confirm that the execution time is $O(n)$. The mean number of visits was 109, which is less than half of 253, reflecting the fact that many radials are not visited at all in a given application of the procedure, since they are have already been sufficiently updated in a previous iteration.

The conclusion from this analysis is that the execution time of this procedure will only increase in linear proportion to the number of radials, rather than in exponential proportion, which means that the procedure will scale up to three and even four dimensions, especially as hardware becomes much faster.

A more important measure of usefulness is not execution time, but how often the technique makes mistakes requiring user corrections, that is how many radials must be user-defined. If the number of corrections is too large then even if the procedure executes instantaneously it will be no better than a manual method.

In this evaluation the number of radials was 24, of which 2 were long-axis radials that were always user-defined. Thus, the maximum potential usefulness of a particular shape model would occur when the number of user-defined radials was 2, whereas the model would be useless if the number of user-defined radials was 24, meaning that all 22 radials found by the program had to be corrected by the user.

The second two columns of Table 2 show that without any shape knowledge, the NO-SHAPE model increases usefulness slightly over the manual method (24 user-defined radials for the manual method, 18.1 for the NO-SHAPE model). The shape model further increases usefulness by a factor of 2 over the NO-SHAPE model, and by nearly a factor of 3 over the manual method. Thus, assuming that each manual radial correction takes about the same amount of time as it does for manual segmentation, the model-based approach is expected to speed up segmentation time by a factor of 3 over a manual method, and the addition of shape knowledge, which by the nature of this evaluation is the sole difference between the models, speeds up segmentation time by a factor of 2 over the NO-SHAPE model.

4.3.3. Usefulness for different structures. Table 3 breaks down the usefulness results according to structure. The results in this table were created by combining the trials for all shape models comprising a structure. For example, the 48 Kidney trials are the combined trials from the 16 trials each for Kidney1, Kidney2, and Kidney3.

Table 3 shows that models created from several of the structures were considerably more useful than the average shown in Table 2. For example, the spinal cord had 2 user-defined radials (that is, no corrections), and the eye had a mean of 2.6 user-defined radials. On the other hand the liver had a mean of 15.8 user-defined radials, which is not significantly different than the NO-SHAPE model. The eye and spinal cord are particularly useful because they are critical structures in radiation treatment planning that are routinely manually outlined. The reason these structures are manually outlined is that there is very poor contrast between them and the surrounding tissue, so low-level edge followers or region growers usually fail.

The major reason for the difference between the NO-SHAPE and SHAPE models is the large search regions defined by the NO-SHAPE model. Thus, it is to be expected that an important factor in usability would be the variability in the shape: the greater the variability the larger the search regions and hence the

TABLE 3
USEFULNESS OF THE RADIAL CONTOUR MODEL FOR DIFFERENT STRUCTURES[a]

		Models			Trials		
		Number of models	Variability		Number of trials	User-defined Radials	
Usefulness	Structure	N	Mean	SD	N	Mean	SD
Low	NO-SHAPE				240	18.1	6.8
High	Spinal cord	2	0.24	0.08	16	2.0	0.0
	Eye	2	0.15	0.05	16	2.6	1.1
	Vertebra	6	0.25	0.11	48	4.3	2.6
Medium	Kidney	6	0.40	0.20	48	9.1	5.6
	Spleen	4	0.64	0.55	32	11.3	4.4
Low	Liver	6	1.04	1.03	48	15.8	4.1
Mixed	Ribs	2	1.01	1.11	16	3.6	2.9
	Lung	2	0.25	0.16	16	16.0	4.8

[a] Usefulness is measured by the mean number of user-defined radials. The lower this number the higher the usefulness. Variability for a model is the mean width of a constraint interval. The higher this number the greater the variability. The number of models N is the total number of RCMs created for each structure. The number of trials N for a structure is the number of model-based segmentations of cross-sections through that structure. Differences between mean number of user-defined radials within a usefulness group are generally not significant, differences between groups are significant at $p < .01$. The mean number of user-defined radials for liver and lung are not significantly different from the NO-SHAPE model.

more likely the edge finder will choose an incorrect edge. A convenient measure of model variability is the mean difference $U_{ij} - L_{ij}$ for all constraints $[L_{ij}\ U_{ij}]$ in the model. Thus, high variability is correlated with wide constraint intervals.

Table 3 shows that the relationship between variability and corrections holds true for most structures, the main exceptions being the ribs and the liver in the "Mixed" usefulness category.

In the case of the ribs the number of user-defined radials was low even though the variability was relatively high. Examination of several segmentations using the ribs model showed that the search region usually contained the correct edge, and there was only one edge in the search region because of the good contrast between the rib and the surrounding tissue. Therefore, the size of the search region was not a factor.

In the case of the lung the number of corrections was large even though the model variability was low. The reason for this is that the search regions did not include the actual contour, a situation which occurred in several other instances.

Thus, there are at least two sources of error that can contribute to an incorrect edge: a search region that is so large that too many edges are possible, and a search region that is so small that the correct edge is not considered. On the

TABLE 4
USEFULNESS OF THE RADIAL CONTOUR MODEL FOR MATCHING[a]

Shape	Eye1	Kidney1	Kidney2	Kidney3	Liver1	Liver2	Liver3	Lung2	Ribs1	Spinal Cord1	Spleen1	Spleen2	Vertebra1	Vertebra2	Vertebra3
Eye1	50.0	6.2								43.8					
Kidney1		56.2		25.0	6.2	12.5									
Kidney2			100												
Kidney3		18.8	18.8	50.0	6.2	6.2									
Liver1					93.8	6.2									
Liver2		6.2			43.8	25.0	18.8			6.2					
Liver3		6.2				31.2	62.5								
Lung2								100							
Ribs1					12.5				87.5						
Spinal Cord1	18.8									81.2					
Spleen1					31.2				37.5		31.2				
Spleen2												100			
Vertebra1										6.2			81.2	12.5	
Vertebra2													6.2	81.2	12.5
Vertebra3														6.2	93.8

[a] Percent correct matches out of 16 trials for each shape, where a match is counted correct if a contour representing one of the known shapes in the first column matches the corresponding model in the first row.

other hand if, as in the case of the ribs, there is only one edge along a radial, then this kind of shape knowledge is not useful, since it does not matter how large the search region is.

4.3.4. Matching. Table 4 shows the usefulness of the radial contour model for matching, in terms of percent correct matches out of 16 trials for each shape. A match was called correct if a contour known to represent a shape in the first column matched the corresponding model in the first row. For example, the 16 contours from Eye1 matched the shape model for Eye1 50.0% of the time, for Kidney1 6.2% of the time, and for Spinal Cord1 43.8% of the time.

The percentage of correct matches for all 240 trials from the 15 shapes was 72.9.

When the individual numbers are compared with the appearance of the shape models in Fig. 4, most of the matches make sense. For example, Eye1, Kidney1 and Spinal Cord1 are all almost circles in cross-section, so it is not surprising that an eye contour should match any one of them.

Some of the matches do not appear to be based on similar shape, particularly those from the liver. However, the liver also has high variability, which means that many contours will match it, even if they are not of similar shape. The reason for this is that the match test gives a 0 score (the best possible) if a constraint is satisfied. If a model has very wide constraint limits then many

constraints will be satisfied. The extreme for this is the NO-SHAPE model, in which all constraints are satisfied by every contour, so all contours match the NO-SHAPE model.

On the other hand several shapes had 100% correct matches, namely, Kidney2, Lung2, and Spleen2. What is interesting about these shapes in Fig. 4 is that they all appear very different from the other shapes. The fact that the measuring criterion picks up these distinct shapes is evidence that it is indeed measuring shape similarity, as long as the shapes are not highly variable.

5. Discussion

The evaluation was designed to test three hypotheses regarding the radial contour model: (1) a network of binary constraints captures both the essential or average shape of various contour shape classes, as well as the range of variation; (2) these shape models, when implemented as part of an interactive segmentation system, can significantly speed up the segmentation process over a purely manual method; and (3) the shape models are useful for classifying an unknown contour in a matching system. The evaluation results provide evidence in favor of all three of these hypotheses.

Figure 4 shows that the radial contour model can model a variety of shapes, purely as a network of binary geometric constraints. The middle contour in each of these models represents the essential or average shape and the inner and outer contours are a manifestation of the range of variation. Although more stringent network consistency algorithms might reduce the manifested variability, the key point is that even the limited form of consistency achieved by the relaxation procedure produces an uncertainty region that is small enough to be useful for model-based imaging.

Tables 2 and 3 confirm that the shape models can be useful for interactive image segmentation, both in terms of actual execution time and in the number of required user-defined radials. Because all other variables were held fixed in the evaluation, the demonstrated change in expected usefulness is due entirely to the different search regions provided by the NO-SHAPE and the SHAPE models. The more detailed analysis in Table 3 shows that usefulness varies with the structure, and that the major difference between models is model variability. Thus, a structure such as the liver, that exhibits large variability, will generate large search regions, so the current model-based approach will not be that useful. On the other hand a very regular structure such as the eye will general small search regions, so the model-based approach will be very useful even when the boundary is indistinct.

A second potential source of error is search regions that are too small, as in the case of the lung. However, this error source will likely be minimized when larger numbers of training instances are used. Evidently, eight training instances are not enough to capture the variability for all model classes.

The match results in Table 4 show that the model may also prove useful for shape classification. The fact that many of the incorrect matches were very

similarly-shaped structures, and that structures with very different shapes had 100% correct matches suggest that the utility of the RCM model for matching may be quite high, especially when combined with other knowledge. Examples of additional knowledge that could be added relatively easily are the body region containing the structure, the expected size and orientation of the structure, and the range of intensity values for image regions known to contain the structure. The possibility therefore arises for determining whether a particular contour represents a pathological shape by comparing it with a library of previously classified abnormal shape classes.

A second potential use for matching might be classification of image regions found by low-level image processing operations. That is, the long axis of a thresholded region could be found using principle axes (*32*), radials could be fitted to the region, and the resulting radial contour could be compared against a library of previously classified regions represented as a single RCM. Such a capability would greatly reduce the need for the user to initially define the long axis, although difficulties could arise if the thresholded regions were not representative of the actual shape because of poor contrast between neighboring structures.

6. Conclusion

This paper has described a two-dimensional shape model that is both *flexible,* in that it can deform to fit a specific instance of a shape, and *generic,* in that it captures all expected instances of a shape class. Evaluations on routine CT images have shown that, when the model is incorporated in an interactive system that allows the user to easily correct errors, the overall system is faster than manual segmentation. Thus, the current version of SCANNER should be useful for certain tasks such as radiation treatment planning or ultrasound image analysis. As the model is improved and generalized in the direction of general geometric constraint networks, it should become an increasingly useful tool, not only for medical image segmentation, but also for capturing spatial knowledge at all levels of the anatomic spectrum, from organs to molecules. Such a generalized model would be very valuable for many knowledge-based applications in structural biology (*13*).

Acknowledgments

This work was funded by National Library of Medicine Grant LM04925, National Cancer Institute Grant CA59070, the Murdock Foundation Charitable Trust, the W. M. Keck Foundation, and the University of Washington School of Medicine. The UW medical image archive is maintained in the Department of Electrical Engineering by Greg Zick and Eliot Lim. The images were obtained from the Radiology Department by David Haynor in that department. The imaging project is one component of the Digital Anatomist program in our department, which is directed by Cornelius Rosse, and includes John Prothero, Steve Broderson, Bill Barker, Kraig Eno, Jeff Prothero, David Conley, John Sundsten, and myself. I thank all the members of this project for their support. I also thank Cornelius Rosse, John Prothero, and Linda Shapiro for providing feedback on drafts of the manuscript.

REFERENCES

1. UDUPA, J. K., AND HUNG, H. Surface versus volume rendering: a comparative assessment. *In* "Proc. 1st Conf. on Visualization in Biomedical Computing," pp. 83–91. IEEE Computer Society Press, Washington, D.C., 1990.
2. BRINKLEY, J. F., MCCALLUM, W. D., MURAMATSU, S. K., AND LIU, D. Y. Fetal weight estimation from ultrasonic three-dimensional head and trunk reconstructions: Evaluation *in vitro*. *Amer. J. Obstet. Gynecol.* **144,** 715 (1982).
3. DETMER, P. R., BASHEIN, G., AND MARTIN, R. W. Matched filter identification of left-ventricular endocardial borders in transesophageal echocardiograms. *IEEE Trans. Med. Imaging.* **9,** 396 (1990).
4. MOSS, R. H., STOECKER, W. V., LIN, S-J., MURUGANANDHAN, S., CHU, K.-F., PONELEIT, K. M., AND MITCHELL, C. D. Skin cancer recognition by computer vision. *Comput. Med. Imaging Graphics* **13,** 31 (1989).
5. CONLEY, D. M., KASTELLA, K. G., SUNDSTEN, J. W., RAUSCHNIG, W., AND ROSSE, C. Computer-generated three-dimensional reconstruction of the mediastinum correlated with sectional and radiological anatomy. *Clin. Anat.* **5,** (1992).
6. SUNDSTEN, J. W., KASTELLA, K. G., AND CONLEY, D. M. Videodisc animation of 3D computer reconstructions of the human brain. *J. Biocommun.* **18,** 45 (1991).
7. KALET, I. J., AND JACKY, J. P. A research-oriented treatment planning program system. *Comput. Prog. Biomed.* **14,** 85 (1982).
8. BALLARD, D. H., AND BROWN, C. M. "Computer Vision." Prentice–Hall, Englewood Cliffs, New Jersey, 1982.
9. KARSSEMEIJER, N., VAN ERNING, L. J. TH. O., AND EIJKMAN, EG. G. J. Recognition of organs in CT-image sequences: A model guided approach. *Comput. Biomed. Res.* **21,** 434 (1988).
10. MATSUYAMA, T. Expert systems for image processing: Knowledge-based composition of image analysis processes. *Comput. Vision Graphics Image Process.* **48,** 22 (1989).
11. DAVIS, D. N., AND TAYLOR, C. J. A blackboard architecture for medical image interpretation. *SPIE Medical Imaging V: Image Processing,* 421 (1991).
12. GONG, L., AND KULIKOWSKI, C. A. Automatic generation of plans for biomedical image interpretation. *SCAMC* **15,** 465 (1991).
13. BRINKLEY, J. F., PROTHERO, J. S., PROTHERO, J. W., AND ROSSE, C. A. Framework for the design of knowledge-based systems in structural biology. *SCAMC* **13,** 61 (1989).
14. SOROKA, B. I. Generalized cones from serial sections. *Comput. Graphics Image Process.* **15,** 154 (1981).
15. SHANI, U. "Understanding Three-Dimensional Images: The Recognition of Abdominal Anatomy from Computer Axial Tomograms (CAT)." Ph.D. dissertation, University of Rochester, Department of Computer Science, 1981.
16. KAPOULEAS, I., AND KULIKOWSKI, C. A. A model-based system for the interpretation of MR human brain scans. *SPIE Med. Imaging II* **914,** 429 (1988).
17. BRINKLEY, J. F. Knowledge-driven ultrasonic three-dimensional organ modelling. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI 7,** 431 (1985).
18. FABER, T. L., STOKELY, E. M., PESHOCK, R. M., AND CORBETT, J. R. A model-based four-dimensional left ventricular surface detector. *IEEE Trans. Med. Imaging.* **10,** 321 (1991).
19. BINFORD, T. O. Survey of model-based image analysis systems. *Robotics Res.* **1,** 1 (1982).
20. BRINKLEY, J. F. Representing biologic objects as geometric constraint networks. *In* "Proceedings, AAAI Spring Symposium Series: Artificial Intelligence in Medicine" (R. Patil, Ed.), pp. 7–8. AAAI, Menlo Park, CA, 1988.
21. BRINKLEY, J. F. Hierarchical geometric constraint networks as a representation for spatial structural knowledge. *SCAMC* **16,** 140 (1992).
22. BRINKLEY, J. F., ALTMAN, R. B., DUNCAN, B. S., BUCHANAN, B. G., AND JARDETZKY, O. Heuristic refinement method for the derivation of protein solution structures: Validation on cytochrome b562. *J. Chem. Inf. Comput. Sci.* **28,** 194 (1988).

23. KASS, M., WITKIN, A., AND TERZOPOULOS, D. Snakes: Active contour models. *Int. J. Comput. Vision* **1,** 321 (1988).
24. BAJSCY, R., AND KOVACIC, S. Multiresolution elastic matching. *Comput. Vision Graphics Image Process.* **46,** 1 (1989).
25. SEITZ, R. J., BOHM, C., GREITZ, T., ROLAND, P. E., ERIKSSON, L., BLOMQVIST, G., ROSENQVIST, G., AND NORDELL, B. Accuracy and precision of the computerized brain atlas programme for localization and quantification in positron emission tomography. *J. Cerebral Blood Flow Metabol.* **10,** 443 (1990).
26. MARRETT, S., EVANS, A. C., COLLINS, L., AND PETERS, T. M. A. Volume of interest (VOI) atlas for the analysis of neurophysiological image data. *SPIE Medical Imaging III: Image Processing* **1092,** 467 (1989).
27. BROOKS, R. A. Model-based three-dimensional interpretation of two-dimensional images. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-5,** 140 (1983).
28. MACKWORTH, A. K. Consistency in networks of relations. *Artif. Intell.* **8,** 99 (1977).
29. HARALICK, R., AND SHAPIRO, L. The consistent labelling problem: Part I. *IEEE Trans. Pattern Anal. Mach. Intell.* **1,** 173 (1979).
30. WALTZ, D. Understanding line drawings of scenes with shadows. *In* "The Psychology of Computer Vision" (P. H. Winston, Ed.), pp. 19–91. McGraw–Hill, New York, 1975.
31. ENO, K., SUNDSTEN, J. W., AND BRINKLEY, J. F. A multimedia anatomy browser incorporating a knowledge base and 3-D images. *SCAMC* **15,** 727 (1991).
32. ARUN, K. S., HUANG, T. S., AND BLOSTEIN, S. D. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-9,** 698 (1987).